

KECERDASAN BUATAN DIGITAL



Rizal Furqan Ramadhan, Wahyuddin S, Fahmy Rinanda Saputri,
Jarwo, Johni S Pasaribu, Elisawati, Muhamad Faza Almaliki,
Citra Nurina Prabiantissa, Melissa Indah Fianty,
St. Hajrah Mansyur, Angga Aditya Permana, Ratna Dewi,
Risanto Darmawan, Suwito Pomalingo, Moh Safii

KECERDASAN BUATAN DIGITAL

Rizal Furqan Ramadhan

Wahyuddin S

Fahmy Rinanda Saputri

Jarwo

Johni S Pasaribu

Elisawati

Muhamad Faza Almaliki

Citra Nurina Prabiantissa

Melissa Indah Fianty

St. Hajrah Mansyur

Angga Aditya Permana

Ratna Dewi

Risanto Darmawan

Suwito Pomalingo

Moh Safii



PT GLOBAL EKSEKUTIF TEKNOLOGI

KECERDASAN BUATAN DIGITAL

Penulis :

Rizal Furqan Ramadhan

Wahyuuddin S

Fahmy Rinanda Saputri

Jarwo

Johani S Pasaribu

Elisawati

Muhamad Faza Almaliki

Citra Nurina Prabiantissa

Melissa Indah Fianty

St. Hajrah Mansyur

Angga Aditya Permana

Ratna Dewi

Risanto Darmawan

Suwito Pomalingo

Moh Safii

ISBN : 978-623-198-402-9

Editor : Ari Yanto, M.Pd

Penyunting : Tri Putri Wahyuni, S.Pd

Desain Sampul dan Tata Letak : Atyka Trianisa, S.Pd

Penerbit : PT GLOBAL EKSEKUTIF TEKNOLOGI

Anggota IKAPI No. 033/SBA/2022

Redaksi :

Jl. Pasir Sebelah No. 30 RT 002 RW 001

Kelurahan Pasie Nan Tigo Kecamatan Koto Tangah

Padang Sumatera Barat

Website : www.globaleksekutifteknologi.co.id

Email : globaleksekutifteknologi@gmail.com

Cetakan pertama, Juni 2023

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk
dan dengan cara apapun tanpa izin tertulis dari penerbit.

KATA PENGANTAR

Segala Puji dan syukur atas kehadiran Allah SWT dalam segala kesempatan. Sholawat beriring salam dan doa kita sampaikan kepada Nabi Muhammad SAW. Alhamdulillah atas Rahmat dan Karunia-Nya penulis telah menyelesaikan Buku Kecerdasan Buatan Digital ini.

Buku Ini Membahas Konsep kecerdasan buatan, Masalah dan ruang masalah, Sistem produksi, Pencarian, Metode – metode pencarian, *Reasoning : Propositional logic*, *Reasoning : first order logic*, *Reasoning : Logika fuzzy*, Representasi pengetahuan, Sistem berbasis pengetahuan, Jaringan syaraf tiruan, Planning, Learning, Algoritma genetika, Cognitif science.

Proses penulisan buku ini berhasil diselesaikan atas kerjasama tim penulis. Demi kualitas yang lebih baik dan kepuasan para pembaca, saran dan masukan yang membangun dari pembaca sangat kami harapkan.

Penulis ucapkan terima kasih kepada semua pihak yang telah mendukung dalam penyelesaian buku ini. Terutama pihak yang telah membantu terbitnya buku ini dan telah mempercayakan mendorong, dan menginisiasi terbitnya buku ini. Semoga buku ini dapat bermanfaat bagi masyarakat Indonesia.

Padang, Juni 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB 1 KECERDASAN BUATAN.....	1
1.1 Pendahuluan	1
1.2 Sejarah Kecerdasan Buatan.....	3
1.3 Bagian-bagian Kecerdasan Buatan.....	5
1.4 Perkembangan dan Efek Kecerdasan Buatan	8
DAFTAR PUSTAKA.....	9
BAB 2 MASALAH & RUANG MASALAH.....	11
2.1 Pendahuluan	11
2.2 Perkembangan Kecerdasan Digital	12
2.3 Masalah Umum Kecerdasan Buatan	14
2.3.1 Kepercayaan dan Etika	14
2.3.2 Keamanan dan Perlindungan Data	15
2.3.3 Keterbatasan Teknologi	15
2.3.4 Pengambilan Keputusan dan Tanggung Jawab..	15
2.3.5 Pengaruh Sosial dan Ekonomi	15
2.3.6 Regulasi dan Kebijakan	16
2.3.7 Bias dan Diskriminasi	16
2.3.8 Pengangguran dan Disrupsi Pasar Kerja	16
2.4 Peluang & Ancaman AI.....	17
2.4.1 Peluang Kecerdasan Buatan Digital (AI).....	17
2.4.2 Ancaman Kecerdasan Buatan Digital (AI).....	18
2.5 Tantangan Kecerdasan Buatan Digital (AI)	19
2.6 Hukum Kecerdasan Buatan Digital (AI)	21
DAFTAR PUSTAKA.....	23
BAB 3 SISTEM PRODUKSI	25
3.1 Pendahuluan	25
3.2 Konsep Sistem Produksi.....	27

3.3 Komponen Sistem Produksi	29
3.2.1 Basis Pengetahuan	29
3.2.2 Basis Data dan Working Memory	31
3.2.3 Sistem Kendali	33
3.4 Prosedur Kerja Sistem Produksi	34
3.5 Strategi Pencarian di Sistem Produksi	35
3.6 Metode Penalaran dalam Sistem Produksi	38
3.7 Studi Kasus	45
3.7.1 Studi Kasus menggunakan Metode Forward Chaining	45
3.7.2 Studi Kasus menggunakan Metode Backward Chaining	46
3.7.3 Studi Kasus menggunakan Metode Abduction ...	47
3.7.4 Studi Kasus menggunakan Metode Deduction ...	48
3.7.5 Studi Kasus menggunakan Metode Induction	49
DAFTAR PUSTAKA	51
BAB 4 PENCARIAN (SEARCHING)	53
4.1 Pengertian Searching dalam AI	53
4.2 Ruang Lingkup Searching dalam AI	55
4.3 Tujuan Searching	58
4.4 Sifat Masalah dalam Searching AI	59
4.5 Kriteria Searching dalam AI	61
4.6 Algoritma Pencarian dalam AI	62
DAFTAR PUSTAKA	65
BAB 5 METODE-METODE PENCARIAN	67
5.1 Pendahuluan	67
5.2 Tinjauan pustaka	69
5.3 Traversal Graf	71
5.4 Metode pencarian dalam kecerdasan buatan	73
5.5 Metode pencarian buta (<i>exhaustic/blind</i>)	75
5.5.1 Breadth-first search	75
5.5.2 Depth-first search	79
5.6 Metode pencarian informasi (heuristik)	81

5.6.1 Hill climbing search	82
5.6.2 Best-first search	83
5.6.3 A* search	86
DAFTAR PUSTAKA	89
BAB 6 REASONING : PROPOSITIONAL LOGIC.....	91
6.1 Pegantar Reasoning	91
6.2 Proposional Logic.....	91
6.2.1 Inferensi logika proposisi.....	96
DAFTAR PUSTAKA	100
BAB 7 REASONING : FIRST ORDER LOGIC	101
7.1 Definisi First Order Logic.....	101
7.2 Simbol First Order Logic	102
7.3 Representase Semantik	104
7.4 Model, Domain, dan Formula	106
7.5 Contoh Kasus Implementasi First Order Logic.....	106
7.6 Arsitektur Analisis Semantik.....	107
7.7 Leksikon.....	108
7.8 Semantik Leksikon	109
7.9 Pendekatan Montague.....	110
DAFTAR PUSTAKA	111
BAB 8 REASONING : LOGIKA FUZZY	113
8.1 Reasoning (Penalaran).....	113
8.2 Definisi Logika Fuzzy	114
8.3 Sejarah Logika Fuzzy	115
8.4 Himpunan Fuzzy.....	116
8.4.1 Crisp Set.....	116
8.4.2 Fuzzy Set (Himpunan Fuzzy)	116
8.4.4 Fuzzy Set Operation	117
8.5 Tahapan Proses Logika Fuzzy	119
8.5.1 Fuzzyfikasi	120
8.5.2 Fuzzy Inferensi System (FIS).....	120
8.5.3 Defuzzifikasi.....	124
8.6 Penerapan Logika Fuzzy	124

DAFTAR PUSTAKA	129
BAB 9 REPRESENTASI PENGETAHUAN.....	131
9.1 Definisi Representasi Pengetahuan.....	131
9.2 Teknik Representasi Pengetahuan	136
9.2.1 Klasifikasi Representasi Pengetahuan	138
9.2.2 Syarat Representasi Pengetahuan.....	139
9.2.3 Keuntungan Representasi Pengetahuan	139
9.3 Representasi Logika	140
9.4 List dan Tree.....	145
9.4.1 List	145
9.4.2 Tree.....	146
9.5 Jaringan Sematik.....	146
9.6 Frame	146
9.7 Script (Naskah).....	148
9.8 Aturan Produksi (Kaidah Produksi).....	150
DAFTAR PUSTAKA	153
BAB 10 SISTEM BERBASIS PENGETAHUAN:	
KONSEP, PERANCANGAN, DAN IMPLEMENTASI.....	155
10.1 Pendahuluan	155
10.1.1 Apa itu sistem berbasis pengetahuan?.....	155
10.1.2 Sejarah dan Perkembangan Sistem Berbasis Pengetahuan	155
10.1.3 Manfaat dan Aplikasi Sistem Berbasis Pengetahuan	157
10.2 Konsep Dasar Sistem Berbasis Pengetahuan.....	158
10.2.1 Pengenalan Pengetahuan	158
10.2.2 Sistem Berbasis Pengetahuan.....	165
10.2.3 Representasi Pengetahuan	170
10.2.4 Bentuk-Bentuk Representasi Pengetahuan	172
10.2.5 Logika Proposisional dan Logika Predikat	174
10.2.6 Representasi Pengetahuan dengan Aturan	175
10.3 Perancangan Sistem Berbasis Pengetahuan	177
10.3.1 Tahap Analisis.....	177

10.3.2 Tahap Perancangan.....	184
10.3.3 Tahap Implementasi	190
10.4 Aplikasi Sistem Berbasis Pengetahuan	199
10.4.1 Sistem Pakar.....	199
10.4.2 Penunjang Keputusan.....	214
10.4.3 Sistem Rekomendasi.....	226
10.5 Tantangan dan Tren Masa Depan	238
10.5.1 Tantangan dalam Sistem Berbasis Pengetahuan	238
10.5.2 Tren Masa Depan.....	248
10.5.3 Penggunaan big data dalam sistem berbasis pengetahuan.....	253
10.5.4 Penerapan sistem berbasis pengetahuan dalam industri dan sektor lainnya	256
DAFTAR PUSTAKA.....	261
BAB 11 JARINGAN SYARAF TIRUAN.....	263
11.1 Pendahuluan.....	263
11.2 Sejarah Jaringan Syaraf Tiruan	269
11.3 Kelebihan Jaringan Syaraf Tiruan	274
11.4 Kekurangan Jaringan Syaraf Tiruan	278
11.5 Penggunaan Algoritma JST	281
11.6 Contoh Kasus Penggunaan Algoritma JST.....	288
11.7 Cara Kerja Algoritma JST	292
DAFTAR PUSTAKA.....	294
BAB 12 PLANNING.....	295
12.1 Pengertian Planning.....	295
12.2 Kecerdasan buatan dapat dilihat dari sudut pandang, adalah sebagai berikut:	299
12.3 Faktor-faktor yang mendorong pengembangan kecerdasan.....	300
12.4 Teknik Dasar dalam Artificial Intelligence	301
12.5 Konsep Dasar Kecerdasan Buatan (<i>Artificial Intelligence</i>)	303

12.6 Teknologi Artificial Intelligence dan Manusia	305
12.7 Pendekatan Planning.....	307
DAFTAR PUSTAKA	310
BAB 13 LEARNING	313
13.1 Pendahuluan.....	313
13.2 Konsep Dasar.....	314
13.3 Teknik Pembelajaran	315
13.3.1 Supervised Learning.....	316
13.3.2 Unsupervised Learning	317
13.3.3 Semi-supervised learning.....	318
13.3.4 Reinforcement Learning	319
13.4 Algoritma Pembelajaran	320
13.4.1 Decision Tree.....	320
13.4.2 Naive Bayes.....	321
13.4.3 K-Nearest Neighbors	323
13.4.4 Artificial Neural Networks	324
13.4.5 Support Vector Machines	325
13.5 Aplikasi Pembelajaran	326
13.6 Evaluasi Pembelajaran.....	328
DAFTAR PUSTAKA	337
BAB 14 ALGORITMA GENETIKA.....	339
14.1 Pengantar Algoritma Genetika	339
14.1.1 Sejarah dan Evolusi Algoritma Genetika	339
14.1.2 Prinsip Dasar Algoritma Genetika.....	340
14.2 Representasi dan Kromosom	341
14.2.1 Representasi Biner.....	342
14.2.2 Representasi Real.....	342
14.2.3 Representasi Permutasi	343
14.3 Operator Algoritma Genetika.....	344
14.3.1 Seleksi	345
14.3.2 Crossover (Rekombinasi).....	346
14.3.3 Mutasi.....	348
14.4 Strategi dan Parameter Algoritma Genetika	348

14.4.1 Populasi dan Inisialisasi	349
14.4.2 Kriteria Terminasi	350
14.5 Evaluasi dan Fitness Function.....	350
14.5.1 Konsep Fitness Function	351
14.5.2 Desain Fitness Function yang Efektif	352
14.6 Aplikasi Algoritma Genetika.....	353
14.6.1 Optimasi Fungsi Matematika.....	353
14.6.2 Optimasi Rute dan Penjadwalan	354
14.7 Algoritma Genetika Paralel dan Hibrid.....	356
14.7.1 Algoritma Genetika Paralel	356
14.7.2 Teknik Integrasi dengan Algoritma Lain.....	358
14.8 Evaluasi Kinerja dan Metrik.....	359
14.9 Studi Kasus dan Implementasi.....	360
14.9.1 Studi Kasus: Optimasi Fungsi Ackley.....	360
14.9.2 Studi Kasus: Penjadwalan Mesin Job Shop.....	361
DAFTAR PUSTAKA	364
BAB 15 COGNITIF SCIENCE.....	365
15.1 Sejarah perkembangan cognitif science	365
15.2 Tujuan dari cognitif science	366
15.3 Cognitif science dalam bidang teknologi.....	367
15.4 Pentingnya mempelajari Cognitif science.....	374
15.5 Kajian Cognitif science.....	376
15.6 Algoritma dalam Cognitice Science	377
DAFTAR PUSTAKA	382
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 1.1. Diagram Kecerdasan Buatan.....	2
Gambar 1.2. Diagram data dan Informasi	4
Gambar 1.3. Bagan Sistem Komputer	6
Gambar 1.4. Digram Computer Science.....	7
Gambar 4.1. Chatbot, terapan Searching AI Digital	54
Gambar 4.2. Ruang Keadaan Lingkup Search Robot dengan keterangan aktifitas Vacuum. L = left, kiri, R= Right, Kanan, S = Suck, menghisap.	57
Gambar 5.1. Graf berarah	72
Gambar 5.2. <i>Social Graph</i>	72
Gambar 5.3. Algoritma Pencarian.....	75
Gambar 5.4. Pohon traversal breadth-first.....	76
Gambar 5.5. Graf 16-node untuk pencarian informasi...77	
Gambar 5.6. Representasi graf dari urutan simpul dalam pencarian breadth-first dari simpul awal N_1 ke simpul tujuan N_9	78
Gambar 5.7. Pohon traversal depth-first.....	79
Gambar 5.8. Representasi graf dari urutan simpul dalam pencarian depth-first dari simpul awal N_1 ke simpul tujuan N_7	80
Gambar 5.9. Representasi graf dari urutan simpul dalam pencarian hill-climbing.....	82
Gambar 5.10. Representasi graf dari langkah-langkah contoh <i>best-first search</i>	85
Gambar 5.11. Representasi graf untuk pencarian A^* dengan jarak garis lurus dari setiap node ke node tujuan	87
Gambar 8.1. Proses Logic.....	114
Gambar 8.2. Nilai tegas dan fuzzy.....	115
Gambar 8.3. Operasi pada himpunan fuzzy	118

Gambar 8.4. Fuzzyfikasi Suhu 35 °C	120
Gambar 8.5. Proses Inferensi Metode Tsukamoto.....	121
Gambar 8.6. Metode Defuzzifikasi pada Metode Mamdani	123
Gambar 12.1. Kecerdasan Buatan digital planning	298
Gambar 12.2. Teknik Dasar dalam <i>Artificial Intelligence</i>	301
Gambar 13.1. Teknik pembelajaran.....	316
Gambar 13.2. Supervised Learning	317
Gambar 13.3. Unsupervised Learning.....	318
Gambar 13.4. Semi Supervised Learning	319
Gambar 13.5. Reinforcement Learning.....	320
Gambar 13.6. Decision Tree	321
Gambar 13.7. Naïve Bayes	322
Gambar 13.8. KNN.....	323
Gambar 13.9. ANN.....	324
Gambar 13.10. SVN.....	326
Gambar 13.11. Confusion Matrix	329
Gambar 13.12. Cross Validation	330
Gambar 13.13. AUC - ROC Curve	331
Gambar 13.14. AUC - ROC Curve	333
Gambar 13.15. Mean Squared Error	333
Gambar 13.16. Mean Absolute Error	335
Gambar 13.17. Root Mean Square Error	335
Gambar 13.18. Mean Absolute Percentage Error	336

DAFTAR TABEL

Tabel 6.1. Tabel Konjungsi (<i>AND</i>).....	93
Tabel 6.2. Tabel Disjungsi (<i>OR</i>).....	93
Tabel 6.3. Tabel Implikasi (<i>IF-THEN</i>).....	93
Tabel 6.4. Tabel Equivalensi (<i>IF AND ONLY IF</i>).....	93
Tabel 6.5. Tabel Negasi (<i>NOT</i>).....	93
Tabel 6.6. Tabel Tautologi atau Bukan	94
Tabel 6.7. Tabel Tautologi atau Bukan	95
Tabel 6.9. Hukum Inferensi Proposisi.....	96

BAB 1

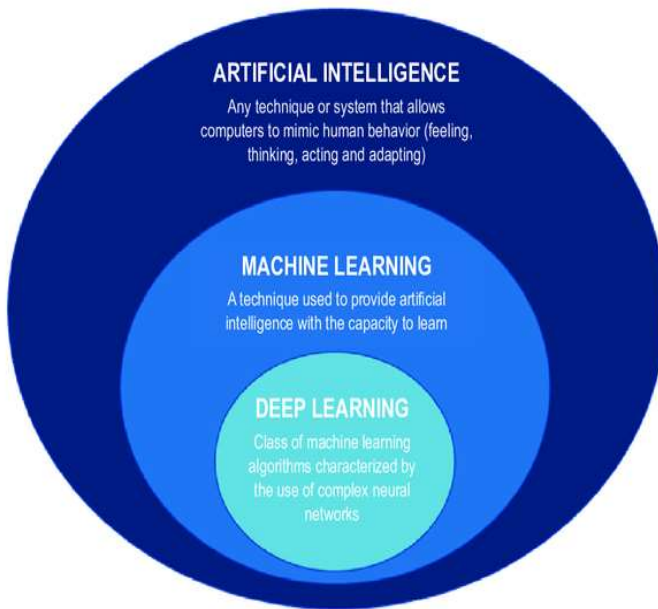
KECERDASAN BUATAN

Oleh Rizal Furqan Ramadhan

1.1 Pendahuluan

Manusia merupakan makhluk yang dianugerahi kemampuan akal yang luar biasa. Dunia semakin pesat kemajuannya merupakan salah satu akibat semakin meningkatnya pola pikir manusia serta didukung dengan perkembangan zaman yang cepat. Dewasa ini, bidang teknologi informasi yang begitu cepat mempengaruhi berbagai bidang yang lain (Ramadhan 2023). Salah satu faktor yang membuat teknologi menjadi berkembang pesat adalah transformasi akal pikiran manusia kedalam sebuah teknologi informasi. Pekerjaan dan aktifitas manusia yang awalnya dikerjakan oleh manusia sendiri mampu digantikan oleh sistem komputer yang tidak memiliki rasa lelah dan *real time* atau dapat menghasilkan keluaran sesuai dengan keinginan pengguna (*user*)(R. Ramadhan & Eliyen 2022).

Kecerdasan buatan berasal dari kata *Artificial Intelligent*, yang mempunyai arti sebagai berikut. Kata "*intelligence*" bersalah dari bahasa Latin yakni "*intelligo*" yang memiliki makna "saya paham". Makna dari *intelligence* adalah suatu keahlian dalam memahami serta melakukan aksi.



Gambar 1.1. Diagram Kecerdasan Buatan
Sumber. <https://www.researchgate.net/>

Didalam kecerdasan buatan terdapat proses perhitungan matematis yang berbeda-beda untuk pemecahan masalah (R. F. Ramadhan & Eliyen 2022). Proses matematis yang dirumuskan oleh manusia kemudian diterjemahkan dalam bentuk bahasa pemrograman atau sejenisnya kemudian dieksekusi oleh mesin komputer. Keluaran dari proses tersebut dapat menggantikan kinerja manusia serta dapat diproses dengan cepat dan waktu yang tidak terlalu lama. Kemampuan komputer dalam menerapkan teknik kecerdasan buatan sudah banyak dirasakan dan ditemui dikalangan masyarakat, sehingga dengan adanya kecerdasan buatan kegiatan serta aktifitas manusia sangat terbantu khususnya mengurangi

beban kerja serta tenaga yang banyak (Ramadhan & Widodo 2022).

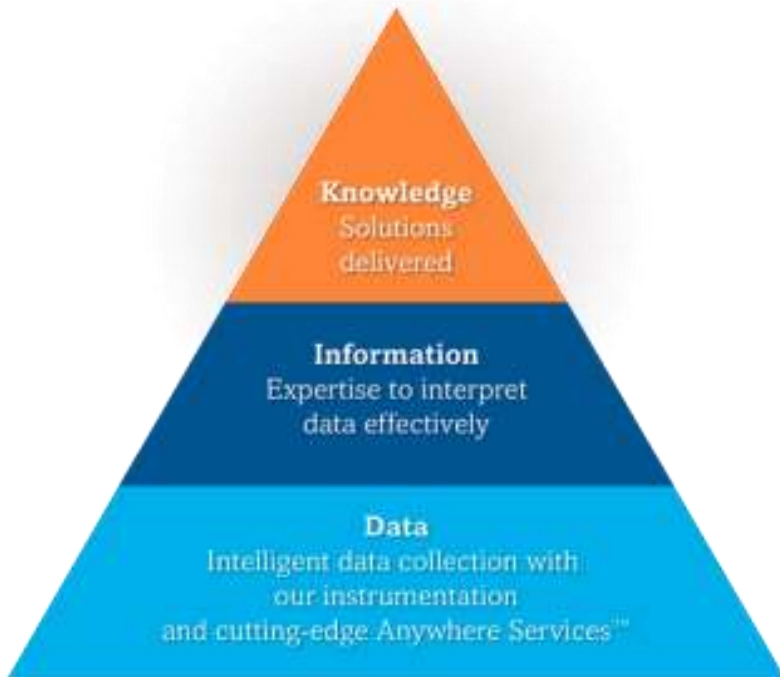
1.2 Sejarah Kecerdasan Buatan

Sekitar tahun 1956 pada musim panas diadakan sebuah pertemuan yang dihadiri oleh pakar komputer, peneliti dari berbagai perguruan tinggi di Dartmouth College. Pada pertemuan tersebut membahas sebuah topik yakni teknik transformasi kecerdasan manusia kedalam komputer sehingga dapat menggantikan kinerja manusia sebagai bentuk keluarannya. Peneliti atau pakar komputer yang hadir pada pertemuan tersebut antara lain Allen Newel, Herbert Simon, Marvin Minsky, Oliver Selfridge, dan John McCarthy (Amrizal & Aini 2013). Dari pertemuan inilah para ahli memulai mengembangkan teknik kecerdasan buatan hingga saat ini dapat dinikmati oleh manusia pada umumnya. Pada saat itu ilmuwan tidak hanya mengembangkan dari segi pola pikir saja namun juga pada fasilitas seperti laboratorium dikembangkan supaya dapat mendorong teknik kecerdasan buatan dikembangkan.

Pada kisaran tahun 1970an sampai awal 1980an, produk kecerdasan buatan mulai dikembangkan dalam bentuk sebuah produk. Pada tahun tersebut mulailah manusia memanfaatkan teknologi komputer yang berangsur-angsur dapat mempercepat aktifitas manusia. Pemasaran produk kecerdasan buatan di era awal membawa dampak yang signifikan bagi manusia, dari situ manusia mulai sadar bahwa perlu sebuah tool (alat) yang dapat menggantikan kemampuan akal manusia.

Komputer pada awalnya fokus untuk mengolah data mentah kemudian ditampilkan dalam bentuk interface sehingga dapat dipahami oleh manusia pada umumnya salah satunya untuk alat bantu mengambil sebuah keputusan.

Dengan adanya pemanfaatan kecerdasan buatan, yang diolah oleh komputer tidak sebatas data saja namun pengetahuan. Pengetahuan yang diolah oleh komputer dapat menghasilkan keputusan dan keluaran yang cukup akurat dan cepat.



Gambar 1.2. Diagram data dan Informasi

Sumber :

<https://donyprisma.wordpress.com/2011/10/21/data-informasi-dan-pengetahuan/>

Komputer yang mampu mengolah data menjadi sebuah informasi kemudian ditampilkan dalam interface (antarmuka) tentu seorang user (pengguna) masih mengkonversi data tersebut menjadi sebuah informasi kemudian dapat mengolah pengetahuan sehingga menghasilkan sebuah keputusan.

Berbeda saat komputer langsung mengolah pengetahuan, karena pengetahuan didalamnya sudah terdapat data dan informasi sehingga user (pengguna) tidak perlu melakukan konversi data dan informasi dan proses yang dikerjakan oleh manusia cenderung lebih cepat serta akurat jika dilihat dari hasilnya.

1.3 Bagian-bagian Kecerdasan Buatan

Terdapat beberapa penyebab kecerdasan buatan diciptakan oleh para ahli komputer serta ilmuwan. Secara umum beberapa penyebab diciptakannya kecerdasan buatan adalah sebagai berikut :

1. **Perkembangan teknologi informasi**

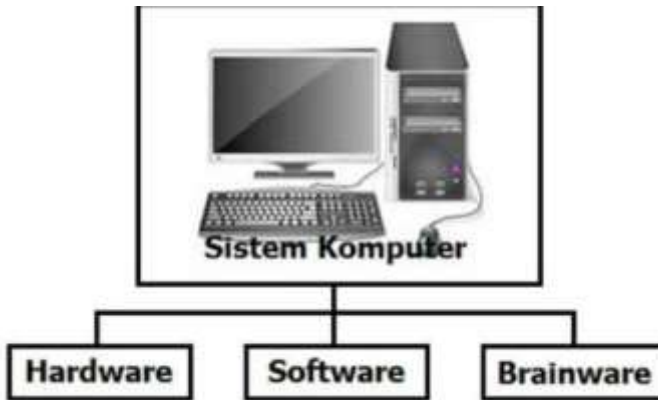
Teknologi informasi menjadi sebuah tren dan gaya hidup masyarakat. Dari pernyataan tersebut, dapat disimpulkan bahwa kehadiran teknologi informasi mempengaruhi kegiatan rutin manusia. Semakin cepatnya informasi diterima manusia tanpa melalui media cetak merupakan salah satu efek dari perkembangan teknologi informasi. Kelebihan sebuah teknologi informasi adalah pada pola penyimpanan data yang dapat diakses dimanapun berada.

2. **Perkembangan Software dan Hardware**

Sistem Komputer terdiri dari 3 komponen dasar yakni Brainware (manusia), Hardware (perangkat keras) dan Software (perangkat lunak). Manusia merupakan komponen utama dalam perkembangan sistem komputer. Manusia semakin berkembang pola pikir sehingga dapat menciptakan sebuah tool (alat) berupa hardware dan berkembang menjadi software (perangkat lunak) .

Hardware yang dulunya hanya sebatas mesin saat ini sudah dapat dikolaborasikan dengan software. Hardware bisa dikatakan sebagai rumah sedangkan software adalah

perabotan rumahnya. Semakin kompleksnya dua tool tersebut membuat sistem komputer semakin canggih dan dapat menjalankan pengetahuan yang berasal dari teknik kecerdasan buatan.



Gambar 1.3. Bagan Sistem Komputer
Sumber (Google.com)

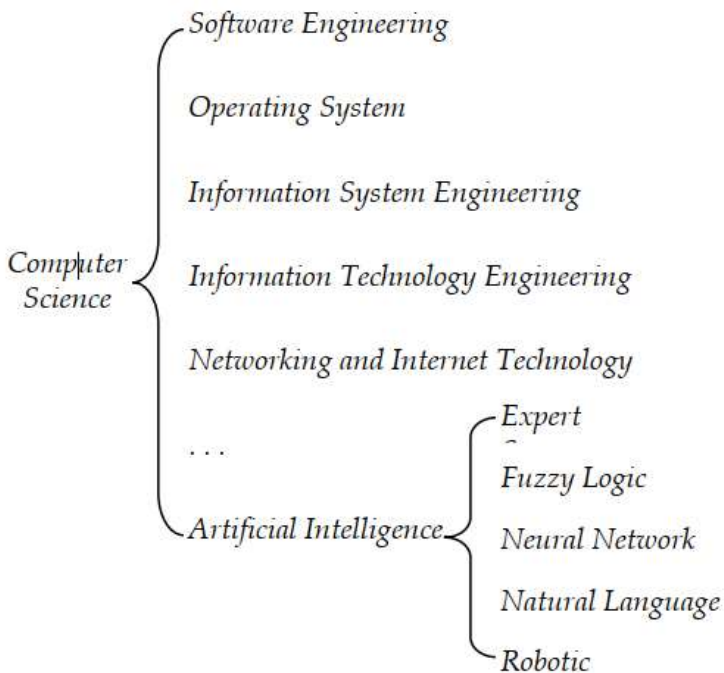
3. Teknologi sudah berkontribusi terhadap dunia ekonomi dan bisnis

Komunikasi yang dulu dapat dilakukan dengan cara *face to face* saat ini dapat dilakukan dengan hanya bermodalkan software dan hardware saja. Pernyataan ini berimbas pada kegiatan jual beli di dunia ekonomi. Marketing (pemasaran) yang biasa dilakukan dengan cara konvensional dapat dilakukan dengan bermodalkan tool berupa marketplace dan ecommerce.

Fokus pada kedua tool tersebut adalah image produk yang dijual. Penjual tidak perlu susah payah memasarkan menggunakan pamflet, banner dan lain-lain. Cukup dengan upload image produk dapat dilihat oleh konsumen dengan scop satu negara. Disisi lain dampak juga muncul yakni

persaingan antara penjual semakin ketat dan konsumen dapat melakukan perbandingan harga yang akurat dan cepat meskipun produknya sama.

Kecerdasan buatan merupakan bagian dari bidang ilmu komputer (*computer science*). Pada dasarnya kecerdasan buatan mentransformasikan kecerdasan otak manusia kedalam perangkat komputer.



Gambar 1.4. Digram Computer Science
Sumber (Buku Kecerdasan Buatan, Victor Amrizal)

Kecerdasan buatan dapat dihasilkan dari 3 komponen utama yakni struktur data yang dibuat secara runtut, algoritma yang disesuaikan dengan permasalahan sehingga mampu

menghasilkan pengetahuan dan bahasa pemrograman sebagai tahapan teknis akhir sesuai dengan bentuk software yang diinginkan (website, android, iOS dan lain-lain).

Pengetahuan sebagai bagian utama kecerdasan buatan merupakan beberapa wilayah subyek yang diperoleh melalui pendidikan dan pengalaman. Bagian utama yg dibutuhkan untuk aplikasi kecerdasan buatan adalah Basis pengetahuan (*knowledge base*) dan Motor inferensi (*inference engine*).

1.4 Perkembangan dan Efek Kecerdasan Buatan

Kecenderungan manusia dengan sebuah kemudahan mendorong berbagai dampak atau efek baik positif dan negatif. Salah satu kemudahan yang dinikmati manusia adalah hadirnya kecerdasan buatan. Salah satu efek negatif dari hadirnya kecerdasan buatan adalah penggunaan tenaga manusia yang minim sehingga lapangan pekerjaan akan semakin sedikit. Pada kenyataannya di lapangan, saat manusia yang melakukan pekerjaan pelayanan (*service*) kemudian melakukan kesalahan. Kesalahan yang dilakukan manusia masih dapat diperbaiki dengan dasar pikiran dan perasaan. Namun saat algoritma pada kecerdasan buatan yang melakukan kesalahan maka harus menghadirkan pihak developer untuk melakukan maintenance sehingga tool dapat berfungsi dengan baik seperti sedia kala. Efek positif jelas dapat diamati yakni semakin akurat dan cepatnya informasi yang diterima manusia.

DAFTAR PUSTAKA

- Amrizal, V. & Aini, Q., 2013, *KECERDASAN BUATAN*, Halaman Moeka Publishing, Jakarta.
- Ramadhan, R. & Eliyen, K., 2022, 'IMPLEMENTASI METODE TOPSIS PADA DECISION SUPPORT SYSTEM UNTUK PENILAIAN MAHASISWA BERBASIS PRESTASI AKADEMIK DAN NON AKADEMIK', *Rabit: Jurnal Teknologi dan Sistem Informasi Univrab*, 7(2).
- Ramadhan, R.F., 2023, *Implementasi Metode Simple Multi Attribute Rating Technique untuk Pemilihan Platform Jual Beli Berbasis Sistem Pendukung Keputusan*, Jakarta Selatan.
- Ramadhan, R.F. & Eliyen, K., 2022, 'Implementasi Metode Analytical Hierarchy Process pada Penilaian Mahasiswa Berprestasi Berbasis Decision Support System', *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 4(2), 98–105.
- Ramadhan, R.F. & Widodo, A.A., 2022, 'Penilaian Mahasiswa Berprestasi Menggunakan Metode Simple Additive Weighting Berbasis Decision Support System', *Jurnal Sistem Informasi dan Informatika (JUSIFOR)*, 1(2), 90–97.

BAB 2

MASALAH & RUANG MASALAH

Oleh Wahyuddin S

2.1 Pendahuluan

Kecerdasan Buatan atau *Artificial Intelligence* (AI) Digital merupakan salah satu teknologi yang semakin berkembang pesat dan memiliki banyak potensi untuk meningkatkan efisiensi dan efektivitas dalam berbagai bidang ilmu, seperti kesehatan, pendidikan, bisnis, dan lain sebagainya. Namun, di balik potensi keuntungan tersebut, ada juga masalah dan ruang masalah yang perlu diperhatikan dalam pengembangan, implementasi, dan penggunaan AI digital.

Masalah utama dalam AI digital adalah kepercayaan dan etika. Kita perlu memastikan bahwa teknologi AI dapat diandalkan dan aman bagi pengguna. Selain itu, ada juga masalah privasi, transparansi, akuntabilitas, dan bias dalam algoritma AI yang perlu diatasi agar pengguna dapat mempercayai dan memanfaatkan AI secara efektif.

Keterbatasan teknologi juga merupakan masalah utama dalam pengembangan AI digital. AI masih memiliki keterbatasan dalam pemahaman bahasa manusia, memahami konteks sosial, dan mengenali emosi. Selain itu, ketika AI digunakan dalam pengambilan keputusan kritis, persoalan tentang tanggung jawab dan akuntabilitas menjadi penting.

Dampak sosial dan ekonomi juga perlu diperhatikan dalam perkembangan AI digital. Dampak terhadap lapangan kerja, perubahan dalam paradigma bisnis, penggunaan AI dalam propaganda dan disinformasi, serta penguatan

kesenjangan digital dan ketidaksetaraan dapat menjadi masalah yang perlu diatasi.

Regulasi dan kebijakan yang tepat juga perlu ditempatkan untuk mengatur pengembangan, implementasi, dan penggunaan AI digital. Namun, pengaturan AI masih terus berkembang dan bisa menjadi tantangan dalam menghadapi perkembangan AI digital yang cepat.

Ketergantungan pada teknologi AI digital juga dapat menciptakan masalah. Kecenderungan untuk mengandalkan AI dalam pengambilan keputusan atau otomatisasi tugas dapat menciptakan ketergantungan pada teknologi, yang pada gilirannya dapat memengaruhi kemampuan manusia dalam memecahkan masalah secara mandiri.

Dalam rangka menghadapi masalah dan ruang masalah dalam AI digital, perlu dilakukan penelitian yang lebih lanjut dan kolaborasi antara para ahli, pengguna, dan pembuat kebijakan untuk mengembangkan teknologi AI yang lebih baik, lebih andal, dan lebih aman untuk digunakan (Dawis *et al.*, 2022).

2.2 Perkembangan Kecerdasan Digital

Perkembangan kecerdasan buatan (AI) digital telah mengalami kemajuan yang signifikan dalam beberapa tahun terakhir. Beberapa tren dan perkembangan terbaru dalam kecerdasan buatan digital antara lain:

1. *Deep Learning*

Deep learning atau pembelajaran dalam kedalaman adalah salah satu teknik dalam AI yang telah mengalami perkembangan pesat. Dengan menggunakan jaringan saraf tiruan yang kompleks, *deep learning* dapat mengenali pola yang sangat kompleks dalam data, seperti gambar, suara, atau teks. Hal ini telah memungkinkan pencapaian baru

dalam bidang seperti pengenalan gambar, pengenalan suara, dan pemrosesan bahasa alami.

2. *Internet of Things (IoT) dan AI*

Kombinasi antara IoT dan AI telah memungkinkan pengembangan sistem yang cerdas dan terhubung secara digital. Data yang dihasilkan oleh perangkat IoT dapat diolah oleh algoritma AI untuk mengambil keputusan secara otomatis dan mengoptimalkan kinerja sistem. Contoh penggunaan AI dalam IoT adalah smart city, smart home, dan smart manufacturing.

3. Pembelajaran Mesin (*Machine Learning*) yang Terdistribusi

Machine learning yang terdistribusi adalah pendekatan di mana model AI ditempatkan pada berbagai perangkat atau node di jaringan yang saling terhubung, seperti edge computing, cloud computing, atau sistem terdistribusi lainnya. Pendekatan ini memungkinkan pengolahan data yang lebih cepat dan responsif, serta mengurangi ketergantungan pada sumber daya pusat (Permana et al., 2023).

4. Pengolahan Bahasa Alami (*Natural Language Processing/NLP*)

Kemajuan dalam pengolahan bahasa alami telah memungkinkan AI untuk memahami dan menghasilkan teks manusia dengan lebih akurat. Penggunaan NLP telah diterapkan dalam berbagai aplikasi, seperti chatbot, analisis sentimen, pemrosesan teks otomatis, dan penerjemahan bahasa.

5. AutoML dan AI Ops

AutoML atau otomatisasi pembelajaran mesin, serta AI Ops yang merupakan gabungan antara AI dan operasi TI, telah memungkinkan pengguna yang tidak memiliki latar belakang teknis yang mendalam untuk mengimplementasikan dan mengelola model AI dengan

lebih mudah. Hal ini dapat mempercepat adopsi AI dalam berbagai industri dan sektor.

6. Eksplanabilitas dan Transparansi AI

Kepentingan untuk memahami bagaimana model AI membuat keputusan dan menjelaskan proses yang ada di baliknya semakin meningkat. Oleh karena itu, perkembangan AI telah mengarah pada upaya untuk membuat model AI yang lebih eksplisit, interpretable, dan transparan untuk pengguna dan pemangku kepentingan lainnya.

7. Kombinasi AI dengan Teknologi Lain

Kecerdasan buatan juga semakin terintegrasi dengan teknologi lain, seperti *augmented reality* (AR), *virtual reality* (VR), blockchain, dan komputasi kuantum. Kombinasi ini membuka potensi baru dalam pengembangan aplikasi AI yang lebih kompleks dan canggih (Muttaqin, Yuswardi, *et al.*, 2023).

2.3 Masalah Umum Kecerdasan Buatan

Masalah dan Ruang Masalah dalam Kecerdasan Buatan (AI) Digital adalah area yang mencakup berbagai tantangan dan keterbatasan dalam pengembangan, implementasi, dan penggunaan AI dalam konteks digital. Beberapa masalah utama dan ruang masalah dalam AI digital antara lain:

2.3.1 Kepercayaan dan Etika

Salah satu masalah utama dalam AI digital adalah kepercayaan dan etika. Keberlanjutan kecerdasan buatan digital membutuhkan kepercayaan pengguna terhadap teknologi tersebut. Masalah privasi, transparansi, akuntabilitas, dan bias dalam algoritma AI dapat menjadi isu yang mempengaruhi kepercayaan pengguna terhadap AI digital. Etika dalam penggunaan AI, termasuk kebijakan, regulasi, dan

pertanggungjawaban penggunaan AI, juga menjadi perhatian utama dalam menghadapi perkembangan AI digital.

2.3.2 Keamanan dan Perlindungan Data

Keberlanjutan AI digital membutuhkan perlindungan data yang adekuat. Data yang digunakan dalam AI digital harus dijaga keamanannya agar tidak jatuh ke tangan yang salah atau digunakan secara salah. Serangan terhadap sistem AI, manipulasi data, dan ancaman keamanan siber dapat merusak integritas dan kehandalan AI digital.

2.3.3 Keterbatasan Teknologi

Meskipun kemajuan pesat dalam AI digital, masih ada keterbatasan teknologi yang perlu diatasi. Salah satu tantangan utama adalah dalam pengenalan pola dan pemahaman konteks yang kompleks. Kemampuan AI digital dalam memahami bahasa manusia, memahami konteks sosial, dan mengenali emosi masih terbatas dan perlu ditingkatkan.

2.3.4 Pengambilan Keputusan dan Tanggung Jawab

Ketika AI digunakan dalam pengambilan keputusan kritis, seperti dalam pengambilan keputusan medis, kebijakan publik, atau keputusan bisnis, persoalan tentang tanggung jawab dan akuntabilitas menjadi penting. Siapa yang bertanggung jawab jika sistem AI membuat kesalahan atau membuat keputusan yang kontroversial? Bagaimana AI harus diberdayakan untuk mengambil keputusan yang benar, transparan, dan adil?.

2.3.5 Pengaruh Sosial dan Ekonomi

Perkembangan AI digital juga memiliki implikasi sosial dan ekonomi yang signifikan. Dampak terhadap lapangan kerja, perubahan dalam paradigma bisnis, penggunaan AI dalam

propaganda dan disinformasi, serta penguatan kesenjangan digital dan ketidaksetaraan dapat menjadi masalah yang perlu diperhatikan dalam konteks AI digital.

2.3.6 Regulasi dan Kebijakan

Regulasi dan kebijakan yang tepat perlu ditempatkan untuk mengatur pengembangan, implementasi, dan penggunaan AI digital. Namun, pengaturan AI masih terus berkembang dan bisa menjadi tantangan dalam menghadapi perkembangan AI digital yang cepat. Terdapat tantangan dalam mengatur dan mengawasi penggunaan kecerdasan buatan digital. Perlu adanya regulasi dan kebijakan yang diperbarui untuk menghadapi perkembangan teknologi AI yang cepat, untuk memastikan penggunaan yang bertanggung jawab, aman, dan adil.

2.3.7 Bias dan Diskriminasi

Model AI dapat menyerap bias dari data yang digunakan dalam pelatihannya, yang dapat menghasilkan keputusan yang diskriminatif atau tidak adil. Bias ini dapat termanifestasi dalam berbagai bentuk, seperti gender, ras, agama, atau orientasi seksual, dan dapat mengakibatkan ketidakadilan dalam pengambilan keputusan atau tindakan yang diambil oleh AI.

2.3.8 Pengangguran dan Disrupsi Pasar Kerja

Implementasi AI dapat menggantikan pekerjaan manusia dalam beberapa sektor, yang dapat mengakibatkan pengangguran dan disrupsi pasar kerja. Dibutuhkan perencanaan dan persiapan yang baik untuk mengatasi dampak sosial dan ekonomi dari penggunaan kecerdasan buatan dalam dunia kerja.

2.4 Peluang & Ancaman AI

Kecerdasan Buatan Digital (AI) memiliki peluang dan ancaman yang berbeda dalam berbagai sektor. Berikut adalah beberapa peluang dan ancaman utama kecerdasan buatan digital:

2.4.1 Peluang Kecerdasan Buatan Digital (AI)

1. Peningkatan efisiensi dan produktivitas

AI dapat membantu meningkatkan efisiensi dan produktivitas di berbagai sektor, seperti industri, kesehatan, dan transportasi, dengan melakukan tugas-tugas yang dulunya membutuhkan waktu dan usaha manusia yang besar.

2. Perbaikan kualitas hidup

AI dapat membantu meningkatkan kualitas hidup melalui aplikasi dalam kesehatan, pendidikan, dan lingkungan. Contohnya, AI dapat membantu mendeteksi penyakit secara dini, meningkatkan kualitas pengajaran di sekolah dan universitas, dan membantu menganalisis dan memprediksi dampak lingkungan.

3. Pengembangan teknologi baru

Pengembangan AI telah mendorong pengembangan teknologi baru yang dapat membantu meningkatkan efisiensi dan efektivitas di berbagai sektor, seperti robotik, *Internet of Things* (IoT), dan kendaraan otonom.

4. Penghematan biaya dan waktu

AI dapat membantu menghemat biaya dan waktu di berbagai sektor, seperti perbankan, keuangan, dan manufaktur. Contohnya, AI dapat membantu melakukan pemrosesan transaksi secara otomatis, mengurangi kebutuhan tenaga kerja manusia (Wahyuddin, Rismayani, *et al.*, 2023).

2.4.2 Ancaman Kecerdasan Buatan Digital (AI)

Penggantian tenaga kerja manusia: Pengembangan AI dapat menggantikan pekerja manusia dalam berbagai sektor, yang dapat mengakibatkan hilangnya lapangan pekerjaan bagi manusia.

1. Ketidakadilan dalam pengambilan keputusan
AI dapat menyerap bias dari data yang digunakan dalam pelatihan model, yang dapat menghasilkan keputusan yang diskriminatif atau tidak adil. Hal ini dapat menyebabkan ketidakadilan dalam pengambilan keputusan atau tindakan yang diambil oleh AI.
2. Penyalahgunaan dan ancaman keamanan
Penggunaan AI yang tidak etis dapat menyebabkan penyalahgunaan dan ancaman keamanan. Contohnya, penggunaan AI dalam sistem pemantauan dapat mengancam privasi dan keamanan individu.
3. Ketergantungan pada teknologi
Penggunaan AI yang terlalu berlebihan dan ketergantungan pada teknologi dapat menyebabkan kehilangan kemampuan manusia untuk menyelesaikan tugas secara mandiri, serta mengurangi kemampuan manusia untuk berpikir kritis dan kreatif.
4. Regulasi dan kebijakan yang tidak memadai
Tidak adanya regulasi dan kebijakan yang memadai dalam penggunaan AI dapat menyebabkan penyalahgunaan dan dampak negatif pada masyarakat dan lingkungan.
5. Keterbatasan teknologi AI
Teknologi AI masih memiliki keterbatasan dalam hal kemampuan untuk memahami konteks, situasi, dan kompleksitas masalah yang dihadapi. AI masih belum dapat menggantikan sepenuhnya kemampuan manusia dalam hal kreativitas, penalaran moral, dan kepemimpinan (Muttaqin, Jaya, *et al.*, 2023).

2.5 Tantangan Kecerdasan Buatan Digital (AI)

Ada beberapa tantangan yang dihadapi dalam pengembangan dan implementasi kecerdasan buatan digital, atau AI, di berbagai sektor. Berikut adalah beberapa tantangan utama dalam pengembangan dan penerapan AI:

1. Data yang Berkualitas dan Ketersediaan Data

Pengembangan AI memerlukan data yang berkualitas dan ketersediaan data yang memadai. Data yang digunakan harus representatif dan dapat dipercaya, dan harus cukup besar dan bervariasi untuk melatih model AI. Namun, seringkali sulit untuk memperoleh data yang memenuhi kriteria ini, terutama dalam sektor-sektor yang masih tergolong baru dalam penerapan AI.

2. Kapasitas Komputasi

AI memerlukan kapasitas komputasi yang besar untuk memproses data dengan cepat dan akurat. Meskipun teknologi komputasi semakin maju, namun untuk mengembangkan model AI yang kompleks dan akurat masih memerlukan sumber daya komputasi yang besar.

3. Keterbatasan Teknologi AI

Teknologi AI masih memiliki keterbatasan dalam hal kemampuan untuk memahami konteks, situasi, dan kompleksitas masalah yang dihadapi. AI masih belum dapat menggantikan sepenuhnya kemampuan manusia dalam hal kreativitas, penalaran moral, dan kepemimpinan.

4. Bias dalam Model AI

Model AI dapat menyerap bias dari data yang digunakan dalam pelatihan model, yang dapat menghasilkan keputusan yang diskriminatif atau tidak adil. Bias ini dapat termanifestasi dalam berbagai bentuk, seperti gender, ras, agama, atau orientasi seksual, dan dapat mengakibatkan ketidakadilan dalam pengambilan keputusan atau tindakan yang diambil oleh AI.

5. Privasi dan Keamanan

Penggunaan data yang melibatkan informasi pribadi dalam konteks AI dapat menimbulkan masalah privasi dan keamanan. Pengumpulan, pengolahan, dan penyimpanan data yang digunakan dalam model AI harus dilakukan dengan memperhatikan kerahasiaan dan integritas data, serta mengikuti peraturan dan regulasi yang berlaku.

6. Akuntabilitas dan Tanggung Jawab

Masalah akuntabilitas dan tanggung jawab dalam penggunaan kecerdasan buatan digital masih menjadi perhatian utama. Ketika kesalahan atau kerugian terjadi akibat penggunaan AI, sulit untuk menentukan siapa yang bertanggung jawab, baik itu pengembang, operator, atau pengguna akhir. Diperlukan kerangka kerja etis dan hukum yang jelas untuk mengatur aspek ini.

7. Perubahan Sosial dan Ekonomi

Penggunaan AI dapat mengubah cara kerja dan pasar kerja, serta mempengaruhi aspek-aspek sosial dan ekonomi. Ini dapat menghasilkan tantangan dalam mengatasi dampak sosial dan ekonomi dari penggunaan kecerdasan buatan dalam dunia kerja dan masyarakat.

8. Kebijakan dan Regulasi

Tantangan terakhir adalah pengaturan dan pengawasan penggunaan kecerdasan buatan digital. Pengaturan dan kebijakan yang berkaitan dengan penggunaan AI masih terus berkembang dan belum selalu konsisten di berbagai negara dan yurisdiksi. Peraturan yang jelas dan konsisten diperlukan untuk mengatur penggunaan AI secara bertanggung jawab, termasuk dalam hal privasi, keamanan, etika, dan akuntabilitas (Wahyuddin, Sudipa, *et al.*, 2023).

2.6 Hukum Kecerdasan Buatan Digital (AI)

Hukum tentang kecerdasan buatan (*Artificial Intelligence/AI*) masih terus berkembang dan berbeda di setiap yurisdiksi. Beberapa hukum yang berkaitan dengan AI antara lain:

1. Regulasi Privasi Data

Hukum yang mengatur perlindungan data pribadi, seperti *General Data Protection Regulation* (GDPR) di Uni Eropa, memiliki ketentuan-ketentuan yang berhubungan dengan penggunaan data dalam konteks AI. GDPR misalnya, mengatur bagaimana data pribadi dapat dikumpulkan, diproses, dan digunakan dalam pengembangan dan penggunaan AI.

2. Hukum Properti Intelektual

Hukum yang mengatur hak kekayaan intelektual, seperti paten, merek, dan hak cipta, dapat menjadi relevan dalam konteks AI. Hal ini terkait dengan perlindungan hukum terhadap teknologi dan inovasi yang terkait dengan AI, seperti algoritma atau model yang dihasilkan oleh AI.

3. Regulasi Keamanan dan *Cybersecurity*

Hukum yang mengatur keamanan dan kebijakan *cybersecurity* dapat berdampak pada pengembangan dan penggunaan AI. Regulasi ini dapat mencakup persyaratan keamanan untuk sistem AI, perlindungan terhadap serangan cyber, dan penggunaan AI dalam kegiatan yang berkaitan dengan keamanan nasional atau kebijakan publik.

4. Etika dan Kode Etik AI

Beberapa yurisdiksi telah mengeluarkan panduan atau kode etik khusus yang berkaitan dengan penggunaan AI secara etis. Hal ini meliputi prinsip-prinsip dan panduan tentang kebijakan etis, transparansi, akuntabilitas, dan

pengambilan keputusan yang bertanggung jawab dalam pengembangan dan penggunaan AI.

5. Regulasi Industri dan Profesional

Beberapa sektor industri, seperti kendaraan otonom atau pengobatan berbasis AI, mungkin memiliki regulasi khusus yang mengatur penggunaan AI dalam sektor tersebut. Regulasi ini dapat mencakup persyaratan sertifikasi, lisensi, dan penggunaan yang aman dalam lingkungan industri tertentu.

6. Hukum Kepribadian dan Tanggung Jawab

Pertanyaan hukum tentang kepribadian dan tanggung jawab AI menjadi isu yang semakin relevan. Misalnya, pertanyaan tentang siapa yang bertanggung jawab jika terjadi kerugian atau kerusakan akibat tindakan atau keputusan yang diambil oleh AI tanpa campur tangan manusia.

7. Regulasi Pengambilan Keputusan Otomatis

Beberapa yurisdiksi memiliki regulasi khusus yang mengatur pengambilan keputusan otomatis oleh sistem AI, terutama dalam konteks yang memiliki dampak signifikan terhadap hak, kewajiban, atau kepentingan individu.

Namun, perlu dicatat bahwa regulasi AI masih dalam tahap perkembangan dan belum konsisten di seluruh dunia. Perubahan dan penyesuaian regulasi terhadap AI terus berlangsung untuk menghadapi tantangan dan perubahan yang cepat dalam teknologi ini (Wibowo *et al.*, 2023).

DAFTAR PUSTAKA

- Dawis, A. M., Himawan, I. S., Meidelfi, D., Ikhrum, F., Intan, I., Harun, R., Haris, M. S., Wahyuddin, S., Yuniar, E. & Purnomo, R. 2022. *Artificial Intelligence: Konsep Dasar Dan Kajian Praktis*. TOHAR MEDIA.
- Muttaqin, M., Jaya, A. K., Harlina, S., Wahyuddin, S., Hakim, L., Anshori, M., Ambarwari, A., Kaunang, F. J., Sandag, G. A. & Harizahayu, H. 2023. *Data Science dan Pembelajaran Mesin*. Yayasan Kita Menulis.
- Muttaqin, M., Yuswardi, Y., Parewe, A. M. A. K., Ashari, I. F., Munsarif, M., Wahyuddin, S., Pasnur, P., Rahmah, S. A., Defiariyany, D. & Gustiana, Z. 2023. *Pengantar Sistem Cerdas*. Yayasan Kita Menulis.
- Permana, A. A., Wahyuddin, S., Santoso, L. W., Wibowo, G. W. N., Wardhani, A. K., Wahidin, A. J., Yuliastuti, G. E. & Wijayanti, R. R. 2023. *Machine Learning*. Global Eksekutif Teknologi.
- Wahyuddin, S., Rismayani, R., Sihotang, J. I., Aisa, S., Gunawan, H., Tamsir, N., Masturoh, S., Radiyah, U., Gustiana, Z. & Harlina, S. 2023. *Data Warehouse dan Data Mining*. Yayasan Kita Menulis.
- Wahyuddin, S., Sudipa, I. G. I., Putra, T. A. E., Wahidin, A. J., Syukrilla, W. A., Wardhani, A. K., Heryana, N., Indriyani, T. & Santoso, L. W. 2023. *Data Mining*. Global Eksekutif Teknologi.
- Wibowo, S. H., Wahyuddin, S., Permana, A. A., Sembiring, S., Wahidin, A. J., Nugroho, J. W., Rahajeng, E., Kurnaedi, D., Bau, R. T. R. L. & Adhicandra, I. 2023. *Teknologi Digital Di Era Modern*. Global Eksekutif Teknologi.

BAB 3

SISTEM PRODUKSI

Oleh Fahmy Rinanda Saputri

3.1 Pendahuluan

Sistem produksi pada kecerdasan buatan (Artificial Intelligence atau AI) adalah sebuah pendekatan dalam pembuatan sistem AI yang bertujuan untuk membangun sistem yang dapat menghasilkan output atau tindakan berdasarkan pada input atau masukan yang diterimanya. Sistem produksi pada AI berfungsi sebagai kerangka kerja atau algoritma yang mengatur bagaimana suatu sistem AI bekerja, termasuk bagaimana sistem tersebut mengumpulkan, memproses, dan merespons informasi.

Pada dasarnya, sistem produksi pada AI terdiri dari sejumlah aturan atau strategi yang diprogramkan ke dalam sistem. Aturan-aturan ini membantu sistem AI untuk memahami masukan yang diterimanya dan memberikan keluaran atau tindakan yang sesuai. Dalam beberapa kasus, aturan-aturan ini juga dapat diprogramkan untuk mengambil keputusan dan bertindak secara mandiri tanpa intervensi manusia.

Sistem produksi pada AI dapat diterapkan dalam berbagai bidang, seperti sistem deteksi kecurangan, sistem pemberian rekomendasi, dan sistem pengenalan suara atau gambar. Dalam pengembangan sistem produksi pada AI, para pengembang harus mempertimbangkan berbagai faktor, seperti akurasi, efisiensi, dan skalabilitas sistem, serta keamanan dan privasi data yang digunakan oleh sistem tersebut.

Untuk mengembangkan sistem produksi pada AI, para pengembang juga dapat menggunakan berbagai teknik dan algoritma yang sesuai dengan masalah yang ingin dipecahkan. Beberapa teknik dan algoritma yang umum digunakan dalam sistem produksi pada AI antara lain:

1. Sistem berbasis aturan (*rule-based systems*): sistem ini memanfaatkan sejumlah aturan untuk mengambil keputusan atau tindakan. Aturan-aturan ini biasanya ditentukan oleh para ahli di bidang yang relevan dan diprogram ke dalam sistem. Contoh penggunaannya adalah pada sistem deteksi kecurangan pada perbankan, dimana aturan-aturan digunakan untuk mendeteksi aktivitas transaksi yang mencurigakan.
2. Sistem berbasis pengetahuan (*knowledge-based systems*): sistem ini menggunakan basis pengetahuan atau knowledge base untuk membuat keputusan atau tindakan. Basis pengetahuan ini dapat diisi dengan informasi yang diperoleh dari berbagai sumber, seperti data empiris atau pengetahuan yang dimiliki oleh ahli di bidang tertentu. Contoh penggunaannya adalah pada sistem diagnosis medis, dimana basis pengetahuan berisi informasi mengenai gejala, penyakit, dan pengobatan.
3. Sistem berbasis logika (*logic-based systems*): sistem ini menggunakan algoritma logika untuk mengambil keputusan atau tindakan. Algoritma logika ini biasanya didasarkan pada prinsip-prinsip matematika dan logika formal. Contoh penggunaannya adalah pada sistem kontrol lalu lintas, dimana algoritma logika digunakan untuk mengoptimalkan arus lalu lintas.
4. Sistem berbasis jaringan saraf tiruan (*artificial neural network systems*): sistem ini menggunakan model matematis yang terinspirasi dari jaringan saraf biologis untuk mengambil keputusan atau tindakan. Model ini

- terdiri dari sejumlah "neuron" atau unit pemrosesan yang terhubung dalam jaringan. Contoh penggunaannya adalah pada sistem pengenalan suara atau gambar, dimana model jaringan saraf digunakan untuk mempelajari pola-pola yang ada dalam suara atau gambar.
5. Sistem berbasis pembelajaran mesin (*machine learning systems*): sistem ini menggunakan algoritma pembelajaran mesin untuk mempelajari pola-pola dari data masukan dan menghasilkan keluaran atau tindakan yang sesuai. Algoritma pembelajaran mesin ini dapat dibagi menjadi beberapa jenis, seperti *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Contoh penggunaannya adalah pada sistem pemberian rekomendasi, dimana algoritma supervised learning digunakan untuk mempelajari preferensi pengguna dan memberikan rekomendasi yang sesuai.

Secara keseluruhan, sistem produksi pada AI dapat membantu meningkatkan efisiensi dan akurasi dalam berbagai bidang, dari manufaktur hingga layanan kesehatan. Namun, seperti halnya dengan teknologi apapun, sistem produksi pada AI juga memiliki risiko dan tantangan, seperti masalah etika, keamanan dan privasi data, serta bias algoritma. Oleh karena itu, para pengembang harus mempertimbangkan dengan matang semua faktor dan membangun sistem AI yang aman, adil, dan transparan.

3.2 Konsep Sistem Produksi

Konsep sistem produksi adalah sebuah model atau kerangka kerja yang digunakan dalam manufaktur dan produksi untuk mengoptimalkan penggunaan sumber daya dan meningkatkan efisiensi produksi. Konsep ini mencakup

berbagai elemen, seperti perencanaan produksi, pengendalian kualitas, manajemen persediaan, dan pengembangan produk. Konsep sistem produksi juga dapat diterapkan pada pengembangan kecerdasan buatan (artificial intelligence/AI). Pada dasarnya, sistem produksi pada AI melibatkan pengumpulan data, analisis data, pembelajaran mesin (machine learning), pengujian dan validasi, serta pengembangan dan peningkatan terus-menerus.

Berikut adalah konsep sistem produksi pada kecerdasan buatan:

1. Pengumpulan data: Tahap ini melibatkan pengumpulan data yang diperlukan untuk mengembangkan sistem AI. Data yang dikumpulkan harus akurat, relevan, dan cukup besar untuk dapat dilakukan analisis data dan pembelajaran mesin.
2. Analisis data: Setelah data terkumpul, tahap selanjutnya adalah melakukan analisis data. Pada tahap ini, data dianalisis untuk mengidentifikasi pola, tren, dan relasi antara data. Analisis data ini kemudian menjadi dasar untuk mengembangkan model AI.
3. Pembelajaran mesin: Pada tahap ini, model AI dikembangkan dengan menggunakan algoritma pembelajaran mesin. Model ini diuji dan disesuaikan dengan data yang ada untuk meningkatkan akurasi dan kinerja sistem AI.
4. Pengujian dan validasi: Setelah model AI dikembangkan, tahap selanjutnya adalah melakukan pengujian dan validasi sistem AI. Hal ini bertujuan untuk memastikan bahwa sistem AI dapat bekerja dengan baik dan memenuhi standar kualitas yang ditetapkan.
5. Peningkatan terus-menerus: Tahap terakhir dalam konsep sistem produksi pada AI adalah melakukan peningkatan terus-menerus pada sistem AI. Peningkatan ini dilakukan

dengan melakukan evaluasi kinerja sistem AI, mengumpulkan dan menganalisis data baru, dan melakukan pembelajaran mesin untuk meningkatkan kinerja dan akurasi sistem AI.

Dalam pengembangan sistem AI, konsep sistem produksi dapat membantu dalam memastikan bahwa sistem AI yang dihasilkan efisien, akurat, dan dapat beradaptasi dengan cepat terhadap perubahan lingkungan dan permintaan pasar. Dalam hal ini, prinsip-prinsip seperti penggunaan algoritma yang efisien, pengendalian kualitas, dan peningkatan terus-menerus sangat penting untuk diterapkan pada sistem AI. Selain itu, penggunaan sistem produksi pada AI juga harus memperhatikan aspek etika dan keamanan data untuk memastikan bahwa sistem AI yang dihasilkan dapat diterima oleh masyarakat dan tidak membahayakan privasi dan keamanan data.

3.3 Komponen Sistem Produksi

Dalam sistem produksi kecerdasan buatan digital, terdapat tiga komponen utama yang berperan dalam pengambilan keputusan atau pemecahan masalah, yaitu struktur kendali, basis pengetahuan, dan basis data (Ritha, 2020).

3.2.1 Basis Pengetahuan

Basis pengetahuan atau knowledge base merupakan komponen dalam sistem produksi yang berisi aturan-aturan atau pengetahuan yang diperoleh dari pakar atau pengalaman dalam suatu domain tertentu. Basis pengetahuan digunakan oleh sistem untuk membuat keputusan atau memecahkan masalah dengan cara mencocokkan informasi yang diberikan dengan aturan yang ada di dalam basis pengetahuan. Basis pengetahuan biasanya disusun dalam bentuk aturan-aturan

logika yang dapat digunakan oleh mesin untuk mengambil keputusan.

Setiap aturan terdiri dari dua bagian yaitu kondisi (*condition*) dan aksi (*action*). Kondisi merupakan serangkaian syarat yang harus dipenuhi agar aturan dapat dijalankan, sedangkan aksi merupakan tindakan yang harus dilakukan jika kondisi terpenuhi.

IF-THEN adalah salah satu bentuk komponen aturan atau rule dalam sistem produksi kecerdasan buatan digital. IF-THEN dapat digunakan sebagai bentuk dasar dari aturan-aturan dalam basis pengetahuan yang digunakan oleh sistem dalam pengambilan keputusan atau pemecahan masalah.

Bentuk dasar IF-THEN adalah sebagai berikut:

"IF kondisi tertentu terpenuhi THEN lakukan tindakan tertentu."

Contoh aturan IF-THEN dalam sistem produksi mungkin seperti:

"IF suhu ruangan lebih dari 25 derajat Celsius THEN hidupkan AC."

"IF ketinggian air di waduk lebih rendah dari batas minimum THEN hidupkan pompa untuk menambah air."

"IF nilai saham perusahaan A turun di bawah ambang batas minimum THEN jual saham perusahaan A."

Dalam contoh-contoh di atas, kondisi IF mewakili situasi atau keadaan yang harus terpenuhi agar tindakan tertentu dalam bagian THEN dilakukan. Aturan-aturan semacam ini dapat disimpan dalam basis pengetahuan dan digunakan oleh sistem produksi untuk membuat keputusan atau memecahkan masalah dalam waktu nyata.

IF-THEN merupakan bentuk aturan yang sederhana dan mudah dimengerti oleh manusia dan mesin. Namun, aturan-aturan yang lebih kompleks dapat dibangun dengan menggunakan bentuk IF-THEN yang lebih kompleks atau

kombinasi dari beberapa aturan IF-THEN yang lebih sederhana.

IF-THEN atau dapat diganti dengan JIKA-MAKA seperti pada contoh berikut.

Dalam sistem produksi untuk memprediksi kualitas produk, aturan dapat dituliskan sebagai berikut:

“JIKA suhu ruangan rendah dan kelembaban tinggi, MAKA kualitas produk akan rendah.”

Aturan ini menyatakan bahwa jika suhu ruangan rendah dan kelembaban tinggi, maka kualitas produk akan rendah. Dalam hal ini, suhu ruangan rendah dan kelembaban tinggi adalah kondisi, sedangkan kualitas produk yang rendah adalah aksi.

Dalam sistem produksi, aturan dapat dijalankan secara berurutan atau sekaligus. Jika aturan dijalankan secara berurutan, maka aturan pertama akan dievaluasi terlebih dahulu dan jika kondisi terpenuhi, maka aksi akan dijalankan dan aturan berikutnya akan dievaluasi. Jika kondisi pada aturan pertama tidak terpenuhi, maka aturan kedua akan dievaluasi, demikian seterusnya hingga ditemukan aturan yang sesuai.

Sedangkan jika aturan dijalankan sekaligus, maka semua aturan akan dievaluasi secara simultan dan aksi dari aturan yang terpenuhi akan dijalankan. Keuntungan dari penggunaan aturan dalam sistem produksi adalah dapat meningkatkan efisiensi dan konsistensi dalam pengambilan keputusan, serta memudahkan pengembangan dan pemeliharaan sistem produksi.

3.2.2 Basis Data dan Working Memory

Basis data dan working memory adalah dua hal yang berbeda dalam sistem produksi kecerdasan buatan digital. Basis data adalah sebuah tempat penyimpanan yang digunakan untuk menyimpan informasi yang dibutuhkan oleh sistem

untuk pengambilan keputusan atau pemecahan masalah. Sementara itu, *working memory* adalah suatu bentuk memori sementara yang digunakan oleh sistem untuk menyimpan informasi yang sedang digunakan dalam proses pengambilan keputusan atau pemecahan masalah.

Basis data merupakan sumber informasi utama dalam sistem produksi dan biasanya berisi informasi tentang fakta, angka, atau karakteristik dari suatu masalah atau situasi yang sedang dihadapi. Basis data dapat diakses oleh sistem secara langsung dan informasi di dalamnya dapat digunakan untuk mengambil keputusan atau memecahkan masalah.

Working memory, di sisi lain, berfungsi sebagai area kerja sementara di mana sistem dapat menyimpan informasi yang sedang diproses. *Working memory* sering disebut sebagai memori jangka pendek dalam sistem produksi. *Working memory* digunakan untuk menyimpan informasi seperti data input, hasil proses intermediate, dan informasi lainnya yang diperlukan untuk pengambilan keputusan atau pemecahan masalah. *Working memory* biasanya dibatasi oleh kapasitas yang terbatas dan informasi di dalamnya biasanya dihapus setelah digunakan atau ketika tidak lagi dibutuhkan.

Dalam beberapa sistem produksi kecerdasan buatan, basis data dan *working memory* dapat digabungkan menjadi satu entitas yang disebut dengan *knowledge base*. *Knowledge base* adalah kumpulan informasi yang mencakup informasi dari basis data dan *working memory* serta aturan-aturan atau pengetahuan dari basis pengetahuan yang digunakan oleh sistem dalam pengambilan keputusan atau pemecahan masalah. *Knowledge base* dapat diakses oleh sistem secara langsung dan digunakan untuk mengambil keputusan atau memecahkan masalah dengan cepat dan efektif.

3.2.3 Sistem Kendali

Dalam sistem produksi kecerdasan buatan digital, struktur kendali adalah cara untuk mengatur aturan-aturan dalam basis pengetahuan atau knowledge base. Struktur kendali ini biasanya terdiri dari beberapa komponen, seperti *antecedent*, *consequent*, *rule weight*, dan konflik set.

Antecedent adalah bagian aturan yang berisi kondisi atau input yang harus terpenuhi agar aturan tersebut bisa diterapkan. Sedangkan *consequent* adalah bagian aturan yang berisi tindakan atau output yang akan dihasilkan jika kondisi terpenuhi.

Rule weight adalah bobot atau nilai penting dari sebuah aturan dalam basis pengetahuan. Bobot ini dapat digunakan untuk menentukan urutan aturan yang akan dievaluasi atau diaplikasikan.

Konflik set adalah kumpulan aturan yang memenuhi kondisi pada suatu waktu tertentu. Konflik set ini digunakan untuk menentukan urutan evaluasi aturan, sehingga dapat meminimalkan jumlah konflik yang terjadi.

Beberapa metode yang dapat digunakan untuk menentukan urutan evaluasi aturan dalam konflik set adalah sebagai berikut:

1. *Metode ranking*: Metode ini mengurutkan aturan berdasarkan rule weight, sehingga aturan dengan bobot tertinggi dievaluasi terlebih dahulu.
2. *Longest matching*: Metode ini mencari aturan yang memiliki *antecedent* paling panjang dan dievaluasi terlebih dahulu.
3. *Priority*: Metode ini menentukan urutan evaluasi aturan berdasarkan prioritas yang telah ditentukan sebelumnya. Aturan dengan prioritas tertinggi akan dievaluasi terlebih dahulu.

4. *Recently used*: Metode ini mengevaluasi aturan yang baru saja digunakan atau dipanggil. Aturan yang baru saja digunakan akan dievaluasi terlebih dahulu.

Setiap metode memiliki kelebihan dan kekurangan tertentu, dan pemilihan metode tergantung pada karakteristik dan kebutuhan sistem produksi kecerdasan buatan digital yang digunakan.

3.4 Prosedur Kerja Sistem Produksi

Sistem produksi dalam kecerdasan buatan biasanya melibatkan beberapa tahapan proses, seperti berikut:

1. **Pengumpulan Data**: Proses ini melibatkan pengumpulan data dari berbagai sumber seperti sensor, perangkat lunak, atau database. Data tersebut nantinya akan digunakan sebagai bahan input untuk proses berikutnya.
2. **Preprocessing Data**: Data yang telah terkumpul akan diproses agar bisa dipahami oleh sistem. Proses preprocessing melibatkan pemrosesan data mentah menjadi format yang dapat diproses oleh sistem. Contohnya, jika sistem akan menggunakan data gambar, maka data mentah akan diubah menjadi format yang bisa diolah seperti JPEG atau PNG.
3. **Pemodelan Data**: Setelah data diproses, selanjutnya dilakukan proses pemodelan data dengan menggunakan teknik-teknik statistik dan matematika. Proses ini bertujuan untuk mengidentifikasi pola dalam data dan membuat model prediksi yang akurat.
4. **Pelatihan Model**: Setelah model dibuat, maka model tersebut akan dilatih menggunakan data latih. Proses pelatihan model ini akan mengoptimalkan model prediksi sehingga dapat memberikan hasil yang lebih akurat.

5. Validasi Model: Setelah model dilatih, maka model tersebut akan divalidasi dengan menggunakan data uji. Proses validasi ini bertujuan untuk mengukur seberapa akurat model tersebut dalam memprediksi hasil yang sebenarnya.
6. Implementasi Model: Setelah model divalidasi, maka model tersebut dapat diimplementasikan dalam sistem produksi. Model akan diintegrasikan dengan sistem produksi untuk digunakan dalam pengambilan keputusan atau pengendalian proses produksi.
7. Monitor dan Evaluasi: Setelah model diimplementasikan, maka akan dilakukan proses monitoring dan evaluasi secara berkala untuk memastikan model masih dapat memberikan hasil yang akurat dan relevan. Proses ini juga akan memastikan sistem produksi dapat berjalan dengan baik dan meminimalkan risiko kesalahan dalam produksi.

Demikianlah prosedur kerja sistem produksi dalam kecerdasan buatan. Penting untuk diingat bahwa proses ini dapat bervariasi tergantung pada jenis aplikasi dan kebutuhan bisnis yang spesifik.

3.5 Strategi Pencarian di Sistem Produksi

Dalam sistem produksi pada kecerdasan buatan, terdapat dua strategi pencarian yang umum digunakan, yaitu data-driven dan goal-driven.

1. *Data-Driven Strategy*: Strategi pencarian data-driven bertujuan untuk menemukan pola atau informasi yang terkandung dalam data secara otomatis. Metode ini seringkali digunakan pada data yang bersifat kompleks, besar, dan beragam. Strategi ini lebih fokus pada pengolahan data untuk menghasilkan informasi yang relevan. Teknik yang sering digunakan dalam strategi

pencarian data-driven adalah *clustering*, *association rule*, dan *classification*.

Karakteristik dari Data-Driven Strategy antara lain:

- a. Data yang besar dan kompleks: Strategi ini cocok digunakan pada data yang jumlahnya besar dan beragam, sehingga diperlukan algoritma yang mampu memproses data tersebut secara efektif dan efisien.
- b. Tidak memiliki hipotesis awal: Strategi ini tidak bergantung pada hipotesis awal atau asumsi yang dibuat sebelumnya. Algoritma akan mengekstrak pola dari data secara otomatis, tanpa dipengaruhi oleh faktor-faktor subjektif.
- c. Berfokus pada pengolahan data: Strategi ini lebih fokus pada pengolahan data untuk menghasilkan informasi yang relevan. Hal ini berbeda dengan strategi pencarian goal-driven yang lebih fokus pada mencapai tujuan.
- d. Menggunakan teknik pengolahan data: Strategi ini menggunakan berbagai teknik pengolahan data, seperti clustering, association rule, dan classification untuk mengekstrak pola dari data.
- e. Tidak memerlukan pengetahuan domain: Strategi ini tidak memerlukan pengetahuan domain yang mendalam. Sehingga, algoritma dapat digunakan pada berbagai jenis data, seperti data medis, data bisnis, dan data lainnya.
- f. Fokus pada akurasi dan kinerja: Strategi ini fokus pada akurasi dan kinerja algoritma dalam menghasilkan informasi yang relevan dari data. Hal ini dapat diukur menggunakan berbagai metrik, seperti precision, recall, dan F1-score.

Dalam mengaplikasikan Data-Driven Strategy, perlu diperhatikan bahwa hasil yang dihasilkan sangat

bergantung pada kualitas data yang digunakan. Oleh karena itu, penting untuk melakukan pre-processing data dan pemilihan teknik pengolahan data yang tepat agar informasi yang dihasilkan lebih akurat dan relevan.

2. *Goal-Driven Strategy*: Strategi pencarian goal-driven bertujuan untuk mencapai tujuan atau target yang telah ditentukan. Metode ini seringkali digunakan pada tugas yang membutuhkan pengambilan keputusan berdasarkan informasi yang tersedia. Strategi ini lebih fokus pada pencapaian tujuan dengan memanfaatkan informasi yang ada. Teknik yang sering digunakan dalam strategi pencarian goal-driven adalah rule-based system, decision tree, dan reinforcement learning. Karakteristik dari Goal-Driven Strategy antara lain:
 - a. Memiliki tujuan yang jelas: Strategi ini memiliki tujuan atau target yang telah ditentukan sebelumnya. Algoritma akan bekerja untuk mencapai tujuan tersebut dengan memanfaatkan informasi yang tersedia.
 - b. Bergantung pada hipotesis awal: Strategi ini bergantung pada hipotesis awal atau asumsi yang dibuat sebelumnya. Algoritma akan mencari solusi berdasarkan hipotesis tersebut dan memperbaikinya secara iteratif.
 - c. Fokus pada pencapaian tujuan: Strategi ini lebih fokus pada pencapaian tujuan dengan memanfaatkan informasi yang ada. Hal ini berbeda dengan strategi pencarian data-driven yang lebih fokus pada pengolahan data untuk menghasilkan informasi yang relevan.
 - d. Menggunakan teknik pengambilan keputusan: Strategi ini menggunakan berbagai teknik pengambilan

keputusan, seperti rule-based system, decision tree, dan reinforcement learning untuk mencapai tujuan.

- e. Memerlukan pengetahuan domain: Strategi ini memerlukan pengetahuan domain yang mendalam untuk membuat hipotesis awal yang tepat dan memperbaikinya secara iteratif. Sehingga, algoritma dapat digunakan pada berbagai jenis masalah, seperti pengambilan keputusan bisnis dan diagnosa medis.
- f. Fokus pada kecepatan dan efisiensi: Strategi ini fokus pada kecepatan dan efisiensi algoritma dalam mencapai tujuan yang telah ditentukan. Hal ini dapat diukur menggunakan berbagai metrik, seperti waktu eksekusi dan akurasi.

Dalam mengaplikasikan Goal-Driven Strategy, perlu diperhatikan bahwa penentuan tujuan yang jelas dan hipotesis awal yang tepat akan sangat mempengaruhi keberhasilan algoritma dalam mencapai tujuan. Oleh karena itu, penting untuk memiliki pengetahuan domain yang mendalam dan melakukan evaluasi secara berkala untuk memperbaiki hipotesis awal dan mencapai tujuan dengan lebih efektif.

Pilihan antara strategi pencarian data-driven atau goal-driven akan tergantung pada karakteristik masalah dan tujuan yang ingin dicapai. Pada beberapa kasus, kedua strategi dapat digabungkan untuk mencapai tujuan yang lebih kompleks dan efektif.

3.6 Metode Penalaran dalam Sistem Produksi

Metode penalaran dalam sistem produksi kecerdasan buatan berfungsi untuk memecahkan masalah dan mencapai tujuan yang diinginkan. Dengan menggunakan metode penalaran yang tepat, sistem produksi dapat mengambil

keputusan secara otomatis berdasarkan aturan dan fakta yang ada pada basis pengetahuan dan working memory.

Metode penalaran juga dapat digunakan untuk meningkatkan efisiensi dan kecepatan sistem produksi dalam memproses informasi. Sebagai contoh, jika menggunakan metode forward chaining, sistem produksi hanya perlu memproses aturan-aturan yang relevan dengan fakta-fakta yang ada pada working memory, sehingga waktu dan sumber daya yang diperlukan dapat lebih efisien dibandingkan dengan metode penalaran lainnya.

Selain itu, metode penalaran dapat membantu sistem produksi untuk mengatasi masalah kompleks dengan cara yang lebih sistematis dan terstruktur, sehingga hasil yang dihasilkan dapat lebih akurat dan konsisten. Metode penalaran juga dapat membantu dalam memperbaiki kinerja sistem produksi dengan mengidentifikasi dan memperbaiki kesalahan yang terjadi pada proses produksi.

Dengan demikian, metode penalaran sangat penting dalam sistem produksi kecerdasan buatan untuk mengoptimalkan kinerja sistem dan mencapai tujuan produksi yang diinginkan.

Terdapat beberapa metode penalaran yang dapat digunakan dalam sistem produksi kecerdasan buatan digital (Kurniawan, 2020), di antaranya adalah:

1. *Forward chaining*

Forward chaining adalah salah satu metode penalaran yang digunakan dalam sistem produksi kecerdasan buatan. Metode ini digunakan untuk memproses aturan-aturan pada basis pengetahuan secara bertahap, dimulai dari fakta-fakta yang ada pada *working memory*, hingga mencapai tujuan yang diinginkan.

Proses *forward chaining* dimulai dengan mengambil fakta-fakta yang ada pada *working memory*, dan mencocokkan

dengan aturan-aturan pada basis pengetahuan. Jika fakta-fakta tersebut sesuai dengan aturan, maka aturan tersebut akan dieksekusi dan hasilnya akan ditambahkan ke *working memory*. Proses ini akan terus berlanjut hingga mencapai tujuan yang diinginkan atau tidak ada lagi aturan yang bisa dieksekusi.

Contoh penggunaan metode *forward chaining* dalam sistem produksi kecerdasan buatan adalah pada sistem pakar diagnosa penyakit. Sistem tersebut akan memulai proses dengan mengambil fakta-fakta gejala yang ada pada pasien, dan mencocokkan dengan aturan-aturan pada basis pengetahuan. Jika ditemukan aturan yang sesuai dengan gejala, maka sistem akan memberikan diagnosis dan rekomendasi pengobatan yang sesuai.

Keuntungan dari metode *forward chaining* adalah sistem produksi hanya perlu memproses aturan-aturan yang relevan dengan fakta-fakta yang ada pada *working memory*, sehingga waktu dan sumber daya yang diperlukan dapat lebih efisien dibandingkan dengan metode penalaran lainnya. Namun, kelemahannya adalah proses *forward chaining* dapat menjadi sangat lambat jika terdapat banyak aturan yang harus diproses.

2. *Backward chaining*

Backward chaining adalah metode penalaran yang digunakan dalam sistem produksi kecerdasan buatan untuk mencapai tujuan dengan cara menganalisis apakah suatu kondisi tertentu memenuhi syarat untuk mencapai tujuan. Metode ini dimulai dari tujuan yang ingin dicapai, kemudian mencari aturan-aturan yang dapat digunakan untuk mencapai tujuan tersebut.

Proses *backward chaining* dimulai dengan menentukan tujuan yang ingin dicapai, dan kemudian mencari aturan-aturan yang berkaitan dengan tujuan tersebut. Jika aturan-

aturan tersebut memerlukan kondisi-kondisi tertentu untuk dieksekusi, maka sistem produksi akan mencari fakta-fakta pada *working memory* yang dapat memenuhi kondisi-kondisi tersebut. Proses ini akan terus berlanjut hingga mencapai fakta-fakta yang diperlukan untuk mencapai tujuan.

Contoh penggunaan metode *backward chaining* dalam sistem produksi kecerdasan buatan adalah pada sistem pakar di bidang hukum. Sistem tersebut akan memulai proses dengan menentukan tujuan yang ingin dicapai, misalnya mengidentifikasi apakah suatu tindakan melanggar hukum atau tidak. Kemudian, sistem akan mencari aturan-aturan yang berkaitan dengan hukum yang berlaku, dan mencocokkan dengan fakta-fakta yang ada pada *working memory*. Jika ditemukan fakta yang memenuhi kondisi-kondisi yang diperlukan, maka sistem akan memberikan keputusan apakah tindakan tersebut melanggar hukum atau tidak.

Keuntungan dari metode *backward chaining* adalah proses penalarannya lebih fokus pada mencapai tujuan yang ingin dicapai, sehingga lebih efektif dan efisien dalam mencapai tujuan yang kompleks. Namun, kelemahannya adalah metode ini dapat memerlukan waktu dan sumber daya yang lebih banyak jika terdapat banyak aturan yang harus diproses atau banyak kondisi yang harus dipenuhi.

3. *Abduction*

Penalaran abduktif juga dapat digunakan dalam sistem produksi kecerdasan buatan digital. Dalam sistem produksi kecerdasan buatan, penalaran abduktif dapat digunakan untuk menghasilkan hipotesis atau penjelasan yang masuk akal berdasarkan data atau fakta yang diperoleh dari sistem tersebut.

Misalnya, dalam sebuah sistem deteksi kecurangan pada transaksi online, penalaran abduktif dapat digunakan untuk menghasilkan hipotesis mengenai kemungkinan adanya kecurangan dalam sebuah transaksi. Hipotesis tersebut kemudian dapat diuji dengan menggunakan metode-metode ilmiah lainnya untuk memastikan kebenarannya.

Selain itu, penalaran abduktif juga dapat digunakan dalam sistem rekomendasi. Dalam sistem rekomendasi, penalaran abduktif dapat digunakan untuk menghasilkan hipotesis mengenai preferensi atau kebutuhan pengguna berdasarkan data historis atau perilaku pengguna. Hipotesis tersebut kemudian dapat diuji dengan memberikan rekomendasi kepada pengguna dan mengamati respons pengguna terhadap rekomendasi tersebut.

Kelemahan penalaran abduktif, yaitu hipotesis yang dihasilkan belum tentu benar atau pasti. Oleh karena itu, perlu dilakukan pengujian dan verifikasi secara terus-menerus untuk memastikan kebenaran hipotesis yang dihasilkan oleh sistem produksi kecerdasan buatan digital.

4. *Deduction*

Penalaran *deduction* atau penalaran deduktif juga dapat digunakan dalam sistem produksi kecerdasan buatan digital. Dalam sistem produksi kecerdasan buatan, penalaran deduktif dapat digunakan untuk menarik kesimpulan logis berdasarkan aturan-aturan atau fakta-fakta yang sudah diketahui sebelumnya.

Misalnya, dalam sistem deteksi penipuan kartu kredit, penalaran deduktif dapat digunakan untuk menarik kesimpulan bahwa sebuah transaksi merupakan transaksi penipuan jika memenuhi aturan-aturan atau kriteria-kriteria yang sudah ditetapkan sebelumnya. Kesimpulan

tersebut didasarkan pada aturan-aturan atau fakta-fakta yang sudah diketahui sebelumnya, seperti lokasi penggunaan kartu kredit, jenis barang atau layanan yang dibeli, dan sebagainya.

Selain itu, penalaran deduktif juga dapat digunakan dalam sistem diagnosis penyakit. Dalam sistem diagnosis penyakit, penalaran deduktif dapat digunakan untuk menarik kesimpulan bahwa seorang pasien menderita suatu penyakit berdasarkan gejala-gejala yang dialami oleh pasien. Kesimpulan tersebut didasarkan pada aturan-aturan atau fakta-fakta yang sudah diketahui sebelumnya mengenai penyakit dan gejalanya.

Kelemahan penalaran deduktif, yaitu dapat menghasilkan kesimpulan yang salah jika aturan atau fakta yang digunakan tidak tepat atau tidak lengkap. Oleh karena itu, perlu dilakukan pengembangan dan pemeliharaan terus-menerus pada aturan-aturan dan fakta-fakta yang digunakan dalam sistem produksi kecerdasan buatan digital untuk memastikan kesimpulan yang dihasilkan akurat dan tepat.

5. *Induction*

Penalaran induction atau penalaran induktif juga dapat digunakan dalam sistem produksi kecerdasan buatan digital. Dalam sistem produksi kecerdasan buatan, penalaran induktif dapat digunakan untuk membuat generalisasi atau prediksi berdasarkan data atau fakta yang diperoleh dari sistem tersebut.

Misalnya, dalam sistem rekomendasi produk, penalaran induktif dapat digunakan untuk membuat prediksi mengenai produk yang disukai oleh pengguna berdasarkan data historis atau perilaku pengguna. Prediksi tersebut didasarkan pada generalisasi atau pola yang ditemukan dari data historis atau perilaku pengguna.

Selain itu, penalaran induktif juga dapat digunakan dalam sistem prediksi cuaca. Dalam sistem prediksi cuaca, penalaran induktif dapat digunakan untuk membuat prediksi cuaca berdasarkan data cuaca historis dan faktor-faktor cuaca seperti suhu, kelembaban, tekanan udara, dan sebagainya. Prediksi tersebut didasarkan pada generalisasi atau pola yang ditemukan dari data cuaca historis dan faktor-faktor cuaca.

Namun, perlu diingat bahwa penalaran induktif juga memiliki kelemahan, yaitu generalisasi atau prediksi yang dihasilkan belum tentu akurat atau tepat. Oleh karena itu, perlu dilakukan pengujian dan verifikasi secara terus-menerus untuk memastikan akurasi dan keandalan generalisasi atau prediksi yang dihasilkan oleh sistem produksi kecerdasan buatan digital. Selain itu, perlu dilakukan pengembangan terus-menerus pada model dan algoritma yang digunakan dalam sistem produksi kecerdasan buatan digital untuk meningkatkan akurasi dan keandalannya.

Metode penalaran yang digunakan tergantung pada masalah yang ingin diselesaikan dan sifat dari basis pengetahuan yang digunakan. Setiap metode penalaran memiliki kelebihan dan kelemahan masing-masing, sehingga pemilihan metode penalaran yang tepat sangat penting untuk mencapai hasil yang optimal.

3.7 Studi Kasus

Beberapa studi kasus dalam sistem produksi dapat dicontohkan melalui metode penalarannya.

3.7.1 Studi Kasus menggunakan Metode Forward Chaining

Salah satu studi kasus yang dapat diimplementasikan menggunakan metode forward chaining adalah sistem diagnosis penyakit pada tanaman.

Dalam sistem ini, data yang dikumpulkan adalah gejala-gejala penyakit pada tanaman. Setiap gejala memiliki kategori atau tipe tertentu, misalnya gejala daun menguning dapat dikategorikan sebagai gejala kekeringan atau kekurangan nutrisi.

Sistem ini bekerja dengan cara menghubungkan gejala-gejala tersebut dengan aturan-aturan atau pengetahuan yang telah diprogramkan di dalamnya. Aturan-aturan ini menggambarkan hubungan antara gejala-gejala dan penyakit pada tanaman.

Misalnya, jika tanaman memiliki gejala daun menguning dan layu, maka sistem akan menganalisis aturan yang terkait dengan gejala-gejala tersebut, seperti "Jika tanaman memiliki gejala daun menguning dan layu, maka kemungkinan tanaman terkena penyakit layu fusarium".

Dalam metode forward chaining, sistem akan mengumpulkan gejala-gejala yang terdeteksi dan mencocokkan dengan aturan-aturan yang telah diprogramkan. Sistem akan terus mengevaluasi gejala-gejala baru yang muncul dan mencari aturan-aturan baru yang sesuai dengan gejala-gejala tersebut.

Jika sistem berhasil menemukan aturan yang sesuai, maka sistem akan menambahkan informasi tersebut ke dalam basis pengetahuan atau knowledge base, dan melakukan diagnosis pada tanaman dengan membandingkan gejala-gejala

yang terdeteksi dengan penyakit yang telah didiagnosis sebelumnya.

Dengan implementasi sistem ini, diharapkan dapat membantu petani dalam melakukan diagnosis penyakit pada tanaman dengan cepat dan akurat, sehingga dapat mengambil tindakan yang tepat dan mencegah penyebaran penyakit pada tanaman.

3.7.2 Studi Kasus menggunakan Metode Backward Chaining

Salah satu studi kasus yang dapat diimplementasikan menggunakan metode backward chaining adalah sistem deteksi kebakaran pada gedung.

Sistem ini bekerja dengan cara memulai dari tujuan akhir atau goal, yaitu deteksi kebakaran pada gedung. Sistem akan mengumpulkan informasi dan memeriksa setiap pernyataan yang tersedia untuk mencapai tujuan tersebut.

Sistem akan memeriksa setiap pernyataan dan mencari hubungan antara pernyataan tersebut dengan tujuan akhir. Jika tidak ada hubungan, sistem akan mencari hubungan pada pernyataan lainnya dan terus memeriksa pernyataan-pernyataan baru hingga menemukan pernyataan yang relevan dengan tujuan akhir.

Misalnya, sistem mulai dengan tujuan deteksi kebakaran pada gedung. Sistem akan memeriksa apakah sensor asap mendeteksi adanya asap di dalam ruangan. Jika sensor asap tidak mendeteksi adanya asap, sistem akan memeriksa sensor suhu dan mencari tahu apakah ada peningkatan suhu yang terdeteksi. Jika sensor suhu tidak mendeteksi adanya peningkatan suhu, maka sistem akan memeriksa sensor api dan mencari tahu apakah ada api yang terdeteksi. Jika sensor api mendeteksi adanya api, maka sistem akan memberikan sinyal alarm dan menghubungi petugas pemadam kebakaran.

Dalam metode backward chaining, sistem akan menghubungkan informasi dan memeriksa hubungan antara informasi-informasi tersebut dengan tujuan akhir. Sistem akan terus mencari informasi baru dan mengevaluasi hubungan antara informasi tersebut dengan tujuan akhir. Jika sistem berhasil menemukan hubungan yang sesuai, maka sistem akan memberikan sinyal alarm dan menghubungi petugas pemadam kebakaran.

Dengan implementasi sistem ini, diharapkan dapat membantu meminimalisir risiko kebakaran pada gedung dan meningkatkan keselamatan penghuni gedung.

3.7.3 Studi Kasus menggunakan Metode Abduction

Salah satu studi kasus yang dapat diimplementasikan menggunakan metode abduction adalah sistem deteksi anomali pada jaringan komputer.

Sistem ini bekerja dengan cara mencari penjelasan atau hipotesis yang paling mungkin untuk menjelaskan anomali pada jaringan komputer. Sistem akan mengumpulkan data atau informasi dari jaringan komputer, seperti log aktivitas, performa jaringan, dan data lainnya.

Sistem akan memulai dengan menentukan anomali atau permasalahan yang terjadi pada jaringan komputer, seperti lambatnya koneksi internet atau server down. Setelah itu, sistem akan mencari hipotesis atau penjelasan yang paling mungkin untuk menjelaskan anomali tersebut.

Misalnya, jika sistem mendeteksi adanya penurunan performa jaringan, sistem akan mencari hipotesis yang mungkin menjadi penyebabnya. Sistem akan memeriksa data aktivitas jaringan dan mencari tahu apakah ada koneksi yang mengalami overload atau terjadi bottleneck pada server. Setelah itu, sistem akan mencari tahu apakah terdapat serangan malware atau virus pada jaringan komputer. Jika

tidak ada hipotesis yang sesuai dengan anomali tersebut, sistem akan mencari informasi lebih lanjut dan terus melakukan penjelasan untuk mencari hipotesis yang sesuai.

Dalam metode abduction, sistem akan mencari hipotesis atau penjelasan yang paling mungkin berdasarkan data atau informasi yang telah dikumpulkan. Sistem akan terus mengumpulkan informasi baru dan memeriksa hubungan antara informasi tersebut dengan anomali yang terjadi. Jika sistem berhasil menemukan hipotesis yang sesuai, maka sistem akan memberikan rekomendasi atau solusi untuk mengatasi permasalahan pada jaringan komputer.

Dengan implementasi sistem ini, diharapkan dapat membantu meningkatkan performa jaringan komputer dan meningkatkan keamanan dari serangan malware atau virus pada jaringan komputer.

3.7.4 Studi Kasus menggunakan Metode Deduction

Salah satu studi kasus yang dapat diimplementasikan menggunakan metode deduction adalah sistem pengenalan pola dalam data biometrik, seperti sidik jari.

Sistem ini bekerja dengan cara membuat aturan-aturan logis atau deduksi untuk mengenali pola pada data biometrik. Sistem akan mengumpulkan data biometrik, seperti sidik jari, dan memprosesnya dengan algoritma pengenalan pola.

Sistem akan memulai dengan membuat aturan-aturan logis atau deduksi yang diperoleh dari pola pada data biometrik. Misalnya, sistem dapat membuat aturan "jika pola pada sidik jari sama dengan pola yang telah terdaftar pada database, maka sidik jari tersebut dapat diidentifikasi sebagai orang yang terdaftar dalam database." Sistem akan melakukan deduksi atau penalaran logis untuk mengenali pola pada sidik jari dan mencocokkannya dengan aturan-aturan yang telah dibuat.

Dalam metode deduction, sistem akan menggunakan aturan-aturan logis atau deduksi untuk mengenali pola pada data biometrik. Sistem akan terus memperbarui dan memperbaiki aturan-aturan tersebut berdasarkan data yang telah dikumpulkan dan hasil pengenalan pola yang diperoleh.

Jika sistem berhasil mengenali pola pada data biometrik, maka sistem akan memberikan identifikasi atau verifikasi terhadap orang yang telah terdaftar dalam database. Jika sistem tidak dapat mengenali pola pada data biometrik, maka sistem akan memberikan informasi bahwa data tersebut tidak dapat diidentifikasi.

Dengan implementasi sistem ini, diharapkan dapat membantu meningkatkan keamanan dan pengamanan pada akses sistem yang memerlukan verifikasi biometrik, seperti pada perbankan atau lembaga keamanan.

3.7.5 Studi Kasus menggunakan Metode Induction

Salah satu studi kasus yang dapat diimplementasikan menggunakan metode induction adalah sistem rekomendasi produk pada situs e-commerce.

Sistem ini bekerja dengan cara mencari pola atau kesamaan pada data transaksi dan preferensi pengguna untuk memberikan rekomendasi produk yang sesuai dengan preferensi pengguna. Sistem akan mengumpulkan data transaksi dan preferensi pengguna, seperti produk yang telah dibeli dan produk yang disukai oleh pengguna.

Sistem akan memulai dengan menentukan pola atau kesamaan pada data transaksi dan preferensi pengguna. Misalnya, sistem akan mencari pola bahwa seorang pengguna yang suka membeli baju anak-anak, kemungkinan besar akan menyukai mainan anak-anak. Sistem akan mencari pola pada data transaksi dan preferensi pengguna dengan menggunakan algoritma induksi.

Dalam metode induction, sistem akan mencari pola atau kesamaan pada data transaksi dan preferensi pengguna dengan menggunakan algoritma induksi. Sistem akan terus memperbarui dan memperbaiki pola atau kesamaan tersebut berdasarkan data yang telah dikumpulkan dan hasil rekomendasi produk yang diberikan.

Jika sistem berhasil menemukan pola atau kesamaan pada data transaksi dan preferensi pengguna, maka sistem akan memberikan rekomendasi produk yang sesuai dengan preferensi pengguna. Jika sistem tidak dapat menemukan pola atau kesamaan pada data transaksi dan preferensi pengguna, maka sistem akan memberikan rekomendasi produk yang umum dan populer.

Dengan implementasi sistem ini, diharapkan dapat membantu meningkatkan penjualan pada situs e-commerce dan memudahkan pengguna dalam mencari produk yang sesuai dengan preferensi mereka.

DAFTAR PUSTAKA

- Kurniawan, R. 2020. *KECERDASAN BUATAN (ARTIFICIAL INTELLIGENCE)*. Medan: Fakultas Sains dan Teknologi UIN-SU Medan.
- Ritha, N. 2020. *Kecerdasan Buatan berbasis Problem Based Learning*. Tanjungpinang: FT UMRAH.

BAB 4

PENCARIAN (SEARCHING)

Oleh Jarwo

4.1 Pengertian *Searching* dalam AI

Searching dalam AI (*Artificial Intelligence*) merupakan sebuah proses menemukan melalui pelacakan ruang pencarian yang telah ditentukan oleh himpunan simpul (node) atau keadaan (*state*) serta keterkaitannya dalam suatu masalah tertentu (Suyanto, *et al.*, 2022).

Searching dalam AI (*Artificial Intelligence*) juga dapat didefinisikan sebagai teknik untuk menemukan solusi atau jawaban dari masalah yang kompleks dengan melakukan pencarian melalui berbagai kemungkinan solusi yang ada dalam sebuah ruang pencarian. Sedangkan teknik *searching*nya dapat digunakan untuk memecahkan masalah dalam berbagai bidang, seperti optimasi, perencanaan, pengambilan keputusan, dan *machine learning*.

Searching dalam AI merujuk pada proses mencari solusi dalam ruang masalah yang kompleks. Dalam hal ini teknik dasar *searching* yang digunakan, berorientasi pada memecahkan berbagai masalah seperti perencanaan, pengambilan keputusan, dan pemrosesan bahasa alami (Russel & Norvig, 2010).

Pencarian dalam AI umumnya dilakukan dengan membandingkan solusi atau jawaban yang dihasilkan dengan kriteria yang telah ditentukan sebelumnya, seperti efisiensi waktu atau keakuratan. Salah satu contoh penggunaan *searching* dalam AI adalah dalam sistem pakar, di mana sistem tersebut mencari jawaban dari pertanyaan yang diajukan oleh

pengguna dengan cara membandingkan input yang diberikan dengan pengetahuan yang telah tersimpan di dalam basis pengetahuan (Suyanto, 2021).

Selain itu, searching juga dapat digunakan dalam machine learning, di mana algoritma pencarian dapat digunakan untuk mencari model yang paling cocok untuk suatu data, atau untuk mencari parameter yang paling optimal untuk suatu model.



Gambar 4.1. Chatbot, terapan Searching AI Digital

(Sumber:<https://stei.itb.ac.id/blog/2019/04/08/bagaimana-ai-dapat-mempelajari-bahasa-manusia-untuk-mempermudah-hidup-kita/>)

Pencarian dalam AI digital dapat merujuk pada proses pencarian informasi yang dilakukan oleh algoritma atau model machine learning dalam lingkungan digital. Ini dapat mencakup pencarian di mesin pencari, pencarian informasi di database besar, atau pencarian informasi dalam data struktur yang kompleks.

Proses pencarian dalam AI digital sering menggunakan teknik seperti pengindeksan, pengklasifikasian, dan analisis teks untuk membantu memproses informasi dan mengembangkan model yang dapat menemukan informasi dengan cepat dan efektif. Contoh pencarian dalam AI digital adalah pencarian produk atau layanan yang spesifik pada situs web e-commerce, pencarian informasi pada database medis, atau pencarian data yang spesifik dalam data ilmiah.

Namun, pencarian dalam AI digital juga dapat menghadapi tantangan seperti pemrosesan bahasa alami, pengenalan gambar, dan penyelesaian masalah yang kompleks. Oleh karena itu, pengembangan algoritma dan model machine learning yang efektif dan canggih menjadi penting untuk meningkatkan kinerja pencarian dalam AI digital (Joshi, 2017).

Secara keseluruhan, searching dalam AI adalah teknik penting yang digunakan untuk memecahkan berbagai masalah kompleks dengan mencari solusi yang optimal atau sub-optimal melalui penggunaan algoritma searching yang efektif dan efisien.

4.2 Ruang Lingkup Searching dalam AI

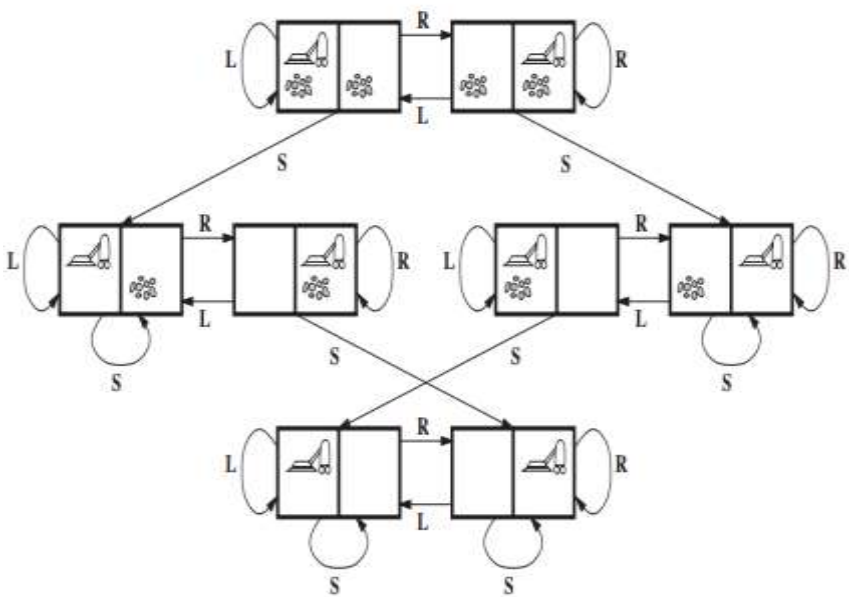
Ruang lingkup searching dalam AI, bisa dikatakan sangat luas, karena searching merupakan teknik dasar yang digunakan dalam banyak bidang AI, seperti machine learning, computer vision, natural language processing, robotics, dan lain-lain. Berikut adalah beberapa contoh ruang lingkup searching dalam AI (Joshi, 2017):

- a. Pencarian jalur terpendek: Pencarian jalur terpendek adalah salah satu bentuk searching dalam AI yang banyak digunakan, seperti mencari rute tercepat dalam peta atau mencari jalur terpendek untuk menghindari hambatan.
- b. Perencanaan tindakan: Perencanaan tindakan melibatkan pencarian jalur untuk mencapai tujuan tertentu, seperti

merencanakan pergerakan robot untuk melakukan tugas tertentu atau merencanakan rute terpendek untuk sebuah kendaraan otonom.

- c. Pengambilan Keputusan: Pencarian solusi terbaik atau optimal dalam situasi yang kompleks dengan mempertimbangkan berbagai faktor dan konsekuensi.
- d. Pengenalan Pola: Pencarian pola atau keteraturan dalam data yang kompleks untuk tujuan seperti klasifikasi atau prediksi.
- e. Pemrosesan bahasa alami: Pencarian dalam pemrosesan bahasa alami digunakan untuk memahami makna dari kalimat atau teks, seperti mencari relasi antara entitas atau topik yang dibahas dalam teks.
- f. Optimisasi: Optimisasi mencari solusi optimal atau sub-optimal dalam masalah yang kompleks, seperti mencari parameter terbaik dalam model machine learning atau mencari solusi terbaik untuk masalah routing.
- g. Pengambilan keputusan: Pencarian juga digunakan dalam pengambilan keputusan, seperti mencari keputusan terbaik dalam permainan atau mencari rekomendasi produk terbaik dalam sistem rekomendasi.
- h. Analisis data: Pencarian digunakan dalam analisis data untuk menemukan pola atau tren dalam data, seperti mencari hubungan antara variabel atau mencari anomali dalam data.
- i. Robotika: Pencarian jalur yang optimal untuk robot dalam ruang tiga dimensi dengan menghindari rintangan.
- j. Sistem Multi-Agen: Pencarian solusi dalam sistem multi-agen, di mana agen-agen tersebut berinteraksi dalam situasi yang kompleks dan dinamis.

Secara keseluruhan, searching dalam AI memiliki ruang lingkup yang sangat luas dan digunakan dalam banyak bidang AI untuk memecahkan masalah yang kompleks dan mencari solusi yang optimal atau sub-optimal. Ilustrasi Mesin Robot Vacuum Cleaner berikut ini didasarkan pada mekanisme searching yang telah ditanamkan pemrograman kecerdasan buatan, yang mampu melakukan gerak ke kiri, ke kanan dan gerak menghisap kotoran.



Gambar 4.2. Ruang Keadaan Lingkup Search Robot dengan keterangan aktifitas Vacuum. L = left, kiri, R= Right, Kanan, S = Suck, menghisap.

(Sumber: (Russel & Norvig, 2010))

Ruang lingkup searching dalam AI terus berkembang seiring dengan kemajuan teknologi dan pengembangan aplikasi AI yang semakin canggih. Dalam setiap domain masalah

tersebut, teknik searching dalam AI digunakan untuk memecahkan masalah secara efisien dan efektif dengan mempertimbangkan berbagai faktor dan kompleksitas masalah (Russel & Norvig, 2010).

4.3 Tujuan Searching

Tujuan utama dari searching dalam AI adalah untuk menemukan solusi yang optimal atau sub-optimal dalam masalah yang diberikan. Tujuan pencarian bergantung pada konteks masalah dan tujuan aplikasi AI yang ingin dicapai. Tujuan pencarian dalam beberapa konteks AI adalah sebagai berikut (Jamaaluddin & Sulistyowati, 2021):

- a. Penyelesaian Pemrosesan Bahasa Alami: Tujuan pencarian dalam konteks ini adalah untuk menemukan jawaban terbaik untuk pertanyaan pengguna dari korpus teks yang tersedia. Tujuan pencarian adalah untuk menemukan jawaban yang paling relevan dan tepat dalam waktu yang efisien.
- b. Pengenalan Pola: Tujuan pencarian dalam konteks ini adalah untuk menemukan pola atau fitur yang paling penting dalam data. Tujuan pencarian adalah untuk mengidentifikasi pola yang paling signifikan dan membedakan dari data yang tidak relevan.
- c. Membantu Pengambilan Keputusan: Tujuan pencarian dalam konteks ini adalah untuk menemukan tindakan yang paling optimal atau keputusan terbaik dalam konteks tertentu. Tujuan pencarian adalah untuk memilih tindakan atau keputusan yang memiliki hasil terbaik dalam waktu yang terbatas.
- d. Mencari solusi optimal: Tujuan utama pencarian adalah untuk menemukan solusi yang optimal dalam masalah tertentu. Solusi optimal adalah solusi yang memiliki nilai maksimum atau minimum yang diinginkan.

- e. Mencari solusi sub-optimal: Jika pencarian optimal terlalu sulit atau memakan waktu terlalu lama, tujuan pencarian dapat diubah menjadi mencari solusi yang sub-optimal yang cukup dekat dengan solusi optimal. Solusi sub-optimal sering kali dapat ditemukan dengan lebih cepat daripada solusi optimal.
- f. Memprediksi atau mengklasifikasikan data: Pencarian digunakan untuk memprediksi atau mengklasifikasikan data dengan mengeksplorasi berbagai fitur atau pola dalam data dan menemukan pola yang paling penting.
- g. Memperbaiki kinerja sistem: Pencarian digunakan untuk memperbaiki kinerja sistem AI dengan mencari solusi yang lebih baik dalam masalah yang diberikan.

Sehingga dapat disimpulkan bahwa tujuan pencarian dalam AI adalah untuk menemukan solusi yang optimal atau sub-optimal dalam suatu masalah tertentu. Pencarian digunakan ketika tidak ada solusi yang diketahui secara langsung dan solusi harus ditemukan dengan mengeksplorasi berbagai kemungkinan.

4.4 Sifat Masalah dalam Searching AI

Masalah yang dapat diselesaikan melalui pencarian dalam AI memiliki beberapa sifat, antara lain (Jaya, *et al.*, 2018):

- a. State Space: Sifat ini mengacu pada ruang keadaan atau ruang pencarian yang terdiri dari semua keadaan yang mungkin dalam masalah. Ruang keadaan dapat berupa graf, pohon, atau struktur data lain yang merepresentasikan semua kemungkinan keadaan.
- b. Operator: Sifat ini mengacu pada tindakan atau operasi yang dapat diterapkan pada keadaan dalam masalah.

Operator ini memungkinkan perpindahan dari satu keadaan ke keadaan lain dalam ruang keadaan.

- c. Tujuan: Sifat ini mengacu pada kondisi akhir atau tujuan yang harus dicapai dalam masalah. Tujuan biasanya ditentukan sebagai keadaan atau kumpulan keadaan yang harus dicapai atau dihindari dalam masalah.
- d. Path Cost: Sifat ini mengacu pada biaya atau nilai yang terkait dengan setiap operasi atau perpindahan dari satu keadaan ke keadaan lain dalam ruang keadaan. Biaya dapat diukur dalam bentuk waktu, energi, biaya finansial, atau kriteria lainnya yang relevan dengan masalah.
- e. Solvability: Sifat ini mengacu pada kemampuan untuk menyelesaikan masalah melalui pencarian. Masalah yang dapat diselesaikan melalui pencarian harus memenuhi beberapa syarat, seperti keberadaan ruang keadaan, operator yang didefinisikan dengan baik, dan tujuan yang dapat dicapai.
- f. Optimalitas: Sifat ini mengacu pada kemampuan pencarian untuk menemukan solusi optimal atau sub-optimal dalam waktu yang terbatas. Solusi yang ditemukan harus memenuhi kriteria pencarian yang ditetapkan dan harus efisien dan efektif.

Dalam *searching* AI, sifat masalah tersebut harus dipertimbangkan saat merancang algoritma pencarian yang sesuai untuk masalah yang diberikan. Algoritma pencarian yang tepat dapat memastikan pencarian mencapai solusi yang optimal atau sub-optimal dengan biaya yang minimal, mempengaruhi kesulitan dan kompleksitas dalam menemukan solusi yang optimal atau sub-optimal. Oleh karena itu, memahami sifat masalah adalah penting dalam memilih teknik dan pendekatan yang tepat untuk menyelesaikan masalah.

4.5 Kriteria Searching dalam AI

Kriteria pencarian yang digunakan dalam suatu masalah tertentu tergantung pada tujuan pencarian dan sifat masalahnya, karena dalam praktiknya, kriteria pencarian yang digunakan tergantung pada jenis masalah dan sumber daya komputasi yang tersedia. Pencarian optimal sering kali lebih kompleks dan memerlukan lebih banyak sumber daya komputasi, sementara pencarian heuristik dapat memberikan solusi yang memadai dengan waktu yang lebih cepat dan biaya yang lebih rendah.

Dalam hal evaluasi teknik searching dalam AI, terdapat beberapa kriteria tersendiri yang digunakan untuk pengukuran serta mengevaluasi keefektifan teknik searching AI. Beberapa kriteria itu meliputi (Suyanto, 2021):

- a. **Kecepatan.** Kriteria ini mengukur dan mengevaluasi seberapa cepat teknik searching dapat menemukan solusi atau jawaban yang optimal atau sub-optimal dalam lingkup masalah tertentu.
- b. **Kebenaran.** Kriteria ini mengevaluasi seberapa akurat solusi atau jawaban yang ditemukan oleh teknik searching dalam AI.
- c. **Ketergantungan pada sumber daya.** Kriteria ini mengukur seberapa bergantungnya teknik searching pada sumber daya tertentu seperti memori atau CPU.
- d. **Keandalan.** Kriteria keandalan mengevaluasi tingkat andalnya sebuah teknik searching dalam AI dalam menemukan solusi atau jawaban yang sama untuk masalah yang sama di waktu yang berbeda.
- e. **Kompleksitas waktu.** Kriteria yang mengevaluasi kompleksitas waktu teknik searching dalam AI dalam menyelesaikan masalah tertentu.

- f. **Kompleksitas ruang.** Kriteria yang mengukur seberapa kompleksitas ruang teknik searching dalam AI dalam menyelesaikan masalah tertentu.
- g. **Skalabilitas.** Kriteria yang mengukur seberapa baik teknik searching dapat disesuaikan dengan skala masalah yang lebih besar.
- h. **Keamanan.** Kriteria yang mengukur seberapa aman teknik searching dalam AI dari serangan atau manipulasi.

Kriteria ini sangat penting dalam memilih teknik searching yang paling efektif dalam menyelesaikan masalah tertentu dalam konteks AI.

4.6 Algoritma Pencarian dalam AI

"*Searching*" dalam konteks AI (*Artificial Intelligence*) biasanya merujuk pada algoritma pencarian yang digunakan untuk menemukan solusi atau informasi dalam dataset atau ruang pencarian. Berikut adalah beberapa algoritma searching yang sering digunakan dalam AI (Russel & Norvig, 2010):

- a. *Breadth-First Search (BFS)*: BFS adalah *algoritma searching* yang mengunjungi simpul secara lebar dan mencari solusi dengan cara mengeksplorasi simpul-simpul pada level yang sama terlebih dahulu, sebelum melanjutkan ke level selanjutnya. Algoritma ini cocok untuk masalah yang memiliki ruang pencarian yang kecil.
- b. *Depth-First Search (DFS)*: DFS adalah *algoritma searching* yang mengunjungi simpul satu per satu secara vertikal, dari simpul awal hingga menemukan simpul tujuan. Algoritma ini cocok untuk masalah yang memiliki ruang pencarian yang besar dan kompleks.
- c. *Uniform Cost Search (UCS)*: UCS adalah *algoritma searching* yang mencari solusi dengan mempertimbangkan biaya dari setiap jalur. Algoritma ini mencari jalur dengan biaya

terkecil dan diterapkan pada masalah dengan biaya yang kompleks.

- d. *A* Search*: *A* Search* adalah algoritma searching yang menggabungkan BFS dan fungsi heuristik untuk mencari jalur yang paling optimal dalam graf. Fungsi heuristik menilai jarak atau biaya dari simpul tertentu ke simpul tujuan, yang memungkinkan *A* Search* untuk memprioritaskan simpul-simpul yang lebih dekat ke simpul tujuan dalam mencari solusi.
- e. *Greedy Best-First Search*: *Greedy Best-First Search* adalah algoritma searching yang memilih simpul dengan nilai heuristik terkecil sebagai simpul berikutnya yang akan dieksplorasi. Algoritma ini biasanya digunakan dalam masalah di mana tujuan pencarian diketahui dengan baik.
- f. *Hill Climbing*: *Hill Climbing* adalah algoritma searching yang mencari solusi dengan mengevaluasi simpul-simpul yang memiliki nilai fungsi heuristik terbaik. Algoritma ini cocok untuk masalah yang tidak memiliki banyak solusi.
- g. *Simulated Annealing*: *Simulated Annealing* adalah algoritma searching yang digunakan untuk menemukan solusi optimal dalam masalah optimisasi yang kompleks dan berbentuk non-linear. Teknik ini menggunakan konsep fisika, di mana algoritma akan menjelajahi lingkungan solusi dan mencari solusi yang lebih baik dengan kemungkinan memilih solusi yang buruk terlebih dahulu, sebelum kemudian merubah ke keadaan yang lebih baik.
- h. *Genetic Algorithm*: *Genetic Algorithm* adalah algoritma *searching* yang menggunakan prinsip evolusi untuk mencari solusi terbaik dalam masalah optimisasi. Teknik ini menciptakan populasi solusi acak, kemudian menggunakan operator genetik seperti crossover dan mutasi untuk menciptakan generasi solusi baru, yang kemudian dievaluasi dan dipilih solusi terbaiknya.

Pemilihan algoritma searching dalam AI tergantung pada karakteristik masalah dan tujuan yang ingin dicapai. Oleh karena itu, pemilihan algoritma yang tepat dapat sangat penting untuk menemukan solusi optimal atau sub-optimal dalam masalah AI.

Dalam Pengembangannya Penerapan Kecerdasan Buatan akan banyak menggunakan searching terhadap data. Sehingga sangat penting data-data yang selalu tersedia agar Kecerdasan Buatan menjalankan fungsi yang dikehendaki. Data dan informasi tersebut harus disimpan dalam suatu penyimpanan terpusat dan terdistribusi dengan sistem manajemen yang terorganisasi. Infrastruktur ini akan digunakan untuk penyiapan, pemindahan, penyimpanan, pengolahan, dan analisa baik yang terpusat maupun yang terdistribusi dengan sistem manajemen dan akses yang aman, dan terkelola. Sumber-sumber data pada umumnya terakumulasi secara sektoral dan di masing-masing instansi (BPPT, 2020).

DAFTAR PUSTAKA

- BPPT, 2020. *Strategi Nasional Kecerdasan Artificial Indonesia 2020-2045*. 1 ed. Jakarta: BPPT.
- Jamaaluddin & Sulistyowati, I., 2021. *Buku Ajar Kecerdasan Buatan*. 1 ed. Sidoarjo: UMSIDA Press.
- Jaya, H. et al., 2018. *Kecerdasan Buatan*. 1 ed. Makassar: FMIPA Universitas Negeri Makassar.
- Joshi, P., 2017. *Artificial Intelligence with Python*. 1 ed. Birmingham UK: Packt Publishing.
- Russel, S. & Norvig, P., 2010. *Artificial Intelligence: A Modern Approach 3rd edition*. 3 ed. New Jersey: Prentice Hall.
- Suyanto, 2021. *Artificial Intelligence: Searching, Reasoning, Planning, dan Learning*. 3 ed. Bandung: Penerbit Informatika.
- Suyanto, et al., 2022. *Explainable Artificial Intelligence Menggunakan Metode-Metode Berbasis Nearest Neighbors*. 1 penyunt. Bandung: Informatika.

BAB 5

METODE-METODE PENCARIAN

Oleh Johni S Pasaribu

5.1 Pendahuluan

Kita hidup di era di mana sebagian besar hal yang kita lakukan dan tugas sehari-hari kita terkomputerisasi dengan tujuan untuk meningkatkan keberlanjutannya dan mengurangi upaya manusia. Ketika mempertimbangkan tugas harian individu, tampaknya tugas-tugas ini dapat dibagi dalam subkategori. Subkategori pertama terdiri dari pekerjaan yang membutuhkan tenaga fisik, sedangkan subkategori kedua berisi pekerjaan yang membutuhkan *brainstorming*. Yang terbaru disebutkan bahwa seseorang dapat menghadapi perhitungan yang rumit. Otak manusia mungkin memerlukan beberapa jam atau bahkan sehari-hari untuk melakukan perhitungan tersebut, sedangkan komputer dengan "instruksi yang baik" mungkin dapat melakukan perhitungan dalam waktu yang sangat singkat.

Diketahui bahwa komputer adalah mesin yang hanya mampu melakukan tugas yang diberikan kepadanya; oleh karena itu "sangat penting" untuk menangani masalah yang terdefinisi dengan baik dan mengetahui kendala dan strategi pemecahan masalah ini dengan menggunakan komputer. Kecerdasan Buatan muncul sebagai ilmu yang mempelajari simulasi cara berpikir otak manusia dan mencoba menerapkannya di komputer. Dengan kata lain, dalam Kecerdasan Buatan tujuannya adalah untuk meniru "komputer pintar" yang artinya komputer dapat bertindak dengan cara yang sama seperti otak manusia.

Perbedaan utama antara manusia dan komputer adalah manusia memiliki otak, sehingga dapat berpikir dan menyesuaikan perilakunya sesuai dengan keadaan. Komputer tidak memiliki otak seperti manusia tetapi dapat dengan cepat melakukan beberapa instruksi yang terdefinisi dengan baik. Kecerdasan Buatan adalah bidang ilmu komputer yang bertujuan untuk memecahkan masalah melalui pemodelan matematika. Bidang ini juga mempelajari berbagai teknik pencarian. Memang, tujuannya adalah menemukan algoritma yang baik untuk mencapai target atau solusi yang memuaskan kondisi awal dari masalah yang diberikan. Manusia seringkali mampu memecahkan masalah tertentu dengan menggunakan intuisinya meskipun pendekatan formal untuk memecahkan masalah itu mungkin ada. Manusia mungkin memerlukan banyak waktu untuk memecahkan masalah yang diberikan. Namun, komputer dapat membantu memecahkan masalah jika ditulis dalam bahasa yang dapat dimengerti oleh mesin.

Dalam Kecerdasan Buatan, algoritma pencarian diimplementasikan untuk memecahkan berbagai masalah. Ada dua kategori algoritma pencarian: pencarian tanpa informasi (*uninformed search*) atau pencarian buta (*blind search*) dan pencarian dengan informasi (*informed search*) dan juga disebut pencarian heuristik (*heuristic search*). Metode pencarian tanpa informasi (*uninformed search*) dicirikan oleh fakta bahwa tidak ada pengetahuan sebelumnya, spesifikasi atau petunjuk untuk solusi masalah yang diberikan sebelumnya. Strategi pencarian dengan informasi (*informed search*) dicirikan oleh fakta bahwa beberapa pengetahuan sebelumnya atau petunjuk diberikan sebagai fungsi heuristik. Heuristik membantu meningkatkan kekuatan dan kinerja algoritma.

Seperti yang telah disebutkan, teknik pencarian dikelompokkan menjadi dua kategori utama yaitu teknik pencarian tanpa informasi dan teknik pencarian informasi.

Teknik pencarian tanpa informasi (*uninformed search*) meliputi seperti breadth-first search, depth-first search, depth-limited search, iterative deepening search, uniform cost search, dan bidirectional search. Pencarian informasi (*informed search*) atau disebut juga pencarian heuristik (*heuristic search*) meliputi best-first search, greedy best-first search, hill climbing, steepest ascent hill climbing dan A* search. Yang dibahas dalam bab ini untuk pencarian tanpa informasi atau pencarian buta (*blind search*) yaitu teknik pencarian breadth-first search dan depth-first search saja. Sedangkan untuk pencarian dengan informasi atau pencarian heuristik (*heuristic search*) meliputi pencarian hill climbing, best-first search dan A* search saja. Sifat **kelengkapan**, **optimal**, **kompleksitas waktu**, dan **kompleksitas ruang** dari semua teknik pencarian yang disebutkan di atas dibahas. **Kelengkapan**: Sifat ini untuk memeriksa apakah algoritma tersebut menjamin menemukan solusi dari masalah, jika ada. **Optimal**: Sifat ini menyelidiki apakah strategi memberikan solusi optimal atau tidak. **Kompleksitas waktu**: Sifat ini mengevaluasi waktu yang diperlukan untuk menemukan solusi masalah. **Kompleksitas ruang**: Sifat ini mengevaluasi memori dan ruang yang diperlukan untuk menjalankan algoritma pencarian.

5.2 Tinjauan pustaka

Subbab ini adalah tinjauan umum dari beberapa literatur yang sudah terbit dalam teknik pencarian dimana fokus pada teknik pencarian tanpa informasi dan dengan informasi serta pada penerapannya. Dalam (Kuruvilla, M., & Tabassum, 2014), misalnya, beberapa algoritma pencarian yang umum dan terkenal seperti Breadth First, Depth First, A* Search, Greedy Best First, dan Hill Climbing dieksplorasi dan kinerjanya dibandingkan. Kesesuaian teknik pencarian khusus atas teknik lain diidentifikasi untuk situasi yang berbeda.

Evaluasi dilakukan dengan mendapatkan solusi yang tepat dari soal N-puzzle dan 8 Queen Puzzle. Hasil percobaan menunjukkan bahwa pencarian A* adalah algoritma terbaik untuk memecahkan N-puzzle dan 8 Queen Puzzle.

Pendekatan berdasarkan integrasi teknik pencarian breadth-first dan depth-first dalam satu algoritma untuk memiliki kekuatan dari kedua pendekatan tersebut adalah direalisasikan untuk masalah lebar pohon (Zhou, R., & Hansen, 2009). Manfaat dari kombinasi baru yang diusulkan dari strategi pencarian tersebut dibandingkan dengan menggunakan salah satu metode saja sudah dapat dibenarkan.

Dalam (Ashwani, C., & Manu, 2014), algoritma pencarian yang berbeda dibandingkan untuk menemukan pendekatan terbaik untuk memecahkan masalah tertentu, karena berbagai algoritma berperilaku berbeda untuk memecahkan masalah. Jadi tujuannya adalah untuk mengkategorikan teknik pencarian sehubungan dengan kinerjanya dan spesifikasi tentang jenis masalah yang lebih kuat untuk digunakan.

Beberapa algoritma pencarian untuk menemukan solusi biaya rendah dari metode yang dipilih (masalah MIN) dibahas dalam (Stern, R., Kiesel, S., Puzis, R., Felner, A., & Ruml, 2015). Sementara algoritma pencarian dibandingkan dalam hal mengamati pendekatan yang paling nyaman untuk memecahkan hadiah yang tinggi (Masalah MAX). Dinyatakan bahwa teknik pencarian tanpa informasi yang efektif untuk masalah MIN bisa jadi tidak efektif untuk masalah MAX. Pada saat yang sama, teknik pencarian heuristik dapat berhasil diterapkan pada masalah MAX.

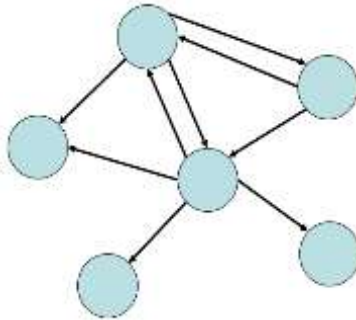
Banyak masalah dalam AI dapat diselesaikan secara teori dengan mencari secara cerdas melalui banyak kemungkinan solusi (Luger, 2009). Penalaran telah terbukti direduksi menjadi melakukan pencarian. Misalnya, pembuktian logis dapat dilihat sebagai pencarian jalur yang mengarah dari

premis ke kesimpulan, dimana setiap langkah merupakan penerapan aturan inferensi (Goebel, 1998). Algoritma perencanaan mencari melalui pohon tujuan dan sub tujuan berusaha untuk menemukan jalur ke tujuan target, sebuah proses yang disebut Analisis Means-End (Goebel, 1998). Algoritma robotik untuk menggerakkan anggota tubuh dan menggenggam objek menggunakan pencarian lokal di ruang konfigurasi (Russell, Stuart J. and Norvig, 2010). Banyak algoritma pembelajaran menggunakan algoritma pencarian berdasarkan optimasi.

Pencarian lengkap yang sederhana (Nilsson, 1998) jarang cukup untuk sebagian besar masalah dunia nyata. Ruang pencarian (jumlah tempat untuk mencari) dengan cepat tumbuh menjadi angka-angka. Hasilnya adalah pencarian yang terlalu lambat atau tidak pernah selesai. Solusinya, untuk banyak masalah, adalah menggunakan "heuristik" atau "aturan praktis" yang menghilangkan pilihan yang tidak mungkin mengarah ke tujuan (disebut pemangkasan pohon pencarian). Heuristik memasok program dengan "tebakan terbaik" untuk apa solusinya terletak pada (Nilsson, 1998; Luger, 2009; Russell, Stuart J. and Norvig, 2010).

5.3 Traversal Graf

Graf terdiri dari sekumpulan node yang dihubungkan oleh edge. Struktur dunia maya disusun sebagai graph; dengan kata lain terdiri dari tempat (node), dan koneksi antara tempat-tempat itu. Sebagian besar aplikasi graf membutuhkan traversal graf. Traversal berarti mengunjungi setiap node dalam graph tepat satu kali. Ini juga berarti bahwa tidak boleh ada siklus dalam traversal. Dua teknik yang umum digunakan dengan traversal graph misalnya adalah: *Depth First Search* (DFS) dan *Breadth First Search* (BFS).



Gambar 5.1. Graf berarah

Graf dapat dimodelkan berupa: situs dan tautan di web, wabah penyakit, jaringan sosial, geografi, grafik tugas dan ketergantungan, pemodelan jaringan dan lalu lintas jaringan, dan banyak lagi. Algoritma traversal graf adalah mengunjungi simpul dengan cara yang sistematis. Graf merupakan representasi persoalan, sehingga Traversal graf adalah pencarian solusi.



Gambar 5.2. *Social Graph*
(Sumber: (Munir, 2015))

Dalam proses pencarian solusi, terdapat dua pendekatan graf:

1. Graf statis: graf yang sudah terbentuk sebelum proses pencarian dilakukan. Graf direpresentasikan sebagai struktur data
2. Graf dinamis: graf yang terbentuk saat proses pencarian dilakukan. Graf tidak tersedia sebelum pencarian, graf dibangun selama pencarian solusi.

5.4 Metode pencarian dalam kecerdasan buatan

Banyak algoritma pencarian tradisional digunakan dalam aplikasi AI. Untuk masalah yang kompleks, algoritma tradisional tidak dapat menemukan solusi dalam batas waktu dan ruang. Akibatnya, banyak teknik khusus yang dikembangkan diantaranya menggunakan fungsi heuristik. Metode pencarian adalah konsep sentral dalam kecerdasan buatan. Ini melekat pada teknik pemecahan masalah kecerdasan buatan (AI). Pada dasarnya, masalah kecerdasan buatan sebenarnya kompleks. Upaya untuk memecahkan masalah yang ditangani manusia dengan menerapkan kemampuan kognitif bawaan, pengenalan pola dan lain-lain menggunakan komputer selalu harus beralih ke pertimbangan pencarian (Luger, 2009).

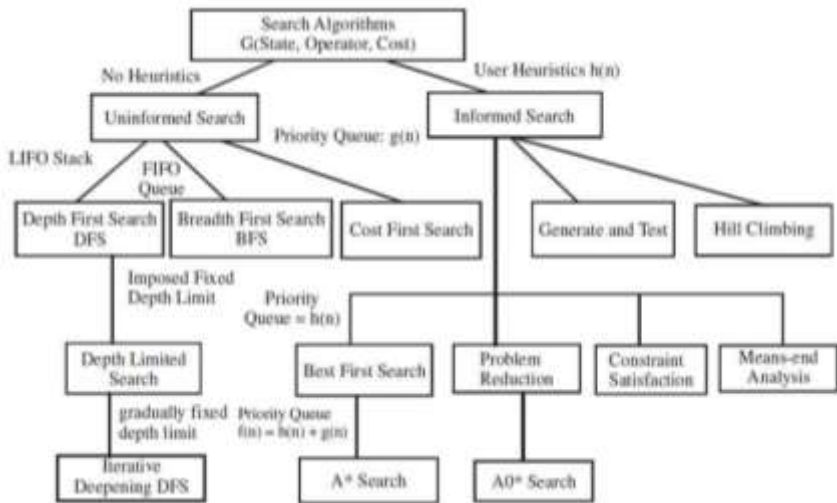
Semua teknik pencarian dapat ditemukan termasuk dalam salah satu dari dua kategori metode pencarian yaitu:

1. Metode tanpa informasi (buta): tidak ada informasi tambahan
2. Metode heuristik atau informasi: pencarian berbasis heuristik

Algoritma pencarian tanpa informasi atau algoritma Brute-force, mencari melalui ruang pencarian semua kandidat yang mungkin untuk solusi memeriksa apakah setiap kandidat memenuhi pernyataan masalah. Uninformed Search: Juga

disebut pencarian buta, lengkap atau brute-force, tidak menggunakan informasi tentang masalah untuk memandu pencarian dan karena itu mungkin tidak terlalu efisien.

Algoritme pencarian informasi menggunakan fungsi heuristik yang khusus untuk masalah, menerapkannya untuk memandu pencarian melalui ruang pencarian untuk mencoba mengurangi jumlah waktu yang dihabiskan dalam pencarian. Algoritma yang menggunakan fungsi heuristik disebut algoritma heuristik. Algoritme heuristik tidak terlalu cerdas; mereka tampak cerdas karena mencapai kinerja yang lebih baik. Algoritma heuristik lebih efisien karena memanfaatkan umpan balik dari data untuk mengarahkan jalur pencarian. Heuristik yang baik akan membuat pencarian informasi secara dramatis mengungguli pencarian tanpa informasi: misalnya *Traveling Salesman Problem* (TSP), di mana tujuannya adalah untuk menemukan solusi yang baik daripada menemukan solusi terbaik. *Informed Search*: Disebut juga pencarian heuristik atau cerdas, ini menggunakan informasi tentang masalah untuk memandu pencarian—biasanya menebak jarak ke keadaan tujuan dan karena itu efisien, tetapi pencarian bisa tidak selalu memungkinkan.



Gambar 5.3. Algoritma Pencarian
(Sumber: (PRASHANTA KUMAR PATRA, no date))

5.5 Metode pencarian buta (*exhaustic/blind*)

Dalam subbab ini fokus pada strategi pencarian tanpa informasi (*blind search*) dimana merujuk pada sekelompok teknik pencarian yang dikenal sebagai pencarian tanpa pengetahuan atau informasi apa pun untuk mencapai tujuan. Pencarian tanpa informasi juga dikenal sebagai pencarian buta.

5.5.1 *Breadth-first search*

Ini adalah strategi pencarian di mana ekspansi dimulai dari root node. Setelah itu penerus simpul akar diperluas. Proses ini diulang sampai penerus terakhir yang terlibat. Struktur data *First in First out* (FIFO) atau antrian (*queue*) lebih disukai untuk menerapkan strategi pencarian breadth-first.

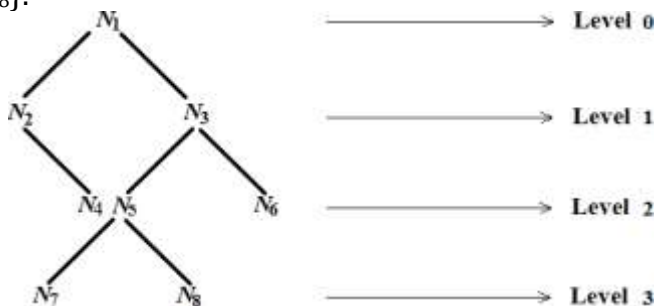
Kriteria kelengkapan dari strategi pencarian breadth-first digunakan untuk mengevaluasi kinerja algoritma. Asumsikan jumlah faktor percabangan adalah $b < \infty$, dan simpul tujuan paling dangkal terletak di kedalaman $d < \infty$. Kemudian

jumlah node dihasilkan oleh algoritma pencarian breadth-first adalah

$$b + b^2 + b^3 + \dots + b^d = O(b^d)$$

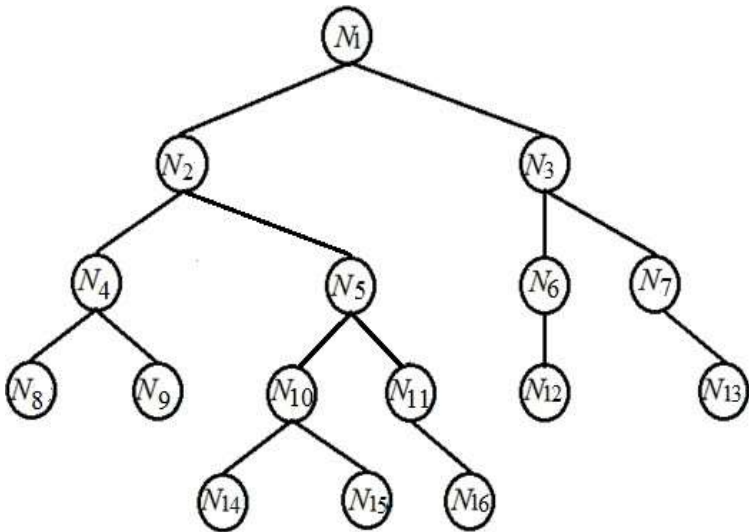
Kelengkapan pencarian breadth-first sudah jelas. Jika lokasinya paling dangkal simpul tujuan berada di kedalaman $d < \infty$, maka algoritma akan menemukannya (tujuan terdangkal node) setelah itu menghasilkan semua node yang lebih dangkal yang semuanya gagal untuk tes tujuan. Breadth-first search adalah tree-traversal jika perluasan dilakukan dari satu level ke level lainnya.

Gambar 5.4 menunjukkan pohon traversal breadth-first dengan level dan node yang sesuai diberikan di bawah ini: Level 0 $\{N_1\}$; Level 1 $\{N_2, N_3\}$; Level 2 $\{N_4, N_5 \text{ dan } N_6\}$, dan Level 3 $\{N_7, N_8\}$.



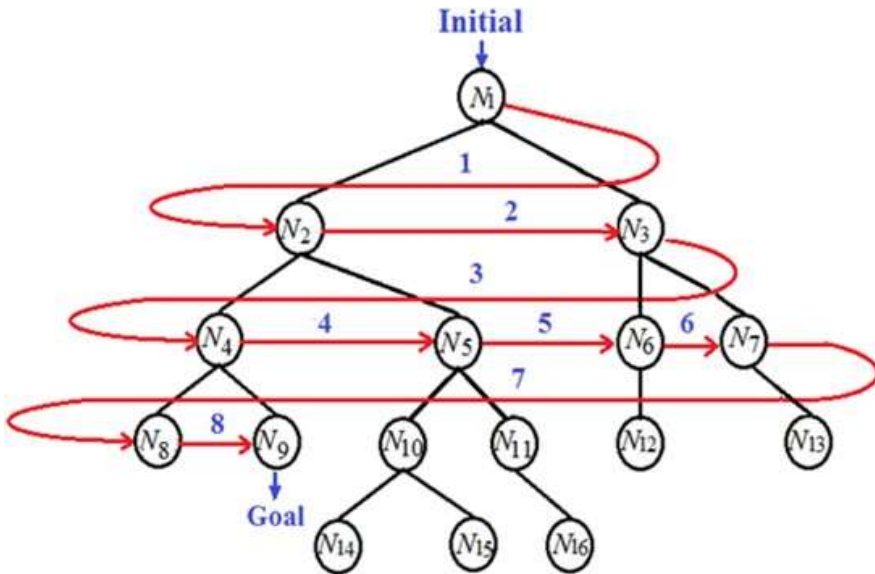
Gambar 5.4. Pohon traversal breadth-first

Urutan node yang diperoleh setelah penerapan algoritma pencarian breadth-first untuk menjelajahi pohon yang diwakili pada Gambar 5.4 adalah $\{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8\}$. Gambar 5.5 menunjukkan graf 16-simpul yang digunakan untuk teknik pencarian informasi.



Gambar 5.5. Graf 16-node untuk pencarian informasi

Misalkan tujuannya adalah untuk berpindah dari simpul awal N_1 ke simpul tujuan N_9 . Gambar 5.6 berikut mengilustrasikan representasi graf dari urutan ekspansi simpul dalam pencarian luas-pertama dari simpul awal N_1 ke simpul tujuan N_9 untuk graf yang direpresentasikan dalam Gambar 5.5.



Gambar 5.6. Representasi graf dari urutan simpul dalam pencarian breadth-first dari simpul awal N_1 ke simpul tujuan N_9

Pencarian breadth-first memiliki sifat-sifat berikut:

Lengkap: Ya;

Optimal: Ya;

Kompleksitas Waktu: $O(b^d)$, dimana b adalah faktor percabangan dan d adalah kedalaman dari solusi;

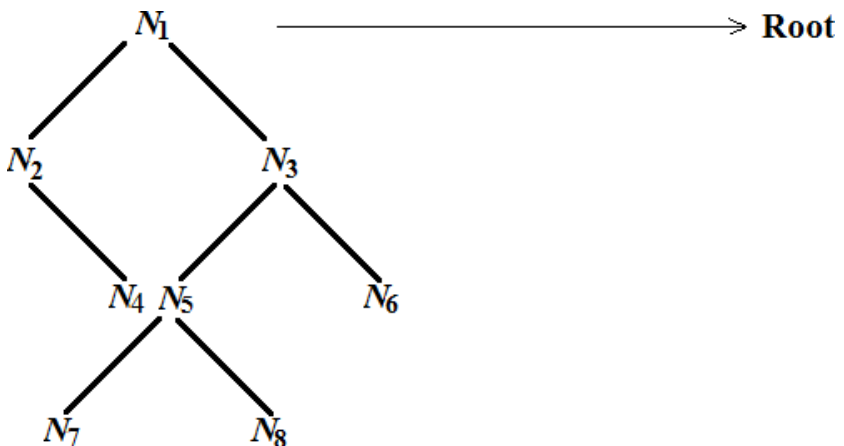
Kompleksitas Ruang: $O(b^d)$.

Pencarian breadth-first selalu menemukan solusi dari masalah jika ada. Jika masalah memiliki lebih dari satu solusi, maka algoritma menemukan solusi dalam urutan naik dari biayanya yang berarti bahwa solusi termurah akan ditemukan terlebih dahulu. Karena strategi pencarian ini bersifat traversal, situasi di mana algoritma akan menjelajahi cabang yang kedalamannya tidak terbatas tidak akan pernah terjadi.

Konsumsi waktu sangat tinggi untuk strategi breadth-first. Faktor ini dapat mempengaruhi performa algoritma jika jumlah faktor percabangan bertambah. Faktanya, jika jumlah faktor percabangan meningkat, kompleksitas waktu meningkat secara eksponensial. Strategi ini sangat memakan ruang. Sebenarnya, ini membutuhkan banyak ruang dalam hal memori untuk menyimpan node yang dihasilkan pada kedalaman tertentu sebelum kedalaman berikutnya dapat dilanjutkan.

5.5.2 Depth-first search

Ide dari algoritma pencarian ini adalah masuk sedalam mungkin ke dalam pohon pencarian tetapi dari satu node tetangga ke node tetangga lainnya sebelum proses *backtracking* dimulai. Algoritma depth-first search disebut juga dengan algoritma Last In First Out (LIFO) atau tumpukan (stack). Gambar 5.7 menunjukkan pohon traversal depth-first:



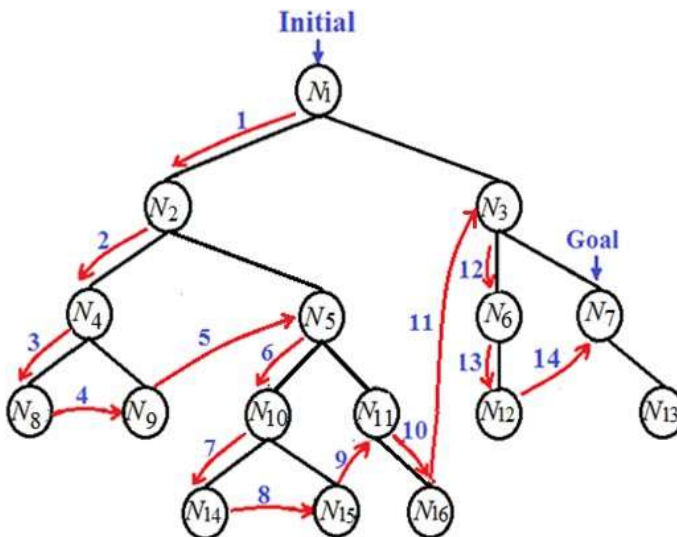
Gambar 5.7. Pohon traversal depth-first

Pencarian *depth-first* mencoba masuk lebih dalam dari sebuah simpul sebelum kembali menjelajahi simpul tetangga. Urutan node yang didapatkan setelah penerapan algoritma

depth-search untuk mengeksplorasi pohon yang direpresentasikan pada Gambar 5.7 adalah $\{N_1, N_2, N_4, N_3, N_5, N_7, N_8, N_6\}$.

Algoritme pencarian depth-first ditentukan dan diimplementasikan tergantung pada apakah kita menggunakan versi pencarian pohon atau versi pencarian graf. Versi sebelumnya tidak lengkap sedangkan yang terbaru sudah lengkap. Baik versi pencarian pohon dan pencarian graph dari pencarian *depth-first* tidak optimal. Selain itu, versi pencarian graf dari pencarian depth-first memiliki kompleksitas waktu yang dibatasi oleh ukuran ruang keadaan. Ukuran ini bisa tak terhingga dalam beberapa kasus versi tree-search dari depth-first search yang memiliki kompleksitas urutan $O(b^d)$.

Gambar 5.8 menunjukkan representasi graf urutan simpul pencarian depth-first dari simpul awal N_1 ke simpul tujuan N_7 untuk graf yang direpresentasikan dalam Gambar 5.5.



Gambar 5.8. Representasi graf dari urutan simpul dalam pencarian depth-first dari simpul awal N_1 ke simpul tujuan N_7

Pencarian depth-first memiliki sifat-sifat berikut:

Lengkap: Tidak;

Optimal: Tidak;

Kompleksitas Waktu: $O(b^m)$, dimana b adalah faktor percabangan dan m adalah kedalaman maksimum dari pohon;

Kompleksitas Ruang: $O(bm)$.

Strategi pencarian depth-first tidak membutuhkan banyak memori seperti yang dibutuhkan oleh strategi pencarian breadth-first. Sebenarnya, kompleksitas ruang itu linier. Strategi pencarian ini dibatasi oleh waktu, tetapi tidak terbatas pada ruang. Oleh karena itu, selalu ada ruang yang cukup untuk menyimpan node yang dihasilkan meskipun ini membutuhkan banyak waktu untuk diproses. Semakin dangkal kedalaman tujuan, semakin sedikit kompleksitas ruang dan kompleksitas waktu.

Faktor percabangan dapat sangat besar dalam strategi ini, dan situasi dapat terjadi bahwa algoritme dapat mengikuti jalur jika memiliki banyak simpul yang tak terhingga. Tujuannya tidak dijamin selalu ditemukan oleh strategi ini jika memang ada. Strategi ini tidak memberikan jaminan untuk selalu menemukan solusi terbaik dari masalah.

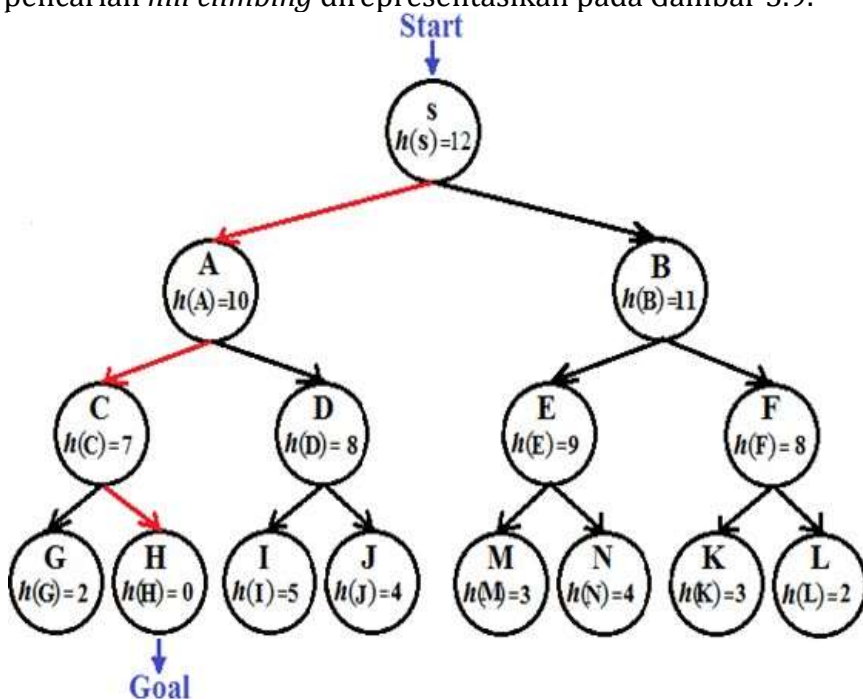
5.6 Metode pencarian informasi (heuristik)

Subbab ini berfokus pada kelompok pencarian yang dikenal sebagai pencarian informasi atau pencarian heuristik. Perbedaan utama antara pencarian informasi (heuristik) dan pencarian tanpa informasi adalah bahwa di luar masalah yang terdefinisi dengan baik, metode pencarian informasi menggunakan beberapa spesifikasi masalah sedangkan strategi pencarian tanpa informasi melakukan pencarian dalam mode buta, tanpa pengetahuan sebelumnya atau spesifikasi masalah.

Spesifikasi masalah dalam pencarian heuristik membantu menemukan solusi masalah secara efisien.

5.6.1 Hill climbing search

Hill climbing adalah teknik pencarian heuristik dan lokal, dan pada tahap awal memilih node awal, dan kemudian pindah ke posisi terdekat secara heuristik yang mengarah ke tujuan. Proses ini berlanjut hingga tidak ada lagi posisi yang lebih baik untuk dipindahkan. Representasi graf dari urutan simpul pada pencarian *hill climbing* direpresentasikan pada Gambar 5.9.



Gambar 5.9. Representasi graf dari urutan simpul dalam pencarian hill-climbing

Dari Gambar 5.9 terlihat bahwa jalur pencarian setelah menggunakan strategi *hill climbing* adalah $S \rightarrow A \rightarrow C \rightarrow H$. Pencarian hill climbing memiliki sifat berikut:

Lengkap: Tidak;

Optimal: Tidak;

Kompleksitas waktu: $O(\infty)$;

Kompleksitas Ruang: $O(b)$, di mana b adalah faktor percabangan.

Dengan menggunakan teknik pencarian hill climbing, dimungkinkan untuk memperluas tepat satu simpul penerus pada satu waktu terlepas dari jumlah penerus yang dimiliki simpul tersebut, dan tidak ada upaya untuk mundur. Dengan kata lain, teknik hill climbing tidak pernah mencoba mempertimbangkan semua simpul yang ditolak. Tampaknya pencarian hill climbing kurang berupaya untuk mencapai keadaan tujuan, tetapi tidak ada jaminan bahwa kualitas penyelesaian masalah akan selesai.

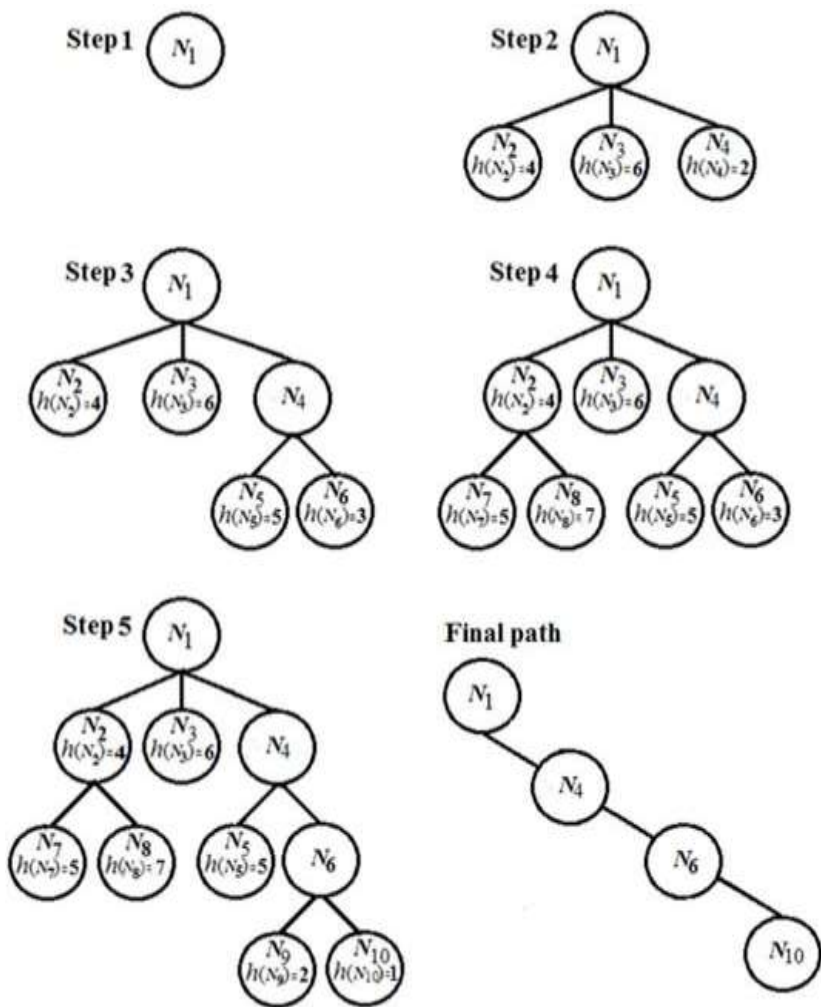
5.6.2 Best-first search

Pencarian *best-first* dibuat dengan menggabungkan teknik pencarian *breadth-first* dan *depth-first*. Karena alasan inilah pencarian *best-first* mendapatkan keuntungan dari teknik pencarian *breadth-first* dan *depth-first*. Pencarian *best-first* tidak mempertimbangkan semua kemungkinan jalur dalam graf, dan strategi ini menemukan jalur yang tidak memiliki loop.

Dalam pencarian best-first, fungsi evaluasi $f(n)$ digunakan untuk setiap node untuk mengetahui keinginan dari sebuah node; jadi node yang tidak diperluas yang paling menjanjikan dan diinginkan diperluas. Jadi simpul yang akan dipilih harus memiliki nilai fungsi heuristik $f(n)$ terendah.

Pada tahap pertama pencarian best-first, daftar OPEN dengan simpul awal dan daftar CLOSED dengan himpunan kosong ditentukan. Jika daftar OPEN tidak memiliki *successor*, maka tidak ada solusi. Jika ada *successor* dari node awal, node dengan nilai heuristik terbaik (minimum) N dipilih. Jika simpul N adalah simpul tujuan maka jalur dari simpul awal ke simpul tujuan ditentukan. Jika tidak, setiap simpul *successor* langsung dari N , jika tidak ada dalam daftar OPEN atau CLOSED, ditambahkan ke daftar OPEN dan simpul N ditetapkan sebagai simpul induk. Setelah itu simpul N dihapus dari daftar OPEN, dan ditambahkan ke daftar CLOSED. Proses ini akan berlanjut hingga simpul tujuan ditemukan atau daftar OPEN kosong.

Karakteristik yang paling penting dari best-first search adalah bahwa setelah node penerus yang paling menjanjikan dipilih dan pergerakan dilakukan menuju penerus ini, node lain dapat diperhitungkan untuk dikunjungi nanti jika diperlukan. Perlu juga dicatat bahwa fungsi heuristik terkadang dapat mengarah ke jalur yang mahal. Pada Gambar 5.10 representasi graf dari langkah-langkah contoh best-first search diberikan. Node awal dan tujuan dari graf tak berarah masing-masing adalah N_1 dan N_{10} .



Gambar 5.10. Representasi graf dari langkah-langkah contoh best-first search

Pencarian best-first memiliki sifat berikut:

Lengkap: Tidak;

Optimal: Tidak;

Kompleksitas Waktu: $O(b^m)$; di mana b adalah faktor percabangan, dan m adalah kedalaman maksimum dari pohon;

Kompleksitas Ruang: $O(b^m)$.

5.6.3 A* search

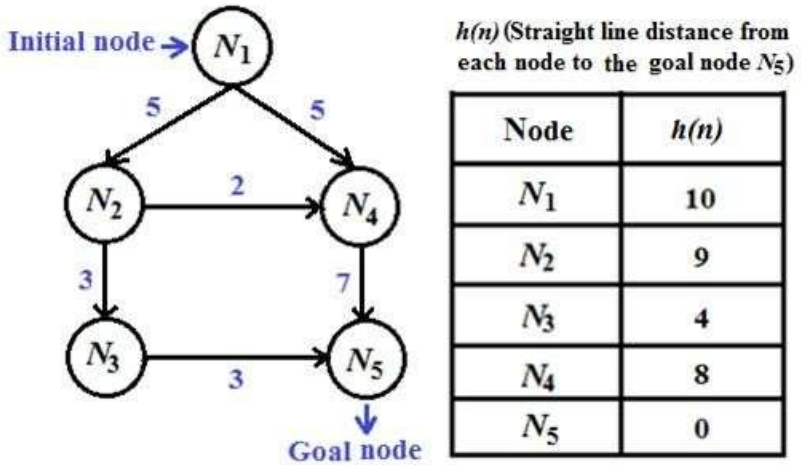
Pencarian A* adalah strategi pencarian informasi lainnya. Pencarian A* sebenarnya adalah strategi pencarian best-first yang paling populer yang diadopsi oleh para ilmuwan dan peneliti.

Pencarian A* adalah algoritma pencarian best-first, tetapi bukan algoritma pencarian greedy best-first (jenis lain dari best-first). Fungsi evaluasi $f(n)$ dalam strategi pencarian A* direpresentasikan sebagai $f(n) = g(n) + h(n)$, di mana n adalah simpul terakhir dari jalur, $g(n)$ adalah biaya jalan dari awal simpul ke simpul n , dan $h(n)$ adalah heuristik digunakan untuk memperkirakan biaya dari node n ke simpul tujuan.

Menggabungkan $g(n)$ dan $h(n)$ mengarah pada estimasi $f(n)$ yang sebenarnya adalah fungsi biaya optimal yang digunakan untuk mencapai solusi. Pencarian biaya uniform dan pencarian best-first adalah kasus khusus dari pencarian A*. Jika heuristik $h(n)$ sama dengan 0 dalam $f(n) = g(n) + h(n)$, maka kita dapatkan $f(n) = g(n)$ yaitu digunakan dalam pencarian biaya uniform. Jika $g(n)$ sama dengan 0 dalam $f(n) = g(n) + h(n)$, maka kita dapatkan $f(n) = h(n)$ yang digunakan dalam pencarian best-first.

Keuntungan besar dari strategi pencarian A* yang tidak ada dalam strategi pencarian pertama yang tamak adalah bahwa pencarian A* selesai serta optimal. Oleh karena itu dapat dikatakan bahwa teknik pencarian A* memenuhi tujuan dari strategi pencarian informasi. Gambar 5.11

merepresentasikan graf untuk pencarian A* dengan jarak garis lurus dari setiap node ke node tujuan.



Gambar 5.11. Representasi graf untuk pencarian A* dengan jarak garis lurus dari setiap node ke node tujuan

Untuk menemukan jalur terpendek dari simpul awal (awal) N_1 ke simpul tujuan N_5 :

$$N_1, g + h = 0 + 10 = 10$$

$$N_1 \rightarrow N_2, g + h = (0 + 5) + 9 = 5 + 9 = 14$$

$$N_1 \rightarrow N_4, g + h = (0 + 5) + 8 = 5 + 8 = 13$$

$$N_1 \rightarrow N_2 \rightarrow N_3, g + h = (5 + 3) + 4 = 8 + 4 = 12$$

$$N_1 \rightarrow N_2 \rightarrow N_4, g + h = (5 + 2) + 8 = 7 + 8 = 15$$

$$N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_5, g + h = (5 + 3 + 3) + 0 = 11$$

$$N_1 \rightarrow N_2 \rightarrow N_4 \rightarrow N_5, g + h = (5 + 2 + 7) + 0 = 14$$

$$N_1 \rightarrow N_4 \rightarrow N_5, g + h = (5 + 7) + 0 = 12$$

Karena $N_1 \rightarrow N_2 \rightarrow N_3 \rightarrow N_5$ adalah jalur terpendek di antara jalur yang tersedia dari simpul awal N_1 ke simpul tujuan N_5 , jalur ini optimal.

Strategi pencarian A* mencakup beberapa kekurangan yang ditemukan dalam pencarian pertama terbaik dan pencarian pertama terbaik rakus. Perbedaan antara strategi pohon ini sebagian besar diamati pada sifatnya berikut:

Pencarian A* memiliki sifat berikut:

Lengkap: Ya;

Optimal: Ya;

Kompleksitas Waktu: $O(b^d)$; di mana b adalah faktor percabangan, dan d adalah kedalaman dari solusi;

Kompleksitas Ruang: $O(b^d)$.

Teknik pencarian A* juga digunakan untuk memecahkan permainan 8-puzzle. Game ini ditemukan oleh Loyd pada tahun 1870. Game 8-puzzle ini adalah game terkenal yang digunakan untuk mengevaluasi dan memamerkan kekuatan Kecerdasan Buatan pada umumnya dan metode pencarian pada khususnya.

DAFTAR PUSTAKA

- Ashwani, C., & Manu, S. 2014. 'Searching and Optimization Techniques in Artificial Intelligence: A Comparative Study & Complexity Analysis', *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 3(1), pp. 866–871.
- Goebel, D. P. A. M. R. 1998. *Computational Intelligence. A Logical Approach*. New York Oxford: Oxford University Press.
- Kuruvilla, M., & Tabassum, M. 2014. 'Experimental Comparison of Uninformed and Heuristic AI Algorithms for N Puzzle and 8 Queen Puzzle Solution', *International Journal of Digital Information and Wireless Communications (IJDIWC)*, 4(1), pp. 143–154.
- Luger, G. F. 2009. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Sixth edit. Pearson Education.
- Munir, R. 2015. *Breadth/Depth First Search (BFS/DFS) Bahan Kuliah IF2211 Strategi Algoritmik*.
- Nilsson, N. J. 1998. *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers Inc.
- PRASHANTA KUMAR PATRA (no date) *Lecture Notes on Artificial Intelligence*. Available at: [https://www.cet.edu.in/noticefiles/271_AI Lect Notes.pdf](https://www.cet.edu.in/noticefiles/271_AI%20Lect%20Notes.pdf).
- Russell, Stuart J. and Norvig, P. 2010. *Artificial Intelligence. A Modern Approach*. Third Edit. Prentice Hall.
- Stern, R., Kiesel, S., Puzis, R., Felner, A., & Ruml, W. 2015. 'Max Is More than Min: Solving Maximization Problems with Heuristic Search', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 4324–4330.

Zhou, R., & Hansen, E. . 2009. 'Combining Breadth-First and Depth-First Strategies in Searching for Treewidth', in *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*. Pasadena, California, USA, pp. 640–645.

BAB 6

REASONING : PROPOSITIONAL LOGIC

Oleh Elisawati

6.1 Pegantar Reasoning

Teknik *Reasoning* (Penalaran) adalah suatu teknik untuk menyelesaikan suatu masalah menggunakan bahasa formal atau logic yang di pahami komputer ke dalam basis pengetahuan guna untuk menemukan solusi.

6.2 Propositional Logic

Propositional logic juga di kenal sebagai logika sentensial dan logika pernyataan merupakan cabang logika yang mempelajari cara menggabungkan atau memodifikasikan seluruh proposisi dan pernyataan.

Logika Proposisi (Pernyataan) adalah merupakan kalimat deklaratif yang mempunyai nilai benar atau salah saja (Salah satu). Tidak sekaligus mempunyai nilai kebenaran keduanya. Nilai benar atau salah sebuah proposisi di sebut nilai kebenaran proposisi. Proposisi biasanya sering di lambang huruf kecil yang disebut proposisi atomik. (Sutojo, Mulyanto and Suhartono, 2011)

Seperti contoh di bawah :

1. George W. Bush adalah presiden Amerika Serikat
2. Paris adalah ibu kota Perancis
3. Setiap orang yang lahir pada hari Senin memiliki rambut ungu

Terkadang sebuah pernyataan dapat berisi satu atau lebih pernyataan lain. Seperti contoh di bawah ini :

- Ganymede adalah bulan jupiter atau Ganymede adalah bulan saturnus

Sementara kalimat majemuk di atas merupakan pernyataan yang kalimat pertama “Ganymede adalah bulan jupiter” adalah benar dan kalimat kedua “Ganymede adalah bulan saturnus” adalah salah.

Proposisi juga dapat menggunakan operator logika sebagai berikut :

- a. Konjungsi merupakan dua pernyataan yang di hubungkan dengan (DAN / AND), dan biasanya di simbolkan dengan simbol ($\wedge$). Bisa di lihat di tabel kebenaran di bawah ini :

Tabel 6.1. Tabel Konjungsi (*AND*)

p	q	$p \wedge q$
True	True	True
True	False	False
False	True	False
False	False	False

Dari tabel di atas dapat di lihat sifat Konjungsi adalah jika kedua pernyataan tersebut bernilai True maka hasilnya True.

Contoh Konjungsi :

Minggu adalah hari libur **dan** sekarang hari minggu

- b. Disjungsi merupakan dua pernyataan yang dihubungkan dengan (ATAU / OR), dan disimbol dengan simbol ($\vee$).

Tabel 6.2. Tabel Disjungsi (OR)

p	q	$p \vee q$
True	True	True
True	False	True
False	True	True
False	False	False

Dari tabel di atas dapat di lihat sifat Disjungsi adalah jika kedua pernyataan tersebut bernilai True atau salah satu pernyataan tersebut bernilai True maka hasilnya True.

Contoh Konjungsi :

Minggu adalah hari libur **atau** sekarang hari minggu

- c. Negasi merupakan pernyataan yang dihubungkan dengan (NOT / TIDAK) atau disimbolkan dengan simbol ($\sim$).

Tabel 6.3. Tabel Negasi (NOT)

p	$\sim p$
True	False
False	True

Dari tabel di atas dapat dilihat sifat dari negasi adalah jika pernyataan awalnya True maka kalau di negasikan menjadi pernyataan yang False.

Contoh Negasi :

Tidak benar bahwa Minggu adalah hari libur

- d. Implikasi merupakan pernyataan yang dihubungkan dengan (IF-THEN / JIKA-MAKA) atau disimbolkan dengan simbol ($\rightarrow$).

Tabel 6.4. Tabel Implikasi (*IF-THEN*)

p	q	$p \rightarrow q$
True	True	True
True	False	False
False	True	True
False	False	True

Dari tabel di atas dapat dilihat sifat dari implikasi adalah bernilai False jika pernyataan p true dan q false.

Contoh Implikasi :

Jika Minggu adalah hari libur **maka** sekarang hari minggu

- e. Equivalensi merupakan pernyataan yang dihubungkan dengan (*IF AND ONLY IF* / JIKA DAN HANYA JIKA) atau disimbolkan dengan simbol ($\leftrightarrow$).

Tabel 6.5. Tabel Equivalensi (*IF AND ONLY IF*)

p	q	$p \leftrightarrow q$
True	True	True
True	False	False
False	True	False
False	False	True

Dari tabel di atas dapat dilihat sifat dari implikasi adalah bernilai true jika kedua pernyataan tersebut bisa bernilai benar atau bernilai salah.

Contoh Equivalensi:

Sekarang hari minggu **jika dan hanya jika** Minggu adalah hari libur

Tautologi adalah suatu pernyataan gabungan yang selalu mempunyai nilai benar. Suatu kontradiksi adalah sebuah pernyataan gabungan mempunyai nilai salah, sedangkan

contingent merupakan suatu pernyataan yang bukan termasuk tautologi ataupun kontradiksi. (Desiani and Arhami, 2006)

Argumen pernyataan valid dan invalid adalah pernyataan yang mempunyai premise dari proposisi pertama, kedua, ketiga dan seterusnya serta menghasilkan proposisi lain yang disebut dengan konklusi.

Contoh pernyataan valid dapat di gambarkan sebagai berikut :

Jika hari ini hujan maka udara dingin

Hari ini hujan

∴ Udara dingin

Misalkan :

s = Hari ini hujan

t = Udara dingin

Sehingga bisa di tulis sebagai :

$s \rightarrow t$ (Premise 1)

s (Premise 2)

∴ t (Konklusi)

Tabel 6.6. Tabel Tautologi atau Bukan

s	t	$s \rightarrow t$	$((s \rightarrow t) \wedge s)$	$((s \rightarrow t) \wedge s) \rightarrow t$
True	True	True	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	True

Tampak tabel di atas merupakan tautologi karena argumennya bernilai valid.

Penerapan contoh Argumen yang invalid seperti di bawah ini :

Jika hari ini hujan maka udara dingin
 Udara dingin

∴ Hari ini hujan

Misalkan :

s = Hari ini Hujan

t = Udara dingin

Sehingga bisa di tulis sebagai :

s → t	(Premise 1)
t	(Premise 2)
∴ s	(Konklusi)

Tabel 6.7. Tabel Tautologi atau Bukan

s	t	s→t	((s→t)∧t)	((s→t)∧t)→s
True	True	True	True	True
True	False	False	False	True
False	True	True	True	False
False	False	True	False	True

Pernyataan di atas bukan tautologi artinya pernyataan tersebut invalid karena terdapat nilai kebenaran yang False.

6.2.1 Inferensi logika proposisi

Inferensi adalah merupakan salah satu cara untuk menarik sebuah kesimpulan berdasarkan premis atau pernyataan yang bernilai valid.

Tabel 6.8. Hukum Inferensi Proposisi

Hukum Inferensi	Skema	
1. Hukum Detasemen	$\begin{array}{l} s \rightarrow t \\ s \\ \hline \therefore t \end{array}$	
2. Hukum Kontrapositif	$\begin{array}{l} s \rightarrow t \\ \hline \therefore \sim t \rightarrow \sim s \end{array}$	
3. Hukum Modus Tollens	$\begin{array}{l} s \rightarrow t \\ \sim t \\ \hline \therefore \sim s \end{array}$	
4. Hukum silogisme	$\begin{array}{l} s \rightarrow t \\ t \rightarrow r \\ \hline \therefore s \rightarrow r \end{array}$	
5. Hukum Inferensi Disjungsi	$\begin{array}{l} s \vee t \\ \sim s \\ \hline \therefore t \end{array}$	$\begin{array}{l} s \vee t \\ \sim t \\ \hline \therefore s \end{array}$
6. Hukum Negasi	$\begin{array}{l} \sim(\sim s) \\ \hline \therefore s \end{array}$	
7. Hukum Demorgan	$\begin{array}{l} \sim(s \wedge t) \\ \hline \therefore \sim s \vee \sim t \end{array}$	$\begin{array}{l} \sim(s \vee t) \\ \hline \therefore \sim s \wedge \sim t \end{array}$
8. Hukum Simplifikasi	$\begin{array}{l} s \wedge t \\ \hline \therefore s \end{array}$	$\begin{array}{l} s \wedge t \\ \hline \therefore t \end{array}$
9. Hukum Konjungsi	$\begin{array}{l} s \\ t \\ \hline \therefore s \wedge t \end{array}$	
10. Hukum Penambahan Disjungsi	$\begin{array}{l} s \\ \hline \therefore s \vee t \end{array}$	
11. Hukum Argumen Konjungtif	$\begin{array}{l} \sim(s \wedge t) \\ s \\ \hline \therefore \sim t \end{array}$	$\begin{array}{l} \sim(s \wedge t) \\ t \\ \hline \therefore \sim s \end{array}$

Contoh penerapan argumen dengan menggunakan hukum inferensi proposisi sebagai berikut :

Saat Anda berangkat ke kampus, anda baru menyadari bahwa tidak membawa tugas. Tugas tersebut terakhir di kumpulkan pada hari ini, jika anda ingin mendapatkan nilai

maka anda harus membawa tugas tersebut. Tetapi ternyata anda lupa menaruh tugas tersebut. Ada beberapa fakta yang anda pastikan kebenarannya.

1. Tugas tidak ada di dalam tas
2. Jika aku membuka tas Maka aku bisa memastikan tugas tersebut di dalam tas atau tidak.
3. Jika Tugas di meja makan, maka aku pasti sudah melihatnya ketika mau berangkat ke kampus.
4. Jika Tugas tidak ada di dalam tas maka aku pasti telah membuka tas tersebut.
5. Jika aku melihat tugas saat berangkat ke kampus, maka pastilah tugas kuletakkan di tas
6. Aku tidak bisa memastikan bahwa tugas tersebut ada di dalam tas atau tidak.

Berdasarkan fakta di atas, tentukan di mana letak Tugas tersebut. Untuk memudahkan pemahaman penggunaan metode inferensi, maka kalimat di atas terlebih dahulu di tuliskan kedalam simbol-simbol sebagai berikut :

p : Tugas ada di dalam tas

q : aku membuka tas

r : aku bisa memastikan tugas tersebut di dalam tas atau tidak

s : tugas di meja makan

t : aku melihatnya ketika mau berangkat ke kampus

Dengan simbol-simbol tersebut maka fakta-fakta di atas dapat di tulis sebagai berikut :

1. $\sim p$
2. $q \rightarrow r$
3. $s \rightarrow t$
4. $\sim p \rightarrow q$
5. $t \rightarrow p$
6. $\sim r$

Hukum inferensi dapat dilakukan sebagai berikut :

$$\begin{array}{rcl} 1. & q \rightarrow r & (2) \\ & \underline{\sim r} & (6) \end{array}$$

$\therefore \sim q$ Aku tidak membuka tas (Konklusi 1)

$$\begin{array}{rcl} 2. & t \rightarrow p & (5) \\ & \underline{\sim p} & (1) \end{array}$$

$\therefore \sim t$ Aku Tidak Melihatnya ketika berangkat ke kampus
(Konklusi 2)

$$\begin{array}{rcl} 3. & s \rightarrow t & (3) \\ & \underline{\sim t} & (\text{Konklusi 2}) \end{array}$$

$\therefore \sim s$ Tas tidak ada di meja makan (Konklusi 3)

$$\begin{array}{rcl} 4. & \sim p \rightarrow q & (4) \\ & \underline{\sim q} & (\text{Konklusi 1}) \end{array}$$

$\therefore p$ Tugas ada di dalam tas (Konklusi 4)

Kesimpulan : Tugas ada di dalam tas

DAFTAR PUSTAKA

- Desiani, A. and Arhami, M. 2006. *Konsep Kecerdasan Buatan*. Edited by D. Hardjono. Yogyakarta: Andi Yogyakarta.
- Sutojo, T., Mulyanto, E. and Suhartono, V. 2011. *Kecerdasan Buatan*. Edited by B. Rini W. Yogyakarta: Andi Yogyakarta.
- <https://nuraimanusm.wordpress.com>
- <https://lmsspada.kemdikbud.go.id/mod/resource/view.php?id=91146>
- <https://ocw.upj.ac.id/files/Textbook-CPS105-Modul-Dasar-Logika-Matematika.pdf>
- <https://pustaka.ut.ac.id/lib/wp-content/uploads/pdfmk/MSIM4103-M1.pdf>

BAB 7

REASONING : FIRST ORDER LOGIC

Oleh Muhamad Faza Almaliki

7.1 Definisi First Order Logic

Logika orde pertama merupakan bahasa resmi dan dimanfaatkan pada bidang matematika, bahasa, filsafat, serta bidang komputer. Logika orde pertama juga biasa dikenal sebagai kalkulus predikat. Logika ini dimanfaatkan dalam menggambarkan permasalahan yang sulit digambarkan oleh proposisi. Logika orde pertama dapat memberi gambaran fakta dalam pernyataan yang dibentuk dengan baik. Kalkulus predikat dapat menganalisis beberapa kalimat dalam bentuk argumen dan subjek dengan cara yang berbeda, sehingga kalkulus predikat dapat dimanfaatkan dalam menyelesaikan masalah pada berbagai kondisi umum (Dawis dkk, 2022).

Penggunaan logika orde pertama dalam menggambarkan fakta merupakan teknik dasar yang telah lama digunakan guna bertujuan menyandikan bahasa natural menjadi bentuk resmi. Berdasarkan penggunaan logika orde pertama, diharapkan fakta (dan juga pertanyaan) dapat disajikan dengan akurat sesuai konteks yang diinginkan, sehingga hasil akhir yang diharapkan memiliki tingkat validitas yang sangat tinggi. Mengenai logika predikat, perlu dicatat yakni pada tahun 1879, filsuf yang berasal dari Jerman yang bernama Gottlob Frege menciptakan risalah yang luar biasa, yang berjudul the Begriffsschrift ("Concept Script"). Monograf luar biasa ini dijadikan sebagai sumber dari teori logika modern. Namun, dalam risalah Friege ini terdapat banyak celah pada beberapa bagian serta kesalahan pada notasi. Meskipun

demikian, temuan Frege ini masih diakui. Selain temuan Frege, rumus dari logika predikat yang umum dipergunakan saat ini adalah *first order logic* atau lebih dikenal dengan sebutan kalkulus predikat seperti yang didokumentasikan dalam *Principle of Theoretical Logic* yang ditulis oleh David Hilbert dan Wilhelm Ackerman pada tahun 1928. Logika orde pertama dalam hal ini adalah dasar pendiri logika matematika modern.

Jika logika proporsional memperhitungkan dunia berisi kebenaran, maka logika orde pertama memperkirakan bahwa dunia berisi “

Object : orang, mobil, huruf, warna, permainan *football*, perang dan lain-lain.

Relation : kuning, segitiga, perdana, saudara, lebih kecil dari, bagian dari, datang antara dan lain-lain.

Function : ibu, teman, satu lebih dari, dikurang dan lain-lain.

Bahasa pada logika orde pertama dibuat dalam hubungan antara objek dan relasi. Hal ini sangat penting di dalam matematika, bidang filsafat, dan kecerdasan buatan karena pada bidang ini merupakan pemikiran yang sangat berguna dalam berurusan dengan objek dan relasi diantara mereka (Susatyono, 2021).

7.2 Simbol First Order Logic

Logika orde pertama menggunakan sejumlah simbol yang menggambarkan maksud dari pengetahuan yang dituliskan. Adapun lambang yang memberitahukan substansi dari pengetahuan yang ditulis yaitu sebagai berikut .

a. $\forall$ = untuk semuanya

Contoh : setiap orang, semua orang, kapanpun, dan sebagainya. Apabila dijelaskan dalam bentuk kalimat, misalnya :

“Setiap mahasiswa di Universitas ABC adalah rajin”

Maka penggunaan universal quantifier-nya adalah :

$\forall x : \text{at}(x, \text{ABC}) \Rightarrow \text{Diligent}(x)$

Contoh lainnya :

$\forall x : \text{Queen}(x) \Rightarrow \text{Person}(x)$

Memiliki arti : “Untuk setiap x , apabila x adalah Queen, maka x adalah person”.

b. $\exists$ = ada sebagian

Contoh : seseorang, pada waktu tertentu, dan sebagainya.

Apabila dijelaskan dalam bentuk kalimat, misalnya :

“Ada seseorang yang membenci seseorang di dalam dunia ini”

Maka penggunaan existential quantifier-nya adalah :

$\exists x : \forall y \text{ Hate}(x, y)$

Contoh lainnya :

$\exists x : \text{Mahkota}(x) \wedge \text{onHead}(x, \text{Smith})$

Memiliki arti

“Smith memiliki sebuah mahkota pada kepalanya”

c. $\Rightarrow$ = Implikasi

Contoh : apabila ... maka ...

d. $\Leftrightarrow$ = Ekuivalen (Biconditional)

Contoh : apabila ... dan ... hanya ... apabila ...

e. $\neg$ = Not / negasi / ingkaran

f. $\vee$ = Or / atau / disjungsi

g. $\wedge$ = And / dan / konjungsi

Adapun contoh sederhana proses penukaran dari bahasa natural menuju bahasa *first order logic* sebagai berikut

1. Ahmad adalah seorang mahasiswa
Mahasiswa(Ahmad)
2. Semua mahasiswa di Jurusan Teknik Informatika adalah anggota dari Himpunan Mahasiswa Jurusan Teknik Informatika
 $\forall x: \text{mahasiswa}(x) \wedge \text{Teknik Informatika}(x) \Rightarrow \text{anggota}(x, \text{Himpunan Mahasiswa Jurusan Teknik Informatika})$
3. Mahasiswa yang tidak masuk kuliah maka tidak ikut belajar
 $\exists x: \text{mahasiswa}(x) \neg \text{masuk}(x, \text{kuliah}) \Rightarrow \text{ikut}(x, \text{belajar})$
4. Ada mahasiswa Universitas ABC yang memakai topi
 $\exists x: \text{mahasiswa}(x) \wedge \text{memakai}(x, \text{topi})$

Logika orde pertama lebih mementingkan sebuah penegasan kebenaran suatu kebenaran atau penarikan kesimpulan. Mengingat fakta atau premis, logika orde pertama dimanfaatkan dalam memeriksa hasil fakta suatu pernyataan. Pengujian ini dilakukan dengan cara mengerjakan langsung dari fakta / premis yang diketahui.

7.3 Representase Semantik

Penggambaran suatu kebenaran pada model logika orde pertama memiliki beberapa keunggulan dibanding logika proposisi. Pada logika proposisi suatu kalimat digambarkan dengan simbol yang menggantikan seluruh kalimat secara utuh. Misalnya: 'ibu minum segelas air', akan dilambangkan dengan seluruh klausa 'I'. Keterbatasan penggambaran tersebut merupakan bagian kebenaran yang berfungsi menjadi jawaban pada sebuah pertanyaan. Contohnya, jika sistem berkeinginan mengambil jawaban 'ibu' sebagai tanggapan atas pertanyaan 'Siapa yang minum segelas air?'.

Dengan logika orde pertama, penggambaran dapat digunakan dengan memberi predikat yang menggambarkan adanya hubungan antara kebenaran yang bersifat atomik (variabel) pada kalimat. Logika orde pertama juga menyediakan kemampuan untuk memenuhi penggambaran berdasarkan kuantifikator yang ada, yang menggambarkan kisaran predikat dalam kalimat bahasa natural. Misalnya kalimat 'ibu minum segelas air', sebenarnya pada kalimat tersebut dijelaskan bahwa 'terdapat seorang ibu yang meminum air menggunakan sejumlah gelas sebagai wadah air tersebut', atau dengan anggapan bahwa ada kebenaran yang bersifat atomik 'ibu' yang melakukan fakta tindakan 'minum' sesuatu, yaitu 'air', yang dimasukkan ke dalam 'gelas'. Ketika digambarkan berdasarkan logika orde pertama, kalimat 'ibu minum segelas air', digambarkan dengan ' $\exists x.(air(x) \wedge minum(ibu, x))$ '. Pada pemberian kuantifikator eksistensial ' $\exists$ ', fakta sejumlah air tertentu diukur dalam kondisi 'minum' yang dilakukan 'ibu'. Bentuk kuantifikator lain pada logika orde pertama adalah universalitas, yang disimbolkan dengan ' $\forall$ '. Kuantifikator ini dapat dimanfaatkan dalam menjelaskan generalisasi tentang suatu kebenaran. Misalnya jika ada kalimat 'semua ibu minum segelas air', maka dapat digambarkan ke dalam logika orde pertama menjadi ' $\forall x \exists y.((ibu(x))(air(y) \wedge minum(x, y)))$ ' atau ' $\exists x \forall y.((air(x) \wedge ibu(y))(minum(x, y)))$ '. Perhatikan bahwa dalam logika orde pertama, makna ganda dalam memaknai sebuah kalimat tetap dapat digambarkan. Berdasarkan hal tersebut, menciptakan dua pemahaman, pertama menjelaskan bahwa 'semua ibu meminum air dari gelas yang berbeda', sedangkan yang kedua menjelaskan bahwa 'semua ibu meminum air dari sebuah gelas yang sama'. Persoalan makna ganda ini bergantung pada lingkup kuantifikator dalam rumus logika orde pertama. Guna memberi penjelasan kuantifikator diperlukan adanya

pemahaman terhadap model, domain dan rumus pada sebuah penggambaran logika orde pertama.

7.4 Model, Domain, dan Formula

Sebuah model merupakan suatu keadaan yang menggambarkan fokus pembicaraan. Kosakata diperlukan untuk membuat suatu model, yaitu daftar istilah yang merumuskan model tersebut. Dalam kosakata meliputi topik pembicaraan dan bahasa (simbol) yang dimanfaatkan dalam pembicaraan. Sebagai contoh kalimat 'ibu dan ani minum segelas teh', memiliki kosakata antara lain : $\{(ibu,1), (ani,0), (makan,2), (nasi,1)\}$. Pada kosakata ini akan tergambarkan bagaimana hubungan antara kebenaran atau variabel yang satu dengan lainnya di dalam penggambaran. Perlu dibedakan antara kebenaran (sebuah konstanta / *non-binding variable*), dengan parameter yang dapat menampung sebuah kebenaran (*binding* variiable). Misalnya, hubungan 'makan' mendeskripsikan bahwasanya kegiatan itu akan terjadi ketika menyangkutkan dua konstanta (hubungan biner yang memiliki arity 2). Angka 1 mendeskripsikan bahwa terjadi hubungan tunggal (arity 1), yang dapat diisi (*binding*) dengan sebuah ketetapan. Angka 0 mendeskripsikan suatu ketetapan, dan bukan sebuah hubungan.

7.5 Contoh Kasus Implementasi First Order Logic

Sistem tanya jawab yaitu suatu sistem yang berusaha untuk melakukan inferensi berdasarkan suatu pertanyaan yang diajukan berdasarkan bahasa alami. Perbedaannya dengan mesin pencari yang menyediakan kumpulan dokumen hasil pencarian, sebuah sistem tanya jawab harus memberikan hasil akhir kepada pemakai.

Penggunaan logika orde pertama dalam menggambarkan kebenaran merupakan kemampuan dasar yang sejak dahulu digunakan dalam menyandikan bahasa natural ke dalam bahasa baku. Dengan logika orde pertama, diharapkan pernyataan maupun pertanyaan dapat tersaji dengan akurat dalam bentuk konteks yang diharapkan.

Logika orde pertama dapat dimanfaatkan guna mengimplementasikan penggambaran semantik yang dapat diperluas untuk menyelesaikan berbagai permasalahan yang tidak jelas, contohnya yaitu ketidakjelasan kuantifikator dimungkinkan untuk menggunakan pendekatan Montague, Cooper Storage, Keller Storage, dan *Hole Semantics*. Guna menyelesaikan ketidakjelasan dalam konteks yang lebih luas, logika orde pertama perlu dilengkapi dengan konsep struktur yang diskursif, yang dapat dipakai dalam pemilihan semesta pembicaraan.

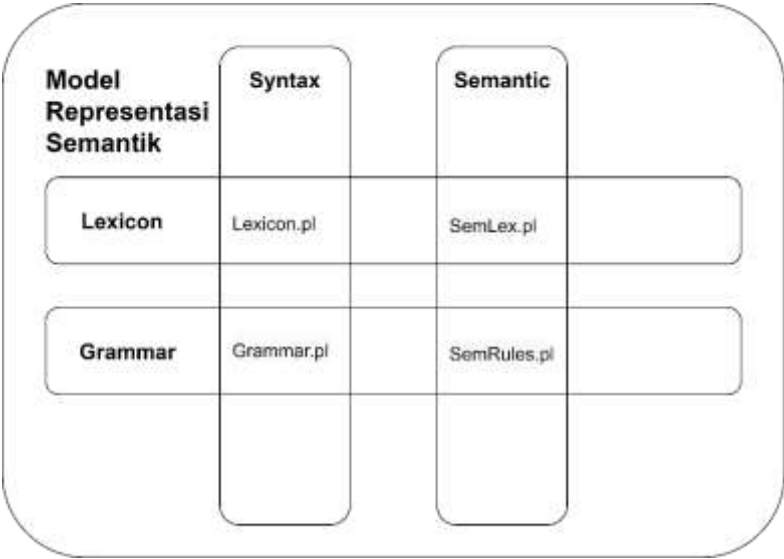
Dengan memilih semesta pembicaraan, dapat dibangun mekanisme wawancara yang mewakili kepentingan pemakai dapat dibangun. Keterbatasan mekanisme wawancara yang dibangun yaitu terbatasnya ketika membuat kesimpulan yang sangat bergantung kepada susunan kalimat hasil analisis semantik dan sintaktik. Hal tersebut dikarenakan sistem masih belum dapat menarik kesimpulan dari model dan hanya sebatas pada unifikasi formula logika orde pertama antara kalimat tanya dengan semesta pembicaraan.

7.6 Arsitektur Analisis Semantik

Desain BB1 ditujukan dalam pemenuhan tiga persyaratan ketika mengimplementasikan kaidah bahasa, yaitu: modular (setiap bagian mempunyai kedudukan secara terinci dan antarmuka yang jelas dengan bagian lain), ekstensibilitas (kaidah bahasa mesti mudah diperluas ketika terdapat kebutuhan baru), dan reusabilitas (usabilitas ulang,

dimana perangkat kaidah bahasa dapat dipergunakan meskipun terdapat pergantian bentuk penggambaran semantik).

Guna memenuhi ketiga persyaratan, maka desain yang dibuat dibagi menjadi dua, yaitu: kaidah sintaktik (Grammar.pl) yang sesuai kaidah semantik gabungan bentuk kalimat yang terhubung (SemRules.pl), dan kosakata (Lexicon.pl, yang berisi kosa-kata) yang sesuai dengan kaidah semantik leksikal (SemLex.pl).



7.7 Leksikon

Leksikon merupakan kumpulan kata yang bakal dikenali pada sistem. Pada kumpulan kata ini akan memberikan bagian dan properti dari setiap kata. Contohnya yaitu dengan kategori PN:

```
lexEntry(pn,[symbol:ibu,syntax:[ibu]]).
```

Pada umumnya, setiap masukkan pada leksikon dapat disediakan berdasarkan predikat: `lexEntry(Category,Features)`. *Category* menyediakan model struktur yang digambarkan dengan kata dalam semantik, sementara *Features* menyediakan sebuah bagian atribut nilai, yang terlibat pada analisis semantik. Misalnya, simbol 'ibu', akan dikelola menjadi rangkaian PN, dengan sintaks (tampilan) yang akan tampil pada kalimat (teks) yang akan dikenali adalah 'ibu'.

Berdasarkan struktur yang diwakili oleh masing-masing kosa kata, mungkin akan ada bagian yang berbeda pada tiap bentuk leksikon. Contoh pada determinator yang membutuhkan kuantifikator bagi nomina yang ditentukan, ataupun verba transitif yang membutuhkan karakteristik dalam mendefinisikan nilai 'numerik' berdasarkan verba, sebagai tunggal ataupun jamak.

7.8 Semantik Leksikon

Semantik leksikon memberi pengertian semantik seutuhnya (seperti sebuah '*macro*') dalam seluruh kata pada leksikon. Contohnya ketika suatu kata merupakan bagian dari PN, maka prosedur KL yang mesti dilaksanakan yaitu :

`semLex(pn,M):-`

`M = [symbol:Sym, sem:lam(P,app(P,Sym))].`

Dapat dilihat berdasarkan pengertian `SemLex` untuk PN, yakni contohnya 'ibu' dihubungkan dengan simbol dalam leksikon serta penggambaran semantik `lam(X,ibu)`

7.9 Pendekatan Montague

Pendekatan Montague atau dikenal sebagai pembangkitan kuantifikator (*quantifier rising*) adalah situasi ketika terdapat kuantifikator yang ingin 'ditonjolkan', ataupun lebih baik dari yang lain, sehingga kuantifikator tersebut harus dihubungkan pada sebuah parameter 'bayangan', yang dikenal dengan suatu '*placeholder*' dan dapat diberikan indeks. Parameter bayangan tersebut akan dimuat sesuai berdasarkan peraturan sintaktik dalam tingkat pemecahan kalimat yang tepat. Fokus utamanya yaitu "merekam" posisi *placeholder*, serta "memutar"-nya kembali ketika akan tampil pada lain kesempatan, yakni ketika akan ditelusuri penggambaran semantik nya berdasarkan penerapan KL (Montague, 2001).

DAFTAR PUSTAKA

- Dawis, A.M. dkk. 2022. *Artificial Intelligence : Konsep Dasar Dan Kajian Praktis*. Edited by Erlangga and S. Riani. TOHAR MEDIA.
- Montague, J. 2001. Dasar-Dasar Gambar Perspektif: Sebuah Pendekatan Visual, in Paulus Hanoto Adjie (ed.) *Perspective Drawing - Technique*. Third Edition. Jakarta: Erlangga, p. 208.
- Susatyono, J.D. 2021. *KECERDASAN BUATAN, Kajian Konsep dan Penerapan*, Penerbit Yayasan Prima Agus Teknik.

BAB 8

REASONING : LOGIKA FUZZY

Oleh Citra Nurina Prabiantissa

8.1 Reasoning (Penalaran)

Penalaran dalam aspek teoritis merupakan sebuah proses berpikir secara logis dan sistematis untuk membentuk dan mengevaluasi suatu keyakinan terhadap pernyataan atau asersi. Tujuan dari penalaran adalah untuk menentukan secara logis dan objektif, apakah suatu pernyataan valid (benar atau salah) sehingga pantas untuk diyakini atau dianut. Dari definisi dan tujuan, dapat dilihat bahwa penalaran digunakan untuk mengevaluasi apakah suatu pernyataan itu dapat diyakini atau dianut atau kembali secara literal dengan cara melihat alasan (reason) dibalik suatu pernyataan.

Berbeda dengan *searching* yang merepresentasikan kedalam state dan ruang masalah serta menggunakan strategi pencarian untuk menemukan solusi, *reasoning* merepresentasikan masalah kedalam basis pengetahuan dan merupakan proses penalaran untuk menemukan solusi. Pada teknik *searching*, dua masalah utama yang dihadapi adalah: kesulitan dalam menentukan apakah aturan produksi (*operator*) sudah lengkap atau belum. Jika masalah yang dihadapi cukup kompleks, maka akan terjadi ketidakefisienan representasi keadaan (*state*). Misalkan untuk masalah *8-puzzle* representasi suatu state adalah: delapan posisi angka, satu *parent*, dan maksimum 4 suksesor. Cara lain yang dapat dilakukan apabila menghadapi masalah yang lebih kompleks adalah dengan menggunakan representasi sederhana yaitu menggunakan logika (logic).

Logic merupakan suatu pengkajian ilmiah tentang serangkaian penalaran, sistem kaidah dan prosedur yang membantu proses penalaran. Untuk memproses sebuah logic dibutuhkan input berupa premis atau fakta. Hasil dari logic merupakan kesimpulan dari fakta/premis.

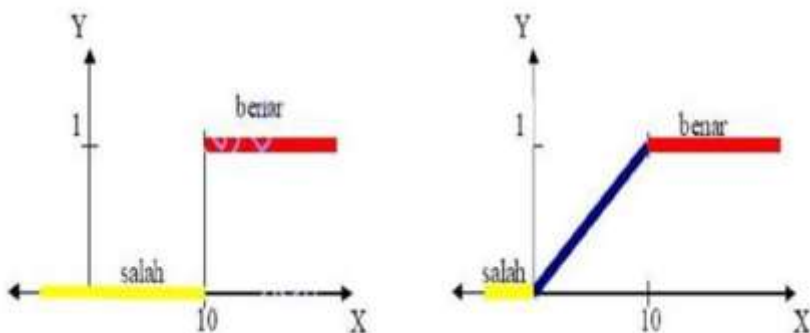


Gambar 8.1. Proses Logic

Ada lima jenis logic untuk merepresentasikan basis pengetahuan dan untuk melakukan penalaran, yaitu *First-order Logic*, *Propositional Logic*, *Probability Theory*, *Temporal Logic*, dan *Fuzzy Logic*. Pada bab ini jenis logic yang akan dibahas adalah tentang *Fuzzy Logic* (logika fuzzy).

8.2 Definisi Logika Fuzzy

Fuzzy berarti samar atau kabur. Sedangkan Logika Fuzzy merupakan peningkatan dari logika Boolean yang memperkenalkan tentang nilai kekaburan atau samar antara benar dan salah (Setia, 2019). Fuzzy memiliki derajat keanggotaan dengan rentang nilai 0 sampai 1 (pada gambar 2). Berbeda dengan sebuah himpunan tegas yang memiliki nilai 0 atau 1 yang menganalogikan sebagai jawaban ya dan tidak.



Gambar 8.2. Nilai tegas dan fuzzy

Logika fuzzy digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (linguistik). Selain itu, logika fuzzy menyediakan cara sederhana untuk menggambarkan kesimpulan pasti dari informasi yang ambigu, samar, dan tidak tepat (Amrizal and Aini, 2013).

8.3 Sejarah Logika Fuzzy

Logika fuzzy diperkenalkan pertama kali pada tahun 1965 oleh Profesor Lotfi Zadeh pada artikel yang berjudul "Fuzzy sets" (Sulandari, Winita; Suhartono and Yudhanto, 2020). Profesor Lotfi berasal dari Universitas California di Berkeley. Pada saat itu fuzzy bukan digunakan sebagai metodologi control tetapi sebagai cara pemrosesan data dengan memperkenankan penggunaan *partial set membership* dibanding *crisp set membership* atau *non-membership*.

Pada tahun 1973, Profesor Ebrahim (Abe) Mamdani dan Sedrak Assilian telah menggunakan logika Fuzzy untuk membangun sistem kontrol mesin uap secara sukses. Sejak saat itu, fuzzy semakin berkembang.

8.4 Himpunan Fuzzy

Konsep dasar dari logika fuzzy adalah himpunan fuzzy. Berikut merupakan penjelasan mengenai himpunan fuzzy:

8.4.1 Crisp Set

Pada himpunan tegas (Crisp Set), nilai keanggotaan dari himpunan ini dinyatakan dalam fungsi karakteristik, setiap objek dalam himpunan memiliki derajat keanggotaan 1 dan objek diluar himpunan memiliki derajat keanggotaan 0. Sebagai contoh, A adalah himpunan anggota yang memiliki tinggi badan. Derajat keanggotaan dalam himpunan A memiliki fungsi karakteristik,

$$\chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

Jika anggota dari himpunan A adalah orang yang memiliki tinggi minimal 170 cm (bernilai 1), maka yang bukan merupakan himpunan A (bernilai 0) adalah orang dengan tinggi badan kurang dari 170 cm.

8.4.2 Fuzzy Set (Himpunan Fuzzy)

Profesor Lutfi A. Zadeh mengembangkan suatu teori yang disebut fuzzy set (himpunan fuzzy). Fuzzy set adalah sebuah kelas dari objek dengan serangkaian kesatuan dari nilai keanggotaan. Teori ini dikembangkan untuk merepresentasikan pernyataan yang tidak pasti antara nilai benar dan salah. Logika fuzzy memberikan nilai yang spesifik pada setiap nilai dengan menentukan fungsi keanggotaan (*membership function*). Nilai keanggotaan fuzzy berada pada rentang [0-1] yang menunjukkan probabilitas kejadian yang memiliki nilai kemungkinan pada rentang [0-1].

Teori fuzzy menggunakan variabel linguistik yang nilainya menggunakan kata-kata fuzzy, atau lebih bersifat ekspresi daripada membuat angka - angka yang berakar pada bahasa alami (Davvaz, Mukhlash and Soleha, 2021). Kata-kata

fuzzy tidak akurat tetapi sangat mudah dimengerti dan digunakan secara luas dalam bahasa sehari-hari. Contoh:

Usia	Muda, Paruh Baya, Tua
Tinggi	Sangat tinggi, rata-rata tinggi, pendek

Berikut ini merupakan contoh himpunan fuzzy:

Diketahui X merupakan himpunan mahasiswa Institut Teknologi Adhi Tama Surabaya dan jika diketahui:

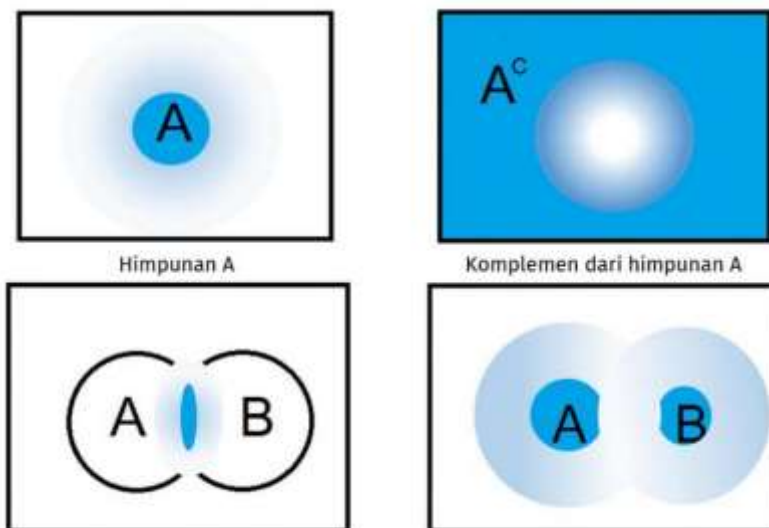
A = Himpunan semua mahasiswa Teknik Informatika;

B = Himpunan semua mahasiswa cerdas;

Dari permasalahan tersebut Himpunan A merupakan himpunan *crisp set*, sedangkan himpunan B adalah himpunan fuzzy.

8.4.4 Fuzzy Set Operation

Fuzzy set operation adalah macam – macam operasi yang dilakukan pada fuzzy set. Operasi ini merupakan generalisasi dari operasi crisp set. Terdapat tiga operasi yaitu *fuzzy unions*, *fuzzy intersection*, dan *fuzzy complements*. Pada gambar 8.3 menunjukkan ilustrasi operasi pada himpunan fuzzy dalam diagram Venn. Warna gradasi menunjukkan perbedaan derajat keanggotaan. Semakin terang semakin kecil derajat keanggotaan.



Gambar 8.3. Operasi pada himpunan fuzzy
Sumber: (Zadeh, 1965)

Himpunan fuzzy A dalam X dengan X adalah universe of discourse (semesta pembicaraan), dapat dinyatakan sebagai

$$A = \{(x, \mu_A(x)) | x \in X\}$$

atau dapat juga dinyatakan sebagai

$$A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \right\}$$

Keterangan:

μ_A = fungsi keanggotaan dari himpunan fuzzy A

Fungsi keanggotaan komplemen dari A adalah $\mu_A^c = 1 - \mu_A$.

Misalkan A dan B adalah 2 himpunan fuzzy dengan masing-masing fungsi keanggotaan $\mu_A(x)$ dan $\mu_B(x)$. Fungsi keanggotaan $\mu_C(x)$ dari himpunan $C = A \cap B$ didefinisikan sebagai

$$\mu_C(x) = \min\{\mu_A(x), \mu_B(x)\}, x \in X$$

Dan fungsi keanggotaan $\mu_C(x)$ dari himpunan $D = A \cup B$ didefinisikan sebagai :

$$\mu_D(x) = \max\{\min\{\mu_A(x), \mu_B(x)\}, x \in X$$

Contoh:

Apabila diketahui

$$A = \left\{ \frac{0}{1} + \frac{0,1}{2} + \frac{0,4}{3} + \frac{0,5}{4} + \frac{0,6}{5} + \frac{0,9}{6} + \frac{1}{7} + \frac{1}{8} \right\}$$

$$B = \left\{ \frac{0,5}{2} + \frac{0,4}{3} + \frac{0,3}{4} + \frac{0,5}{5} + \frac{0,8}{6} + \frac{1}{7} \right\}$$

Tentukan komplemen, irisan, dan union himpunan fuzzy di atas!

Komplemen

$$A^c = \left\{ \frac{1}{1} + \frac{0,9}{2} + \frac{0,6}{3} + \frac{0,5}{4} + \frac{0,4}{5} + \frac{0,1}{6} + \frac{0}{7} + \frac{0}{8} \right\}$$

$$B^c = \left\{ \frac{0,5}{2} + \frac{0,6}{3} + \frac{0,7}{4} + \frac{0,5}{5} + \frac{0,2}{6} + \frac{0}{7} \right\}$$

Irisan

$$A \cap B = \left\{ \frac{0,1}{2} + \frac{0,4}{3} + \frac{0,3}{4} + \frac{0,5}{5} + \frac{0,8}{6} + \frac{1}{7} \right\}$$

Union

$$A \cup B = \left\{ \frac{0}{1} + \frac{0,5}{2} + \frac{0,4}{3} + \frac{0,5}{4} + \frac{0,6}{5} + \frac{0,9}{6} + \frac{1}{7} + \frac{1}{8} \right\}$$

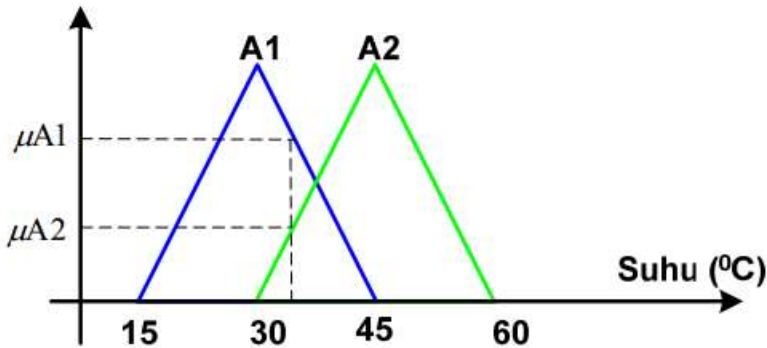
8.5 Tahapan Proses Logika Fuzzy

Dalam proses logika fuzzy terdapat 3 tahapan yang meliputi:

1. Fuzzifikasi
2. *Fuzzy Inferensi System (FIS)*
3. Defuzzifikasi

8.5.1 Fuzzifikasi

Fuzzifikasi merupakan proses perubahan data ke dalam bentuk variabel fuzzy menggunakan fungsi keanggotaan (Khairina, 2017). Contoh pada permasalahan suhu berikut ini merupakan fuzzifikasi dari suhu 35°C:



Gambar 8.4. Fuzzifikasi Suhu 35 °C

Sesuai dengan permasalahan pada gambar 8.4 contoh perhitungan fuzzifikasi dapat ditunjukkan sebagai berikut:

$$\mu_{A1} = \frac{c - x}{c - b} = \frac{45 - 35}{45 - 30} = \frac{2}{3}$$
$$\mu_{A2} = \frac{x - a}{b - a} = \frac{35 - 30}{45 - 30} = \frac{1}{3}$$

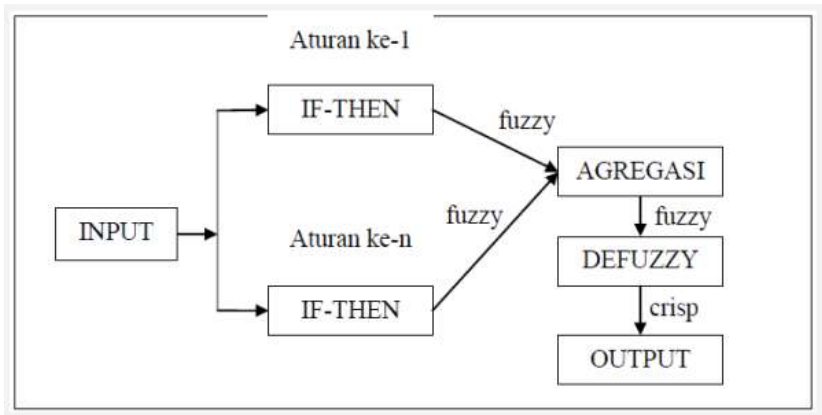
8.5.2 Fuzzy Inferensi System (FIS)

Fuzzy inference engine adalah metode fuzzy yang digunakan untuk mengambil suatu keputusan. Berikut ini merupakan beberapa metode fuzzy yang sering digunakan:

a. Metode Tsukamoto

Metode ini pertama kali diperkenalkan oleh Tsukamoto. Metode ini memiliki konsep dimana setiap aturan yang berkaitan dengan IF-THEN harus dipresentasikan suatu

himpunan logika fuzzy dengan keanggotaan yang monoton (Azmi *et al.*, 2019). Proses inferensi metode tsukamoto ditunjukkan pada skema berikut ini:



Gambar 8.5. Proses Inferensi Metode Tsukamoto

Tahapan proses inferensi metode Tsukamoto:

1. Sistem Inferensi Fuzzy menerima input crisp. Kemudian input ini dikirim ke basis pengetahuan yang berisi n aturan fuzzy dalam bentuk IF-THEN.
2. Metode Tsukamoto menerapkan penalaran monoton pada setiap aturan. Penalaran monoton hanya menggunakan satu aturan, sedangkan metode Tsukamoto memiliki banyak aturan. Oleh karena itu pada metode ini setiap aturan yang berbentuk IF-THEN harus direpresentasikan dengan himpunan fuzzy dengan fungsi keanggotaan yang monoton.
3. Output hasil inferensi pada tiap aturan diberikan secara tegas (crisp) menggunakan defuzzifikasi dengan konsep rata-rata terbobot.

b. Metode Mamdani

Pada tahun 1975, Metode ini diperkenalkan pertama kali oleh Ebrahim Mamdani. Metode ini memiliki 4 tahapan untuk memperoleh output:

1. Pembentukan himpunan fuzzy

Pada metode ini, input atau output dibagi menjadi satu atau lebih himpunan fuzzy. Himpunan fuzzy diambil dari fungsi keanggotaan dinyatakan sebagai fungsi matematis tertentu.

2. Pengaplikasian Fungsi Implikasi (aturan)

Fungsi implikasi yang digunakan metode mamdani adalah Min dan dalam melakukan komposisi dengan menggunakan Max. Metode komposisi ini sering disebut Max-Min.

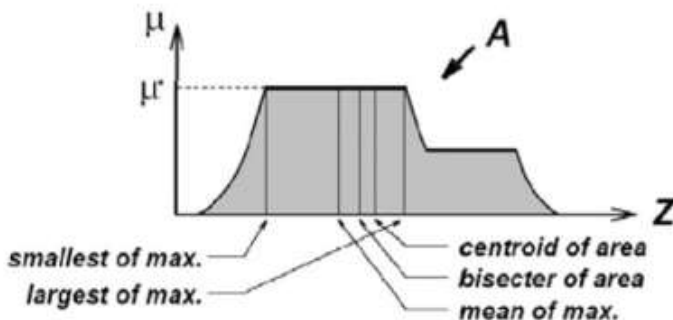
3. Komposisi Aturan

Langkah pertama pengambilan keputusan metode Mamdani adalah melakukan proses fuzifikasi untuk memetakan data tegas masukan kesalahan dan beda kesalahan data fuzzy sesuai dengan tipe dan bentuk fungsi keanggotaan (Setiani Asih, 2018). Langkah kedua adalah melakukan proses terhadap kedua data fuzzy tersebut dengan operator AND yang akan mengambil nilai paling minimal dari dua data tersebut. Langkah ketiga dengan impilasi MIN akan memotong fungsi keanggotaan keluaran setelah melalui operator AND sehingga didapat daerah fuzzy. Ketiga proses tersebut juga diterapkan pada aturan-aturan fuzzy berikutnya. Setelah aturan fuzzy dieksekusi, dilakukan proses komposisi dengan metode MAX yaitu solusi himpunan fuzzy diperoleh dengan cara mengambil nilai maksimum aturan, kemudian menggunakannya untuk memodifikasi daerahfuzzy, dan mengaplikasikan ke output dengan menggunakan operator OR (union). Jika proposisi telah

dievaluasi, maka output akan berisi suatu himpunan fuzzy yang merefleksikan kontribusi dari tiap-tiap proposisi. Setelah proses implikasi dan komposisi telah dilakukan maka proses selanjutnya adalah proses defuzzifikasi.

4. Penegasan (defuzzifikasi)

Input dari proses defuzzyfikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Jika diberikan suatu himpunan fuzzy dalam range tertentu, maka dapat diambil suatu nilai crisp tertentu sebagai output[7]. Defuzifikasi pada komposisi aturan Mamdani. Diantaranya yaitu metode COA, bisektor, Mean of Max, Largest of Max, dan Smallest of Max. Berikut ini gambaran metode defuzzifikasi pada metode Mamdani:



Gambar 8.6. Metode Defuzzifikasi pada Metode Mamdani
Sumber: (Setiani Asih, 2018)

c. Metode Sugeno

Penalaran dengan metode Sugeno hampir sama dengan metode Mamdani. Perbedaan terletak pada output sistem tidak berupa himpunan fuzzy tetapi berbentuk konstanta

atau persamaan linear (Rizky *et al.*, 2020). Metode ini diperkenalkan oleh Takagi-Sugeno kang pada tahun 1985. Ada 2 metode Sugeno yaitu sebagai berikut:

a. Fuzzy Sugeno Orde-Nol

Secara umum bentuk model Fuzzy Sugeno Orde Nol yaitu:

$IF(X_1 \text{ is } A_1) \bullet (X_2 \text{ is } A_2) \bullet (X_3 \text{ is } A_3) \bullet \dots \bullet (X_n \text{ is } A_n) \text{ THEN } z=k$

Dimana:

A_i = himpunan fuzzy ke-i sebagai anteseden.

k = suatu konstanta tegas sebagai konsekuen.

b. Fuzzy Sugeno Orde-Satu

$IF(X_1 \text{ is } A_1) \bullet \dots \bullet (X_n \text{ is } A_n) \text{ THEN } z=p_1 \cdot x_1 + \dots + p_n \cdot x_n + q$

Dimana:

A_i = himpunan fuzzy ke-i sebagai anteseden

p_i = suatu konstanta (tegas) ke-i

q = konstanta dalam konsekuen

8.5.3 Defuzzifikasi

Defuzzifikasi adalah suatu mekanisme untuk mengkonversi atau mengubah hasil output fuzzy menjadi suatu output nonfuzzy yang tegas. Output dalam bentuk fuzzy belum dapat hasil yang dapat langsung digunakan.

8.6 Penerapan Logika Fuzzy

Contoh penerapan logika Fuzzy berikut ini menggunakan metode Tsukamoto:

Diketahui sebuah pabrik mempunyai permintaan barang terbesar sebanyak 5000 barang/hari. Pada suatu hari pabrik tersebut hanya mempunyai 1000 barang/hari. Persediaan barang mencapai titik tertinggi dengan jumlah 600 barang/hari dan titik terendah 100 barang/hari. Dengan semua kondisi tersebut, pabrik tersebut dapat memproduksi barang

maksimum 7000 barang/hari dan minimal 2000 barang/hari.
Aturan fuzzy untuk proses produksi pabrik tersebut yaitu:

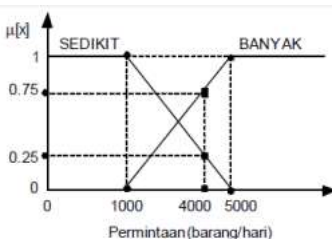
[A1] IF Permintaan BANYAK And Persediaan BANYAK
THEN Produksi Barang BERTAMBAH ;
[A2] IF permintaan SEDIKIT And persediaan SEDIKIT
THEN Produksi Barang BERKURANG ;
[A3] IF Permintaan SEDIKIT And Persediaan BANYAK
THEN Produksi Barang BERKURANG ;
[A4] IF permintaan BANYAK And persediaan SEDIKIT
THEN Produksi Barang BERTAMBAH ;

Soal: Berapa barang yang harus diproduksi jika jumlah permintaan barang sebanyak 4000 barang dan persediaan gudang masih ada sekitar 300 barang?

Jawaban:

Proses Fuzzyfikasi

1. Permintaan, terdiri dari 2 himpunan fuzzy, yaitu Banyak dan Sedikit.

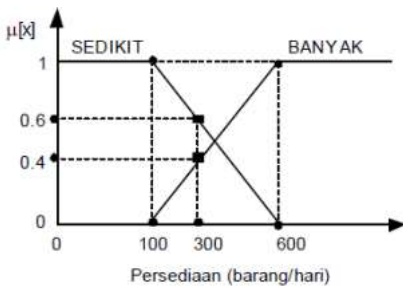


Nilai Keanggotaan untuk Permintaan:

$$\mu_{Sedikit}[4000] = \frac{5000 - 4000}{5000 - 1000} = 0.25$$

$$\mu_{Banyak}[4000] = \frac{4000 - 1000}{5000 - 1000} = 0.75$$

2. Persediaan, terdiri dari 2 himpunan fuzzy, Banyak dan Sedikit.

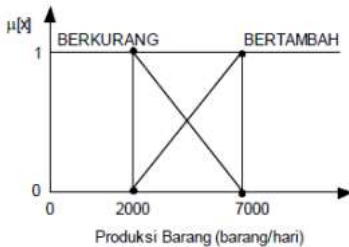


Nilai Keanggotaan untuk Permintaan:

$$\mu_{\text{Sedikit}}[300] = \frac{600 - 300}{600 - 100} = 0.6$$

$$\mu_{\text{Banyak}}[300] = \frac{300 - 100}{600 - 100} = 0.4$$

3. Produksi Barang, terdiri dari 2 himpunan fuzzy, Berkurang dan Bertambah.



$$\mu_{\text{Berkurang}}[z] = \begin{cases} 1, & z \leq 2000, \\ \frac{7000-z}{7000-2000}, & 2000 < z < 7000, \\ 0, & z \geq 7000, \end{cases}$$

$$\mu_{\text{Bertambah}}[z] = \begin{cases} 0, & z \leq 2000, \\ \frac{z-2000}{7000-2000}, & 2000 < z < 7000, \\ 1, & z \geq 7000, \end{cases}$$

Proses Inferensi

Evaluasi rule, operator Zadeh digunakan sehingga didapat α

(alpha) predikat untuk menentukan z:

Aturan-1

R1 : IF permintaan turun AND persediaan banyak THEN produksi barang berkurang

$$\begin{aligned}\alpha_{\text{predikat1}} &= \mu_{\text{permintaanTURUN}} \cap \mu_{\text{permintaanBANYAK}} \\ &= \min (\mu_{\text{permintaan TURUN}}[4000] \cap \mu_{\text{permintaan BANYAK}}[300]) \\ &= \min(0.25;0.4) \\ &= 0.25\end{aligned}$$

Untuk α predikat1 = 0.25 pada himpunan produksi BERKURANG

$$\begin{aligned}&= (7000-z)/(7000-2000) \\ &= 0.25 \\ Z1 &= 5750\end{aligned}$$

Aturan-2

R2 : IF permintaan turun AND persediaan sedikit THEN produksi barang berkurang

$$\begin{aligned}\alpha_{\text{predikat2}} &= \mu_{\text{permintaanTURUN}} \cap \mu_{\text{permintaanSEDIKIT}} \\ &= \min (\mu_{\text{permintaan TURUN}} [4000] \cap \mu_{\text{permintaan SEDIKIT}} [300]) \\ &= \min(0.25;0.6) \\ &= 0.25\end{aligned}$$

Untuk α predikat2 = 0.25 pada himpunan produksi BERKURANG

$$\begin{aligned}&= (7000-z)/(7000-2000) \\ &= 0.25 \\ Z2 &= 5750\end{aligned}$$

Aturan-3

R3 : IF permintaan naik AND persediaan banyak THEN produksi barang bertambah

$$\begin{aligned}\alpha_{\text{predikat3}} &= \mu_{\text{permintaanNAIK}} \cap \mu_{\text{permintaanBANYAK}} \\ &= \min (\mu_{\text{permintaan NAIK [4000]}} \cap \mu_{\text{persediaan BANYAK [300]}}) \\ &= \min(0.75;0.4) \\ &= 0.4\end{aligned}$$

Untuk $\alpha_{\text{predikat3}} = 0.4$ pada himpunan produksi BERTAMBAH

$$\begin{aligned}&= (z-2000)/(7000-2000) \\ &= 0.4 \\ Z_3 &= 4000\end{aligned}$$

R4 : IF permintaan naik AND persediaan sedikit THEN produksi barang bertambah

$$\begin{aligned}\alpha_{\text{predikat4}} &= \mu_{\text{permintaanNAIK}} \cap \mu_{\text{persediaanSEDIKIT}} \\ &= \min (\mu_{\text{permintaan NAIK [4000]}} \cap \mu_{\text{persediaan SEDIKIT[300]}}) \\ &= \min(0.75;0.6) \\ &= 0.6\end{aligned}$$

Untuk $\alpha_{\text{predikat4}} = 0.6$ pada himpunan produksi BERTAMBAH

$$\begin{aligned}&= (z-2000)/(7000-2000) \\ &= 0.6 \\ Z_3 &= 5000\end{aligned}$$

Proses Defuzzifikasi

$$\begin{aligned}Z &= \frac{\alpha_{\text{pred}_1} * Z_1 + \alpha_{\text{pred}_2} * Z_2 + \alpha_{\text{pred}_3} * Z_3 + \alpha_{\text{pred}_4} * Z_4}{\alpha_{\text{pred}_1} + \alpha_{\text{pred}_2} + \alpha_{\text{pred}_3} + \alpha_{\text{pred}_4}} \\ Z &= \frac{0.4 * 4000 + 0.25 * 5750 + 0.25 * 5750 + 0.6 * 5000}{0.4 + 0.25 + 0.25 + 0.6}\end{aligned}$$

$$Z = 4983$$

DAFTAR PUSTAKA

- Amrizal, V. and Aini, Q. 2013. *Kecerdasan Buatan, Kecerdasan Buatan*.
- Azmi, F. *et al.* 2019. 'Design of Water Level Detection Using Ultrasonic Sensor Based On Fuzzy Logic', *Journal of Information Technology Education: Research*, 3(1), pp. 142–149. doi: 10.31289/JITE.V3I1.2668.
- Davvaz, B., Mukhlash, I. and Soleha, S. 2021. 'Himpunan Fuzzy dan Rough Sets', *Limits: Journal of Mathematics and Its Applications*, 18(1), p. 79. doi: 10.12962/limits.v18i1.7705.
- Khairina, N. 2017. 'Analisis Fungsi Keanggotaan Fuzzy Tsukamoto Dalam Menentukan Status Kesehatan Tubuh Seseorang', *Sinkron*, 1(1), p. 19. doi: 10.33395/sinkron.v1i1.5.
- Rizky, R. *et al.* 2020. 'Penerapa Metode Fuzzy Sugeno Untuk pengukuran Keakuratan Jarak Pada Pintu Otomatis di CV Bejo Perkasa', *Jurnal Teknik Informatika UNIKA Santo Thomas*, 05, pp. 33–42. Available at: <http://ejournal.ust.ac.id/index.php/JTIUST/article/view/679>.
- Setia, B. 2019. 'Penerapan Logika Fuzzy pada Sistem Cerdas', *Jurnal Sistem Cerdas*, 2(1), pp. 61–66. doi: 10.37396/jsc.v2i1.18.
- Setiani Asih, M. 2018. 'Sistem Pendukung Keputusan Fuzzy Mamdani pada Alat Penyiraman Tanaman Otomatis', *Jurnal Sistem Informasi*, 5341(April), p. 1.
- Sulandari, Winita; Suhartono, S. and Yudhanto, Y. 2020. *Aplikasi Fuzzy Pada Pemodelan Runtun Waktu*. Vol. 1. Khazanah Intelektual.
- Zadeh, L. A. 1965. 'Fuzzy Set', *Information and Control*, 8(3), pp. 338–353. doi: 10.1016/s0019-9958(65)90241-x.

BAB 9

REPRESENTASI PENGETAHUAN

Oleh Melissa Indah Fianty

9.1 Definisi Representasi Pengetahuan

Apa itu representasi pengetahuan? gagasan tersebut dapat dipahami dengan baik dalam kaitannya dengan lima peran berbeda yang dimainkannya, masing-masing penting untuk tugas yang ada:

Pertama, representasi pengetahuan pada dasarnya adalah pengganti, pengganti benda itu sendiri, yang digunakan untuk memungkinkan entitas menentukan konsekuensi dengan berpikir daripada bertindak, yaitu dengan penalaran tentang dunia daripada mengambil tindakan di dalamnya.

Kedua, itu adalah seperangkat komitmen ontologis, yaitu jawaban atas pertanyaan, Dalam hal apa saya harus berpikir tentang dunia?

Ketiga, itu adalah teori fragmentaris penalaran cerdas yang dinyatakan dalam tiga komponen: (1) konsepsi dasar representasi penalaran cerdas, (2) seperangkat kesimpulan bahwa representasi sanksi, dan (3) serangkaian kesimpulan yang direkomendasikan.

Keempat, ini adalah media untuk komputasi yang efisien secara pragmatis, yaitu lingkungan komputasi di mana pemikiran dicapai. Salah satu kontribusi untuk efisiensi pragmatis ini disediakan oleh panduan yang diberikan representasi untuk mengatur informasi guna memfasilitasi pembuatan kesimpulan yang direkomendasikan (Brachman and Levesque, 2004).

Kelima, itu adalah media ekspresi manusia, yaitu bahasa di mana kita mengatakan hal-hal tentang dunia.

Terdapat lima peran mendasar representasi pengetahuan, yaitu: (Bench-Capon, 2014)

1. *A Knowledge Representation Is a Surrogate*

Representasi Pengetahuan Adalah Pengganti Entitas cerdas mana pun yang ingin bernalar tentang dunianya menghadapi fakta penting yang tak terhindarkan: Penalaran adalah proses yang berlangsung secara internal, tetapi kebanyakan hal yang ada secara eksternal. Program (atau orang) yang terlibat dalam perencanaan perakitan sepeda, misalnya, mungkin harus memikirkan entitas seperti roda, rantai, sproket, dan setang, tetapi hal-hal seperti itu hanya ada di dunia luar.

Dasar pemikiran dan peran representasi: Ini berfungsi sebagai pengganti di dalam penalar, pengganti hal-hal yang ada di dunia. Operasi pada dan dengan representasi menggantikan operasi pada benda nyata, yaitu, menggantikan interaksi langsung dengan dunia. Dalam pandangan ini, penalaran itu sendiri, sebagian, merupakan pengganti tindakan di dunia ketika kita tidak dapat atau (belum) ingin melakukan tindakan itu.

2. *A Knowledge Representation Is a Set of Ontological Commitments*

Semua representasi adalah perkiraan yang tidak sempurna terhadap realitas, setiap perkiraan memperhatikan beberapa hal dan mengabaikan yang lain, maka dalam memilih representasi apa pun, Karena itu, dalam tindakan yang sama mau tidak mau membuat serangkaian keputusan tentang bagaimana dan apa yang harus dilihat di dunia. Artinya, memilih representasi berarti membuat satu set komitmen ontologis. Komitmen tersebut, pada dasarnya, adalah sepasang kaca yang kuat yang menentukan apa yang

dapat kita lihat, membawa beberapa bagian dunia ke dalam fokus yang tajam dengan mengorbankan kekaburan. bagian lain.

Komitmen ini dan efek pengaburan fokusnya bukanlah efek samping insidental dari pilihan representasi; mereka adalah esensi: Representasi pengetahuan adalah seperangkat komitmen ontologis. Hal ini tidak dapat dihindari karena ketidaksempurnaan representasi yang tak terelakkan. Hal ini berguna karena pemilihan komitmen yang bijaksana memberikan kesempatan untuk memusatkan perhatian pada aspek-aspek dunia yang kami yakini relevan.

Efek pemfokusan adalah bagian penting dari apa yang ditawarkan representasi karena kompleksitas dunia alami luar biasa. Kami (dan mesin penalaran kami) membutuhkan panduan dalam memutuskan apa yang harus diperhatikan dan apa yang harus diabaikan. Kacamata yang disediakan oleh representasi dapat memberikan panduan ini: Dalam memberi tahu kita apa dan bagaimana melihat, mereka memungkinkan kita untuk mengatasi kompleksitas dan detail yang tidak dapat dipertahankan. Oleh karena itu, komitmen ontologis yang dibuat oleh representasi dapat menjadi salah satu kontribusi terpentingnya.

3. *A Knowledge Representation Is a Fragmentary Theory of Intelligent Reasoning*

Peran ketiga untuk representasi adalah sebagai teori penalaran cerdas yang terpisah-pisah. Peran ini muncul karena konsepsi awal representasi biasanya dimotivasi oleh beberapa wawasan yang menunjukkan bagaimana orang bernalar secara cerdas atau oleh beberapa keyakinan tentang apa artinya bernalar secara cerdas.

Teori ini terbagi dalam dua pengertian yang berbeda: (1) representasi biasanya hanya menggabungkan sebagian dari wawasan atau kepercayaan yang memotivasinya dan (2)

wawasan atau kepercayaan ini, pada gilirannya, hanyalah bagian dari yang kompleks dan multidimensi. fenomena segi penalaran cerdas.

Teori representasi tentang penalaran cerdas seringkali tersirat tetapi dapat dibuat lebih jelas dengan memeriksa tiga komponennya: (1) konsepsi dasar representasi tentang inferensi cerdas, (2) kumpulan kesimpulan yang disucikan oleh representasi dan (3) kesimpulan yang merekomendasikan.

Jika kesimpulan yang disetujui menunjukkan apa yang dapat disimpulkan sama sekali, kesimpulan yang direkomendasikan berkaitan dengan apa yang harus disimpulkan. (Bimbingan diperlukan karena kumpulan kesimpulan yang disetujui biasanya terlalu besar untuk digunakan tanpa pandang bulu.)

Komponen-komponen ini juga dapat dilihat sebagai jawaban representasi atas tiga pertanyaan mendasar yang sesuai: (1) Apa artinya bernalar secara cerdas? (2) Apa yang dapat kita simpulkan dari apa yang kita ketahui? dan (3) Apa yang harus kita simpulkan dari apa yang kita ketahui? Jawaban atas pertanyaan-pertanyaan ini adalah inti dari semangat dan pola pikir representasi; mengetahui posisinya dalam masalah ini memberi tahu kita banyak hal tentangnya.

Kita mulai dengan yang pertama dari komponen ini, memeriksa dua dari beberapa konsep penalaran cerdas yang berbeda secara mendasar yang telah dieksplorasi dalam AI. Konsepsi-konsepsi ini dan asumsi-asumsi yang mendasarinya menunjukkan berbagai pandangan tentang pertanyaan dan menetapkan konteks penting untuk komponen-komponen yang tersisa.

4. *A Knowledge Representation Is a Medium for Efficient Computation*

Dari pandangan mekanistik murni, penalaran pada mesin (dan, mungkin, pada manusia) adalah proses komputasi. Sederhananya, untuk menggunakan representasi, kita harus menghitungnya. Akibatnya, pertanyaan tentang efisiensi komputasi menjadi pusat gagasan representasi.

Fakta ini telah lama diakui, setidaknya secara implisit, oleh desainer representasi: Pendekatan bersama dapat menawarkan percepatan yang signifikan dan tingkat efisiensi komputasi yang berguna secara pragmatis.

Masing-masing pendekatan ini memiliki kelebihan dan kekurangan. Mengekspresikan strategi penalaran dalam first-order logic sesuai dengan semangat pendekatan ahli logika, yaitu representasi eksplisit pengetahuan dalam representasi deklaratif yang seragam. Tapi pendekatan ini sering bermasalah dalam praktiknya: bahasa yang dirancang untuk mengungkapkan fakta secara deklaratif belum tentu baik untuk mengungkapkan karakteristik informasi imperatif dari strategi penalaran.

5. *A Knowledge Representation Is a Medium of Human Expression*

Representasi pengetahuan juga merupakan cara kita mengungkapkan hal-hal tentang dunia, media ekspresi dan komunikasi di mana kita memberi tahu mesin (dan mungkin satu sama lain) tentang dunia. Peran representasi ini tidak dapat dihindari selama kita perlu memberi tahu mesin (atau orang lain) tentang dunia dan selama kita melakukannya dengan membuat dan mengkomunikasikan representasi. Dengan demikian, peran kelima representasi pengetahuan adalah sebagai media ekspresi dan komunikasi untuk kita gunakan.

9.2 Teknik Representasi Pengetahuan

Representasi pengetahuan adalah suatu teknik untuk merepresentasikan basis pengetahuan yang diperoleh ke dalam suatu skema/ diagram tertentu sehingga dapat diketahui relasi/keterhubungan antara suatu data dengan data yang lain. Teknik ini membantu *knowledge engineer* dalam memahami struktur pengetahuan yang akan dibuat sistem pakarnya.

Representasi ditujukan untuk menemukan sifat-sifat penting problema dan membuat informasi tersebut dapat diakses oleh prosedur pemecahan problematika. Bahasa representasi harus mampu membuat seseorang ahli bahasa program mampu mengekspresikan pengetahuan yang dibutuhkan untuk mendapatkan penyelesaian problematika.

Dua bagian dasar sistem kecerdasan buatan:

1. Basis pengetahuan:
Berisi fakta tentang objek-objek dalam domain yang dipilih dan hubungan diantara domain-domain tersebut
2. Inference Engine:
Merupakan sekumpulan prosedur yang digunakan untuk menguji basis pengetahuan dalam menjawab suatu pertanyaan, menyelesaikan masalah, atau membuat keputusan

Basis pengetahuan berisi struktur data yang dapat dimanipulasi oleh suatu sistem inferensi yang menggunakan pencarian dan teknik pencocokan pola pada basis pengetahuan yang bermanfaat untuk menjawab pertanyaan, menggambarkan kesimpulan atau bentuk lainnya sebagai suatu fungsi kecerdasan

Pengetahuan adalah penjelasan mengenai ranah ilmu yang mampu dimanfaatkan untuk menyelesaikan problematika dalam ranah ilmu tersebut. Untuk menyelesaikan banyak

problematika, diperlukan banyak keahlian. Keahlian tersebut harus diterapkan ke dalam komputer. Sebagai bagian perancangan bahasa pemrograman dalam merampungkan sebuah problem, definisi pengetahuan harus diterapkan dengan jelas.

Representasi ditujukan untuk menemukan sifat-sifat penting problema dan membuat informasi tersebut dapat diakses oleh prosedur pemecahan problematika. Bahasa representasi harus mampu membuat seseorang ahli bahasa program mampu mengekspresikan pengetahuan yang dibutuhkan untuk mendapatkan penyelesaian problematika. Knowledge (pengetahuan) adalah penjelasan mengenai ranah ilmu yang mampu dimanfaatkan untuk menyelesaikan problematika dalam Representasi pengetahuan adalah cara untuk menyajikan pengetahuan yang diperoleh ke dalam suatu skema/diagram tertentu sehingga dapat diketahui relasi antara suatu pengetahuan dengan pengetahuan yang lain dan dapat dipakai untuk menguji kebenaran penalarannya. Secara teknik kita akan membahas representasi pengetahuan menjadi lima kelompok:

1. Representasi Logika
2. List
3. Tree
4. Jaringan Semantik
5. Frame
6. Script (Naskah)
7. Aturan Produksi (Kaidah Produksi)

Dalam representasi pengetahuan ada dua hal yang harus diperhatikan yaitu:

1. Fakta, dapat berupa keterangan/kalimat dalam bahasa alami, logika atau benda.

2. Formula, bentuk representasi yang dapat dimanipulasi dalam mekanisme pemecahan soal. Bentuk ini harus melukiskan hubungan antara komponen-komponen domain problem.

Dalam representasi sebuah fakta yang kita gunakan dalam sebuah program, kita juga harus konsisten dengan representasi yang menggunakan bahasa natural.

Representasi yang baik, harus:

1. Mengemukakan hal secara eksplisit
2. Membuat masalah menjadi transparan
3. Kompit dan efisien
4. Menampilkan batasan-batasan yang diperlukan
5. Dapat dilakukan komputasi (ada batasan/konstraint)

9.2.1 Klasifikasi Representasi Pengetahuan

Secara singkat, representasi pengetahuan diklasifikasikan menjadi 4 kategori:

1. Representasi logika: representasi jenis ini menggunakan ekspresi-ekspresi dalam logika formal untuk merepresentasikan basis pengetahuan.
2. Representasi prosedural: representasi menggambarkan pengetahuan kumpulan instruksi untuk memecahkan suatu problem.
3. Representasi network: representasi ini menangkap pengetahuan sebagai graf dimana simpul-simpulnya menggambarkan obyek atau konsep dari problema yang dihadapi, sedangkan edgenya menggambarkan hubungan atau asosiasi antar merek.
4. Representasi terstruktur: representasi terstruktur memperluas network dengan cara membuat setiap simpulnya menjadi sebuah struktur data kompleks.

9.2.2 Syarat Representasi Pengetahuan

Pengetahuan harus direpresentasikan dengan baik supaya kesimpulan yang diambil tepat. Sebuah representasi yang baik memiliki syarat seperti berikut:

1. Cukup kaya untuk mengekspresikan pengetahuan yang dibutuhkan untuk menyelesaikan permasalahan.
2. Sedekat mungkin dengan permasalahannya. Representasi harus padat, akurat dan dapat dipertahankan. Representasi juga harus memudahkan melihat hubungan antarrepresentasi dengan domain yang direpresentasikan. Alasannya, supaya mudah dalam menentukan ketepatan representasi pengetahuan
3. Dapat digunakan untuk sebuah komputasi yang efisien. Artinya representasi harus mampu untuk mengekspresikan fitur dari masalahnya
4. Dapat diperoleh dari orang, data dan pengalaman di masa lalu.

9.2.3 Keuntungan Representasi Pengetahuan

Dengan representasi, banyak hal yang akan kita dapatkan dalam menyelesaikan suatu masalah. Di bawah ini adalah beberapa keuntungan yang akan kita dapatkan ketika kita membuat representasi pengetahuan, yaitu: (Crawford, Gingerich and Eliasmith, 2016)

1. Dengan representasi yang baik, membuat objek dan relasi yang penting menjadi jelas.
2. Representasi menyingkap batasan dalam suatu permasalahan. Kita dapat mengungkapkan pengaruh sebuah objek atau relasi terhadap objek atau relasi lain.
3. Dengan representasi kita akan mendapatkan objek dan relasi secara bersama-sama. Kita akan dapat melihat semua yang kita inginkan dalam satu waktu.

4. Kita dapat menghilangkan semua komponen yang tidak berhubungan dengan permasalahan yang sedang kita selesaikan. Atau kita dapat menyembunyikan beberapa informasi yang tidak kita butuhkan untuk sementara dan pada saat kita membutuhkannya kita dapat menampilkan kembali.
5. Dengan representasi akan membuat permasalahan yang sedang kita selesaikan menjadi transparan. Kita akan memahami permasalahan yang kita selesaikan.
6. Dengan representasi kita dapat menyingkap suatu permasalahan secara lengkap, sehingga permasalahan dapat diselesaikan.
7. Dengan representasi akan membuat permasalahan menjadi ringkas. Kita akan berpikir ringkas (merepresentasikan apa yang ingin kita representasikan secara efisien)
8. Dengan representasi, maka akan menjadikan pekerjaan kita menjadi cepat.
9. Dengan representasi, menjadikan permasalahan yang kita selesaikan dapat terkomputerisasi. Dengan representasi ini kita akan dapat melakukan prosedur-prosedur dalam menyelesaikan suatu permasalahan.

Selain keuntungan-keuntungan diatas satu hal yang menjadi prinsip dalam representasi pengetahuan adalah jika suatu permasalahan dideskripsikan dengan menggunakan representasi yang tepat, maka dapat dipastikan bahwa permasalahan tersebut dapat diselesaikan.

9.3 Representasi Logika

Representasi pengetahuan terdiri dari logika yang presentasikan array representasi logika, jaringan berupa semantic, bingkai atau frame, naskah atau script serta terakhir produksi yang mempunyai aturan. Represenrasi pengetahuan

yang paling tu adalah logika. Fakta yang sudah ada bisa digunakan untuk menarik kesimpulan logika. Premis pada logika digunakan sebagai input berbeda dengan kesimpulan yang digunakan sebagai output. Ilmu penalaran disebut juga logika, dimana berhubungan dengan kalimat dan aturan tertentu dan bisa juga berbentuk sintaks.

Logika terdiri sintaks (simbol), semantk (fakta), dan keputusan (inferensi). Logika Preposisi terdiri dari and, or, not, implikasi dan ekuivalensi. Kalimat deklaratif terdiri dari kalimat bernilai benar (1) atau salah (0) atau true false saja tetapi bukan juga keduanya. Nilai kebenaran berupa salah atau benar. Proposisi dilambangkan dengan p, q, r, x, y, a, b dan lainnya disebut proposisi atomic (Gabbay et al., 2013).

Ada 2 penalaran yang dapat dilakukan untuk mendapat konklusi:

1. Penalaran deduktif: dimulai dari prinsip umum untuk mendapatkan konklusi yang lebih khusus.

Contoh:

Premis mayor : Jika hujan turun saya tidak akan berangkat kuliah

Premis minor : Hari ini hujan turun

Konsklusi : Hari ini saya tidak akan berangkat kuliah

2. Penalaran induktif: dimulai dari fakta-fakta khusus untuk mendapatkan kesimpulan umum.

Contoh:

Premis -1 : Aljabar adalah pelajaran yang sulit

Premis -2 : Geometri adalah pelajaran yang sulit

Premis -3 : Kalkulus adalah pelajaran yang sulit

Konsklusi : Matematika adalah pelajaran yang sulit

Munculnya premis baru bisa mengakibatkan gugurnya konklusi yang sudah diperoleh, misal:

Premis -4: Kinematika adalah pelajaran yang sulit.

Premis tersebut menyebabkan konklusi: “Matematika adalah pelajaran yang sulit”, menjadi salah, karena Kinematika bukan merupakan bagian dari Matematika, sehingga bila menggunakan penalaran induktif sangat dimungkinkan adanya ketidakpastian.

Representasi Logika dibagi menjadi dua:

1. *Propositional Logic* (Logika Proposisi)

Suatu Proposisi merupakan suatu statemen atau pernyataan yang menyatakan benar (TRUE) atau salah (FALSE). Dalam Propositional Logic fakta dilambangkan dengan simbol misalnya P, Q dan R. Lambang-lambang tersebut dihubungkan dengan relasi-relasi logika menggunakan operator logika.

Dua atau lebih proposisi dapat digabungkan dengan menggunakan operator logika:

- a. Konjungsi : $\wedge$ (and)
- b. Disjungsi : $\vee$ (or)
- c. Negasi : $\neg$ (not)
- d. Implikasi : (if then)
- e. Ekuivalensi : $\leftrightarrow$ (if and only if)

Untuk melakukan inferensi pada logika proposisi dapat dilakukan dengan menggunakan resolusi. Resolusi adalah suatu aturan untuk melakukan inferensi yang dapat berjalan secara efisien dalam suatu bentuk khusus yaitu *conjunctive normal form* (CNF), ciri – cirinya:

- 1) Setiap kalimat merupakan disjungsi literal
- 2) Semua kalimat terkonjungsi secara implisit

Langkah-langkah untuk mengubah suatu kalimat (konversi) ke bentuk CNF:

- 1) Hilangkan implikasi dan ekuivalensi
 - xy menjadi $\neg x \vee y$
 - $x \leftrightarrow y$ menjadi $(\neg x \vee y) \wedge (\neg y \vee x)$
- 2) Kurangi lingkup semua negasi menjadi satu negasi saja
 - $\neg(\neg x)$ menjadi x
 - $\neg(x \vee y)$ menjadi $(\neg x \wedge \neg y)$
 - $\neg(x \wedge y)$ menjadi $(\neg x \vee \neg y)$
- 3) Gunakan aturan assosiatif dan distributif untuk mengkonversi menjadi conjunction of disjunction
 - Assosiatif: $(A \vee B) \vee C$ menjadi $A \vee (B \vee C)$
 - Distributif: $(A \wedge B) \vee C$ menjadi $(A \vee C) \wedge (B \vee C)$
- 4) Buat satu kalimat terpisah untuk tiap-tiap konjungsi

Contoh:

- P: Ani anak yang cerdas
- Q: Ani rajin belajar
- R: Ani akan menjadi juara kelas
- S: Ani makannya banyak
- T: Ani istirahatnya cukup

Kalimat yang terbentuk:

- Ani anak yang cerdas
- Jika ani anak yang cerdas dan ani rajin belajar, maka ani akan menjadi juara kelas
- Jika ani makannya banyak atau ani istirahatnya cukup, maka ani rajin belajar
- Ani istirahatnya cukup

Setelah dilakukan konversi ke bentuk CNF, didapat:

- Fakta ke-2: Ani tidak cerdas atau ani tidak rajin belajar atau ani akan menjadi juara kelas
- Fakta ke-3: Ani tidak makan banyak atau ani rajin belajar

- Fakta ke-4: Ani tidak cukup istirahat atau ani rajin belajar

2. *Predicate Logic* (Logika Predikat)

Pada logika predikat proposisi dibedakan menjadi argumen (objek) dan predikat (keterangan). Secara umum penulisan proposisi dalam logika predikat dapat dinyatakan sebagai berikut: Predikat (argumen-1, argumen-2,..., argumen-3)

Representasi Fakta Sederhana

Misal diketahui fakta-fakta sebagai berikut:

- Andi adalah seorang laki-laki: A
- Ali adalah seorang laki-laki: B
- Amir adalah seorang laki-laki: C
- Anto adalah seorang laki-laki: D
- Agus adalah seorang laki-laki: E

Jika kelima fakta tersebut dinyatakan dengan menggunakan proposisi, maka akan terjadi pemborosan, dimana beberapa pernyataan dengan predikat yang sama akan dibuat dalam proposisi yang berbeda.

Logika predikat digunakan untuk merepresentasikan hal-hal yang tidak dapat direpresentasikan dengan menggunakan logika proposisi. Pada logika predikat kita dapat merepresentasikan fakta-fakta sebagai suatu pernyataan yang disebut dengan wff (well – formed formula). Logika predikat merupakan dasar bagi bahasa AI seperti bahasa pemrograman PROLOG

Pada contoh diatas, dapat dituliskan:

- Laki-laki(x)

Dimana x adalah variabel yang disubstitusikan dengan andi, ali, amir, anto, agus, dan laki-laki yang lain.

Dalam logika predikat, suatu proposisi atau premis dibagi menjadi 2 bagian, yaitu argumen (objek) dan predikat (keterangan). Argumen adalah individu atau

objek yang membuat keterangan. Predikat adalah keterangan yang membuat argumen dan predikat.

Contoh:

- a. Jika besok tidak hujan, Tommy pergi ke gunung
 $\neg \text{cuaca}(\text{hujan, besok}) \text{ pergi}(\text{tommy, gunung})$
- b. Diana adalah nenek dari ibu Amir
 $\text{nenek}(\text{Diana, ibu}(\text{Amir}))$
- c. Mahasiswa berada di dalam kelas
 $\text{didalam}(\text{mahasiswa, kelas})$

Dari contoh diatas dapat dijabarkan sebagai berikut:

- Di dalam = predikat (keterangan)
 - Mahasiswa = argumen (objek)
 - Kelas = argumen (objek)
- d. Johan suka Maria
 $\text{suka}(\text{johan, maria})$
 - e. Pintu terbuka
 $\text{Buka}(\text{pintu})$
 - f. Johan suka Maria
Ramon suka Maria
Misal: Johan = x, Maria = y, Ramon = z
Maka: $\text{suka}(x, y) \wedge \text{suka}(z, y)$ tidak $\text{suka}(x, z)$
Dibaca: Jika Johan suka Maria dan Ramon suka Maria, maka Johan tidak suka Ramon

9.4 List dan Tree

List dan Tree merupakan struktur sederhana yang digunakan dalam representasi hirarki pengetahuan.

9.4.1 List

Adalah daftar dari rangkaian materi yang terkait. Hal ini bisa merupakan suatu daftar (list) nama orang yang anda kenal, barang-barang yang akan dibeli dari toko Serba Ada, hal-

hal yang akan dikerjakan minggu ini, atau produk-produk berbagai jenis barang dalam katalog, dll.

List biasanya digunakan untuk merepresentasikan hirarki pengetahuan dimana objek dikelompokkan, dikategorikan atau digabungkan sesuai dengan urutan atau hubungannya. Objek dibagi dalam kelompok atau jenis yang sama. Kemudian hubungan ditampilkan dengan menghubungkan satu sama lain.

9.4.2 Tree

Struktur pohon adalah struktur grafik hirarki. Struktur ini merupakan cara yang sederhana untuk menggambarkan list dan hirarki pengetahuan lainnya.

9.5 Jaringan Sematik

Jaringan semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tersebut. Objek disini bisa berupa benda atau peristiwa. Antara 2 objek dihubungkan oleh arc yang menunjukkan hubungan antar objek. Pengetahuan disusun dalam sebuah jaringan yang memiliki komponen utama:

1. Node: menyatakan obyek, konsep, atau situasi. Dinyatakan dengan kotak atau lingkaran
2. Arcs/Link: Menyatakan hubungan antar node. Dinyatakan dengan tanda panah.

9.6 Frame

Frame merupakan kumpulan pengetahuan tentang suatu objek tertentu, peristiwa, lokasi, situasi, dll. Frame memiliki slot yang menggambarkan rincian (atribut) dan karakteristik objek.

Frame biasanya digunakan untuk merepresentasikan pengetahuan yang didasarkan pada karakteristik yang sudah dikenal, yang merupakan pengalaman-pengalaman.

Dengan menggunakan frame, sangat mudah untuk membuat inferensi tentang objek, peristiwa, atau situasi baru, karena frame menyediakan basis pengetahuan yang ditarik dari pengalaman.

Tiga komponen utama dari frame

1. Frame name
2. Attributes (slots)
3. Values (subslots)

Plot direpresentasikan melalui frame berupa atribut yang menjelaskan objek. Subplot adalah knowledge dari atribut. Beberapa informasi pada plot:

1. Frame yang mempunyai identitas
2. Frame yang mempunyai hubungan dengan frame tetangga
3. Deskripsi yang diperlukan frame
4. Informasi deskripsi tersebut
5. Default frame yang termasuk informasi
6. Kebaruan informasi

Lain halnya dengan subplot yang terdapat banyak nilai yaitu:

1. Values : atribut yang memiliki nilai
2. Defaults : slot kosong yang tidak memiliki nilai
3. Ranges : kategori informasi
4. IfAddes : slot yang telah terisi dan mempunyai informasi serta siap untuk diambil tindakan
5. IfNeeded : tidak terdapat nilai pada slot sehingga dengan keadaan tersebut digunakan subslot ini
6. Others : input slot dengan frame ataupun rule bisa juga menggunakan semantik berupa jaringan serta yang lainnya

9.7 Script (Naskah)

Script adalah skema representasi pengetahuan yang sama dengan frame, yaitu merepresentasikan pengetahuan berdasarkan karakteristik yang sudah dikenal sebagai pengalaman-pengalaman. Perbedaananya, frame menggambarkan objek, sedangkan script menggambarkan urutan peristiwa. Dalam menggambarkan urutan peristiwa, script menggunakan slot yang berisi informasi tentang orang, objek, dan tindakan-tindakan yang terjadi dalam suatu peristiwa.

Elemen script meliputi:

1. Kondisi input, yaitu kondisi yang harus dipenuhi sebelum terjadi atau berlaku suatu peristiwa dalam script
2. Track, yaitu variasi yang mungkin terjadi dalam suatu script
3. Prop, berisi objek-objek pendukung yang digunakan selama peristiwa terjadi
4. Role, yaitu peran yang dimainkan oleh seseorang dalam peristiwa
5. Scene, yaitu adegan yang dimainkan yang menjadi bagian dari suatu peristiwa
6. Hasil, yaitu kondisi yang ada setelah urutan peristiwa dalam script terjadi.

Contoh script kejadian pada saat **Pesan Antar Makanan**

Script: Pesan Antar Makanan

Jalur (track): pesan antar makanan via telepon

Role (peran): pesanan, call service, koki, kasir, kurir

Prop (pendukung): rumah makan, telepon, komputer (database), perlengkapan masak, bahan-bahan masakan, kendaraan

Kondisi input: pemesan memesan menu dengan memberitahu nama, alamat, dan nomor telepon kepada call center

Adegan (scene) – 1: Pemesan memesan makanan

1. Pemesan menghubungi rumah makan yang dituju
2. Pemesan menyebutkan menu yang ingin dipesan
3. Call service mengecek ketersediaan menu yang dipesan
4. Pemesan menyebutkan nama, alamat, dan nomor telepon kepada call center
5. Call senter menyebutkan jumlah menu yang dipesan dan biayanya

Adegan (scene) – 2: Rumah makan mempersiapkan pesanan:

1. Koki memasak menu makanan yang dipesan
2. Koki membungkus rapi paket menu yang dipesan
3. Kasir membuat struk pemesanan dan menyerahkannya kepada kurir
4. Koki memberikan paket menu tersebut kepada kurir

Adegan (scene) – 3: Kurir mengantar paket pesanan:

1. Kurir menerima dari koki paket menu yang dipesan
2. Kurir membawa struk yang berisi nama, alamat, dan nomor telepon pesanan, beserta jumlah yang harus dibayarkan oleh pemesan
3. Kurir memasukkan paket pesanan ke dalam box
4. Kurir berangkat mencari alamat pemesan
5. Kurir tiba di tujuan
6. Kurir menanyakan kebenaran alamat pemesan

Adegan (scene) – 4: Pemesan menerima pesanan:

1. Pemesan menerima paket pesanan dan struk dari kurir
2. Pemesan mengecek paket, apakah sesuai pesanan atau tidak
3. Pemesan membayar paket pesanan tersebut kepada kurir
4. Pemesan memberi tips kepada kurir
5. Kurir menerima uang pembayaran

Hasil

1. Kurir senang dan berterimakasih
2. Pemesan senang
3. Pemesan kenyang
4. Pemesan puas
5. Pemesan kecewa
6. Pemesan masih lapar

9.8 Aturan Produksi (Kaidah Produksi)

Representasi pengetahuan dengan sistem produksi berupa aplikasi aturan (*rule*) yang berupa: (Costello, 2019)

1. Antecedent, yaitu bagian yang mengekspresikan situasi atau premis (pernyataan berawalan IF)
2. Konsekuen, yaitu bagian yang menyatakan suatu tindakan tertentu atau konklusi yang diterapkan jika suatu situasi atau premis bernilai benar (pernyataan berawalan THEN)

Sistem Produksi secara umum terdiri atas komponen-komponen:

1. Ruang keadaan, yang berisi keadaan awal (*initial state*), tujuan (*goal state*) dan kumpulan aturan (*production rule*) yang digunakan untuk mencapai tujuan.
2. Strategi Kontrol, yang berguna untuk mengarahkan bagaimana proses pencarian akan berlangsung dan mengendalikan arah eksplorasi.

Konsekuensi atau konklusi yang dinyatakan pada bagian THEN baru dinyatakan benar, jika bagian IF pada sistem tersebut juga benar atau sesuai dengan aturan tertentu.

Contoh :

IF lalulintas pagi ini padat

THEN saya naik sepeda motor saja

Aturan dapat ditulis dalam beberapa bentuk :

1. IF premis THEN kesimpulan
Jika pendapatan tinggi MAKA pajak yang harus dibayar juga tinggi
2. Kesimpulan IF premis
Pajak yang harus dibayar tinggi JIKA pendapatan tinggi
3. Inclusion of ELSE
IF pendapatan tinggi OR pengeluaran tinggi, THEN pajak yang harus dibayar tinggi
ELSE pajak yang harus dibayar rendah
4. Aturan yang lebih kompleks
IF rating kredit tinggi AND gaji lebih besar dari \$30,000 OR aset lebih dari \$75,000 AND sejarah pembayaran tidak miskin THEN pinjaman diatas \$ 10,000 disetujui dan daftar pinjaman masuk kategori "B"

Apabila pengetahuan direpresentasikan dengan aturan, maka ada 2 metode penalaran yang dapat digunakan :

1. *Forward Reasoning* (penalaran maju)
Pelacakan dimulai dari keadaan awal (informasi atau fakta yang ada) dan kemudian dicoba untuk mencocokkan dengan tujuan yang diharapkan
2. *Backward Reasoning* (penalaran mundur)
Penalaran dimulai dari tujuan atau hipotesa, baru dicocokkan dengan keadaan awal atau fakta- fakta yang ada.

Ada beberapa faktor yang mempengaruhi pemilihan backward atau forward dalam memilih metode penalaran:

1. Banyaknya keadaan awal dan tujuan. Jika jumlah keadaan awal lebih kecil daripada tujuan, maka digunakan penalaran forward. Sebaliknya jika jumlah tujuan lebih banyak daripada keadaan awal, maka dipilih penalaran backward

2. Rata-rata jumlah node yang dapat diraih langsung dari suatu node. Lebih baik dipilih yang jumlah node tiap cabangnya lebih sedikit
3. Apakah program butuh menanyai user untuk melakukan justifikasi terhadap proses penalaran? Jika ya, maka alangkah baiknya jika dipilih arah yang lebih memudahkan user
4. Bentuk kejadian yang akan memicu penyelesaian masalah. Jika kejadian itu berupa fakta baru, maka lebih baik dipilih penalaran forward. Namun jika kejadian itu berupa query, maka lebih baik digunakan penalaran backward.

DAFTAR PUSTAKA

- Bench-Capon, T.J.M., 2014. Knowledge Representation: An Approach to Artificial Intelligence.
- Brachman, R.J. and Levesque, H.J., 2004. Knowledge Representation and Reasoning.
- Costello, D., 2019. Knowledge Representation About knowledge representation.
- Crawford, E., Gingerich, M. and Eliasmith, C., 2016. Biologically Plausible, Human-scale Knowledge Representation. *Cognitive science*, 40 4, pp.782–821.
- Gabbay, D.M., Horty, J.F., Parent, X., van der Meyden, R. and van der Torre, L., 2013. Handbook of deontic logic and normative systems.

BAB 10

SISTEM BERBASIS PENGETAHUAN: KONSEP, PERANCANGAN, DAN IMPLEMENTASI

Oleh: St. Hajrah Mansyur

10.1 Pendahuluan

10.1.1 Apa itu sistem berbasis pengetahuan?

Sistem Berbasis Pengetahuan (*Knowledge-based System*) adalah jenis sistem komputer yang dirancang untuk menyimpan, mengelola, dan menerapkan pengetahuan dalam suatu domain tertentu. Pengetahuan yang dimaksud di sini mencakup aturan, fakta.

10.1.2 Sejarah dan Perkembangan Sistem Berbasis Pengetahuan

Sistem Berbasis Pengetahuan (*Knowledge-based System*) memiliki sejarah yang cukup panjang dan telah mengalami perkembangan signifikan seiring waktu. Berikut adalah beberapa tonggak sejarah dan perkembangan penting dalam sistem berbasis pengetahuan:

1. Awal Pengembangan Sistem Berbasis Pengetahuan (1960-an - 1970-an)
 - a. Pada tahun 1960-an, peneliti-peneliti di bidang kecerdasan buatan mulai mengembangkan sistem yang dapat menyimpan pengetahuan dan menggunakan pengetahuan tersebut untuk memecahkan masalah.
 - b. Pada tahun 1970-an, pendekatan berbasis aturan mulai dikembangkan, di mana pengetahuan diwakili dalam

bentuk aturan-aturan logika yang dapat digunakan untuk mengambil keputusan atau melakukan inferensi.

2. Sistem Pakar (1980-an)
 - a. Pada tahun 1980-an, perkembangan sistem berbasis pengetahuan difokuskan pada pengembangan sistem pakar. Sistem pakar adalah sistem berbasis pengetahuan yang meniru pengetahuan dan kemampuan pemecahan masalah dari seorang pakar manusia di bidang tertentu.
 - b. Sistem pakar mulai digunakan dalam berbagai aplikasi, seperti diagnosis medis, analisis keuangan, perencanaan produksi, dan sebagainya.
3. Perkembangan Metode Inferensi (1990-an)
 - a. Pada tahun 1990-an, perkembangan metode inferensi dalam sistem berbasis pengetahuan menjadi fokus utama. Metode-metode seperti forward chaining, backward chaining, dan rule-based reasoning diperkenalkan dan dikembangkan lebih lanjut.
 - b. Sistem berbasis pengetahuan mulai mengintegrasikan teknik-teknik logika, probabilistik, dan fuzzy logic untuk meningkatkan kemampuan inferensinya.
4. Integrasi dengan Teknologi lain (2000-an - Sekarang)
 - a. Dalam era ini, sistem berbasis pengetahuan mulai diintegrasikan dengan teknologi lain seperti data mining, mesin pencarian, kecerdasan buatan, dan big data.
 - b. Sistem berbasis pengetahuan juga diterapkan dalam lingkungan yang lebih luas, seperti sistem penunjang keputusan, sistem rekomendasi, analisis data, robotika, kendaraan otonom, dan lain sebagainya.

Perkembangan sistem berbasis pengetahuan terus berlanjut dengan adanya peningkatan kemampuan komputasi,

perluasan aplikasi, dan penemuan-penemuan baru dalam bidang kecerdasan buatan. Sistem berbasis pengetahuan terus menjadi alat yang berguna dalam membantu manusia dalam mengelola, menganalisis, dan memanfaatkan pengetahuan di berbagai domain.

10.1.3 Manfaat dan Aplikasi Sistem Berbasis Pengetahuan

Sistem Berbasis Pengetahuan (*Knowledge-based System*) memiliki manfaat yang signifikan dalam berbagai bidang. Berikut adalah beberapa manfaat dan aplikasi utama dari sistem berbasis pengetahuan:

1. Pengambilan Keputusan

Sistem berbasis pengetahuan dapat membantu dalam pengambilan keputusan kompleks dengan menerapkan pengetahuan yang telah terstruktur. Contoh aplikasi: Sistem penunjang keputusan di bidang keuangan, manajemen proyek, analisis risiko, diagnosis medis, perencanaan produksi, dan lain-lain.

2. Sistem Pakar

Sistem berbasis pengetahuan digunakan dalam pengembangan sistem pakar yang dapat meniru pengetahuan dan keahlian pakar manusia dalam domain tertentu. Contoh aplikasi: Sistem pakar dalam bidang medis, hukum, keuangan, rekayasa, pertanian, dan banyak lagi.

3. Pencarian dan Pengelolaan Informasi

Sistem berbasis pengetahuan dapat digunakan untuk menyimpan, mengatur, dan mencari informasi dengan efisien berdasarkan pengetahuan yang terstruktur. Contoh aplikasi: Sistem manajemen pengetahuan, basis data berbasis pengetahuan, mesin pencari berbasis pengetahuan, dan sebagainya.

4. Sistem Rekomendasi

Sistem berbasis pengetahuan dapat digunakan untuk memberikan rekomendasi yang personal dan relevan kepada pengguna berdasarkan pengetahuan yang diketahui. Contoh aplikasi: Rekomendasi produk atau konten dalam e-commerce, rekomendasi film atau musik, rekomendasi pengobatan medis, dan lain sebagainya.

5. Otomatisasi dan Penghematan Waktu

Sistem berbasis pengetahuan dapat mengotomatisasi tugas-tugas rutin, mempercepat proses pengambilan keputusan, dan mengurangi kesalahan manusia. Contoh aplikasi: Otomatisasi proses bisnis, sistem pelayanan pelanggan otomatis, sistem pengelolaan inventaris, dan lain-lain.

6. Pendidikan dan Pelatihan

Sistem berbasis pengetahuan dapat digunakan sebagai alat pembelajaran yang interaktif dan mendukung proses pembelajaran berbasis pengetahuan. Contoh aplikasi: Sistem tutor berbasis pengetahuan, e-learning, sistem bimbingan karir, dan sebagainya.

Manfaat dan aplikasi sistem berbasis pengetahuan terus berkembang seiring dengan kemajuan teknologi dan kebutuhan dalam berbagai industri dan sektor. Sistem berbasis pengetahuan memberikan nilai tambah dalam mengelola pengetahuan, meningkatkan efisiensi, dan mendukung pengambilan keputusan yang lebih baik.

10.2 Konsep Dasar Sistem Berbasis Pengetahuan

10.2.1 Pengenalan Pengetahuan

a. Definisi pengetahuan dan jenis-jenisnya

Pengetahuan secara umum dapat didefinisikan sebagai kumpulan informasi, fakta, konsep, pengalaman, dan pemahaman yang dimiliki oleh individu atau entitas.

Pengetahuan memberikan pemahaman tentang sesuatu dan memungkinkan individu untuk mengenali, menginterpretasikan, dan menggunakan informasi dengan cara yang berguna.

Jenis-jenis Pengetahuan

1. Pengetahuan Faktual (*Declarative Knowledge*)
Pengetahuan faktual mencakup informasi yang dapat diungkapkan dalam bentuk pernyataan yang benar atau salah. Contoh: Fakta sejarah, data statistik, nama-nama benda, dan sebagainya.
2. Pengetahuan Prosedural (*Procedural Knowledge*)
Pengetahuan prosedural terkait dengan pemahaman tentang cara melakukan suatu tindakan atau prosedur. Contoh: Pengetahuan tentang cara mengemudi mobil, memasak suatu hidangan, atau merakit perangkat elektronik.
3. Pengetahuan Konseptual (*Conceptual Knowledge*)
Pengetahuan konseptual melibatkan pemahaman tentang konsep, prinsip, dan hubungan antara ide-ide. Contoh: Pemahaman tentang prinsip fisika, teori matematika, prinsip ekonomi, dan sebagainya.
4. Pengetahuan Heuristik (*Heuristic Knowledge*)
Pengetahuan heuristik melibatkan aturan praktis, strategi, atau panduan yang digunakan dalam pengambilan keputusan atau pemecahan masalah. Contoh: Heuristik dalam pengambilan keputusan bisnis, metode cepat dalam menyelesaikan teka-teki, atau tips dalam mencapai tujuan tertentu.
5. Pengetahuan Kontekstual (*Contextual Knowledge*)
Pengetahuan kontekstual terkait dengan pemahaman tentang konteks atau situasi tertentu yang mempengaruhi interpretasi dan penggunaan pengetahuan. Contoh: Pemahaman tentang norma

sosial, kebiasaan budaya, atau tuntutan lingkungan tertentu.

6. Pengetahuan Metakognitif (*Metacognitive Knowledge*)

Pengetahuan metakognitif berkaitan dengan pemahaman tentang proses berpikir, pemantauan, dan pengaturan diri terhadap pengetahuan dan keterampilan. Contoh: Kesadaran akan kekuatan dan kelemahan diri sendiri, strategi belajar yang efektif, atau pemahaman tentang bagaimana belajar secara efisien.

Jenis-jenis pengetahuan ini saling terkait dan sering kali digunakan bersama-sama dalam konteks yang berbeda. Penggunaan kombinasi pengetahuan ini memungkinkan individu atau sistem untuk memahami dan menghadapi situasi dengan lebih baik serta membuat keputusan yang lebih baik pula.

b. Perbedaan antara data, informasi, dan pengetahuan

Data, informasi, dan pengetahuan adalah konsep yang saling terkait tetapi memiliki perbedaan penting dalam konteks penggunaan dan tingkat pemrosesan. Berikut adalah perbedaan antara data, informasi, dan pengetahuan:

1. Data

- Data adalah fakta mentah atau kumpulan simbol-simbol yang belum memiliki arti yang jelas atau konteks.
- Data tidak memiliki interpretasi atau konteks yang terkait secara langsung dengan pemahaman atau tindakan.
- Contoh: Sekumpulan angka, huruf, atau simbol yang belum diolah menjadi informasi yang berarti, seperti "120", "A", "1, 2, 3".

2. Informasi

- Informasi adalah data yang telah diolah, diorganisasi, dan diberikan konteks sehingga memiliki makna yang bermanfaat bagi penerima.
- Informasi memberikan pemahaman atau wawasan tentang sesuatu dan dapat digunakan untuk mengambil keputusan atau melakukan tindakan.
- Informasi diperoleh dari pengolahan data dengan memberikan struktur, hubungan, atau interpretasi.
- Contoh: Hasil perhitungan matematika, laporan penjualan bulanan, grafik statistik, atau ringkasan berita.

3. Pengetahuan

- Pengetahuan melibatkan pemahaman dan interpretasi yang lebih dalam dari informasi yang dimiliki.
- Pengetahuan melampaui sekadar penerimaan dan pemrosesan informasi, tetapi juga mencakup pemahaman konteks, kaitan, dan implikasi dari informasi tersebut.
- Pengetahuan melibatkan penggunaan informasi dalam konteks yang lebih luas, memahami prinsip, aturan, atau konsep di balik informasi tersebut.
- Pengetahuan juga mencakup pengalaman, wawasan, dan kemampuan untuk mengaplikasikan informasi dalam situasi nyata.
- Contoh: Pemahaman tentang prinsip matematika, keahlian dalam suatu bidang tertentu, atau kemampuan menerapkan pengetahuan dalam pemecahan masalah.

Secara umum, data menjadi informasi ketika diolah dan diberikan konteks yang relevan. Informasi kemudian menjadi pengetahuan ketika dikaitkan dengan pemahaman

yang lebih dalam, pengalaman, dan kemampuan untuk mengaplikasikannya. Proses ini melibatkan transformasi dari level yang lebih rendah (data) menjadi level yang lebih tinggi (informasi) dan kemudian ke level pengetahuan yang lebih tinggi lagi.

Data, informasi, dan pengetahuan merupakan konsep yang membentuk suatu tangga pemahaman yang lebih tinggi. Berikut adalah lanjutan perbedaan antara ketiga konsep tersebut:

1. Karakteristik

- Data: Data bersifat mentah, tidak berarti, dan belum diolah.
- Informasi: Informasi dihasilkan dari pengolahan data dan memiliki arti atau konteks yang relevan.
- Pengetahuan: Pengetahuan melibatkan pemahaman yang lebih dalam, interpretasi, dan aplikasi informasi dalam konteks yang luas.

2. Tingkat pemrosesan

- Data: Data merupakan input awal yang harus diolah dan dianalisis untuk menghasilkan informasi.
- Informasi: Informasi dihasilkan melalui pengolahan, pengorganisasian, dan interpretasi data.
- Pengetahuan: Pengetahuan melibatkan pemahaman yang lebih tinggi dari informasi melalui kaitannya dengan konteks, konsep, dan pengalaman.

3. Arti dan relevansi

- Data: Data memiliki arti dan relevansi yang terbatas tanpa pengolahan dan interpretasi lebih lanjut.
- Informasi: Informasi memberikan arti dan relevansi yang lebih besar, dapat digunakan untuk pengambilan keputusan atau tindakan.
- Pengetahuan: Pengetahuan memiliki arti dan relevansi yang lebih dalam, melibatkan pemahaman

konteks, keterkaitan, dan aplikasi dalam situasi nyata.

4. Sumber daya berharga

- Data: Data sendiri belum memberikan nilai yang signifikan tanpa pengolahan dan transformasi lebih lanjut.
- Informasi: Informasi memiliki nilai yang berharga karena memberikan pemahaman dan wawasan yang berguna.
- Pengetahuan: Pengetahuan merupakan aset berharga karena melibatkan pemahaman mendalam, pengalaman, dan kemampuan untuk mengaplikasikannya dalam situasi yang kompleks.

Penting untuk memahami bahwa data, informasi, dan pengetahuan semuanya berperan penting dalam pengambilan keputusan, pembelajaran, dan pengembangan. Transformasi dari data menjadi pengetahuan melibatkan pengolahan dan interpretasi yang cermat untuk mencapai pemahaman yang lebih dalam dan aplikasi yang efektif.

c. Proses pemerolehan pengetahuan

Proses pemerolehan pengetahuan adalah bagaimana individu atau sistem mengumpulkan, memperoleh, dan membangun pengetahuan baru. Ada beberapa metode atau proses yang digunakan dalam pemerolehan pengetahuan, termasuk:

1. Pengamatan

Pengamatan melibatkan penggunaan panca indera untuk mengumpulkan informasi tentang lingkungan atau objek yang diamati. Proses ini melibatkan pengumpulan data mentah atau fakta yang dapat membentuk dasar bagi pembentukan pengetahuan.

2. Pengalaman

Pengalaman adalah proses pembelajaran yang melibatkan interaksi dengan lingkungan dan situasi nyata. Melalui pengalaman, individu atau sistem dapat mengumpulkan pengetahuan dari hasil tindakan, percobaan, dan interaksi yang dilakukan.

3. Pembelajaran

Pembelajaran melibatkan proses memperoleh pengetahuan melalui studi, pelatihan, pengajaran, atau proses edukasi. Melalui pembelajaran formal atau informal, individu atau sistem dapat memperoleh pengetahuan dari sumber yang lebih berpengalaman atau ahli dalam bidang tertentu.

4. Komunikasi

Komunikasi melibatkan pertukaran informasi dan pengetahuan antara individu atau entitas yang berbeda. Melalui interaksi sosial, diskusi, kolaborasi, dan pertukaran ide, individu atau sistem dapat memperoleh pengetahuan dari orang lain.

5. Penelitian

Penelitian melibatkan eksplorasi sistematis dan metode ilmiah untuk memperoleh pengetahuan baru. Proses ini melibatkan pengumpulan data, analisis, pengujian hipotesis, dan kesimpulan yang dapat menghasilkan pengetahuan yang lebih mendalam dan teruji secara empiris.

6. Pengolahan Informasi

Pengolahan informasi melibatkan transformasi data mentah menjadi informasi yang berarti melalui pemrosesan dan analisis. Dalam proses ini, individu atau sistem dapat mengidentifikasi pola, hubungan, atau aturan yang membentuk pengetahuan baru.

7. Refleksi dan Evaluasi

Refleksi dan evaluasi melibatkan proses introspeksi dan pemikiran kritis terhadap pengalaman, pengetahuan, dan tindakan yang telah dilakukan. Melalui proses ini, individu atau sistem dapat mengidentifikasi kekuatan, kelemahan, dan area peningkatan yang mempengaruhi pemerolehan pengetahuan.

Pemerolehan pengetahuan adalah proses yang berkesinambungan, di mana individu atau sistem terus-menerus memperoleh, mengembangkan, dan mengintegrasikan pengetahuan baru dengan pengetahuan yang sudah ada. Melalui kombinasi metode-metode ini, pengetahuan dapat tumbuh dan berkembang seiring waktu.

10.2.2 Sistem Berbasis Pengetahuan

a. Definisi sistem berbasis pengetahuan

Sistem Berbasis Pengetahuan (*Knowledge-Based System*) adalah sistem komputer yang dirancang untuk menyimpan, mengelola, dan menggunakan pengetahuan dalam memecahkan masalah kompleks. Sistem ini menggunakan basis pengetahuan yang terstruktur dan terorganisir untuk memodelkan pemahaman manusia atau keahlian dalam bidang tertentu.

Sistem Berbasis Pengetahuan menggabungkan pengetahuan eksplisit (yang dapat dinyatakan dalam bentuk aturan, fakta, atau konsep) dengan kemampuan pemrosesan komputer untuk memecahkan masalah, membuat keputusan, atau memberikan rekomendasi. Basis pengetahuan dalam sistem ini dapat berupa himpunan aturan (*rule-based systems*), jaringan semantik, sistem berbasis kasus (*case-based systems*), atau bahkan ontologi.

Salah satu komponen penting dari sistem berbasis pengetahuan adalah mesin inferensi (*inference engine*), yang bertugas mengambil pengetahuan dari basis pengetahuan dan menerapkannya pada data yang diberikan untuk menghasilkan kesimpulan atau solusi. Mesin inferensi ini menggunakan aturan-aturan logika atau mekanisme pemrosesan yang telah ditentukan untuk menghubungkan pengetahuan yang ada dalam basis pengetahuan.

Sistem Berbasis Pengetahuan digunakan dalam berbagai bidang dan aplikasi, termasuk sistem pakar (*expert systems*), diagnosis medis, sistem rekomendasi, analisis keputusan, pengenalan pola, dan banyak lagi. Kelebihan sistem berbasis pengetahuan adalah kemampuannya untuk menyimpan pengetahuan yang kompleks dan melakukan pemrosesan yang lebih cerdas dan kontekstual berdasarkan pengetahuan tersebut.

Dalam sistem berbasis pengetahuan, pengetahuan dapat diperbarui atau diperluas dengan menambahkan aturan baru atau informasi baru ke dalam basis pengetahuan. Hal ini memungkinkan sistem untuk terus berkembang seiring waktu dan menggabungkan pengetahuan baru yang relevan dalam memecahkan masalah yang kompleks dan beragam.

b. Komponen-Komponen Dalam Sistem Berbasis Pengetahuan.

Sistem Berbasis Pengetahuan terdiri dari beberapa komponen penting yang bekerja bersama untuk menyimpan, mengelola, dan menggunakan pengetahuan. Berikut adalah beberapa komponen utama dalam Sistem Berbasis Pengetahuan:

1. Basis Pengetahuan (*Knowledge Base*)

Basis Pengetahuan adalah komponen sentral dalam sistem berbasis pengetahuan yang menyimpan

pengetahuan yang relevan untuk domain atau bidang tertentu. Basis pengetahuan dapat berupa himpunan aturan (*rule-based*), jaringan semantik, graf pengetahuan, atau ontologi yang merepresentasikan fakta, konsep, aturan, dan hubungan dalam domain tersebut. Basis pengetahuan membentuk dasar pemahaman dan pemecahan masalah dalam sistem.

2. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah komponen yang bertanggung jawab untuk mengambil pengetahuan dari basis pengetahuan dan menerapkannya pada data atau fakta yang diberikan. Mesin inferensi menggunakan aturan-aturan logika atau metode pemrosesan lainnya untuk melakukan inferensi atau penalaran berdasarkan pengetahuan yang ada. Mesin inferensi menghasilkan kesimpulan, rekomendasi, atau solusi berdasarkan pengetahuan yang diterapkan pada data input.

3. Antarmuka Pengguna

Antarmuka pengguna adalah komponen yang memungkinkan interaksi antara pengguna dan sistem berbasis pengetahuan. Antarmuka pengguna dapat berupa antarmuka grafis, antarmuka berbasis teks, atau kombinasi keduanya yang memungkinkan pengguna untuk memberikan input, menerima output, atau berinteraksi dengan sistem secara intuitif.

4. Mesin Penalaran (*Reasoning Engine*)

Mesin penalaran adalah komponen yang bertanggung jawab untuk menjalankan proses penalaran atau pemikiran logis dalam sistem berbasis pengetahuan. Mesin penalaran memungkinkan sistem untuk membuat inferensi, menarik kesimpulan, mengidentifikasi pola, dan melakukan penalaran yang kompleks berdasarkan pengetahuan yang ada.

5. Basis Kasus (*Case Base*)

Basis kasus adalah komponen yang digunakan dalam sistem berbasis pengetahuan untuk menyimpan dan mengelola kumpulan kasus-kasus yang relevan dalam domain. Kasus-kasus dalam basis kasus berisi informasi tentang situasi masalah, solusi yang diberikan, dan hasil yang diperoleh. Basis kasus dapat digunakan dalam sistem berbasis pengetahuan untuk melakukan penalaran berdasarkan pengalaman sebelumnya dan menemukan solusi yang serupa untuk masalah baru.

6. Mesin Pengambil Keputusan (*Decision Engine*)

Mesin pengambil keputusan adalah komponen yang bertanggung jawab untuk memproses informasi, pengetahuan, dan aturan dalam sistem berbasis pengetahuan untuk menghasilkan keputusan atau rekomendasi. Mesin pengambil keputusan dapat menggunakan teknik-teknik seperti analisis keputusan, pengambilan keputusan berbasis aturan, atau algoritma pengambilan keputusan.

c. Arsitektur dan model dalam sistem berbasis pengetahuan

Dalam sistem berbasis pengetahuan, terdapat beberapa arsitektur dan model yang digunakan untuk merancang dan mengimplementasikan sistem tersebut. Berikut adalah beberapa arsitektur dan model umum dalam sistem berbasis pengetahuan:

1. Arsitektur Berbasis Aturan (*Rule-Based Architecture*)

Arsitektur ini menggunakan himpunan aturan yang terdiri dari kondisi (*premises*) dan tindakan (*conclusions*) yang dipicu oleh kondisi-kondisi tersebut. Aturan-aturan tersebut direpresentasikan dalam bentuk "Jika kondisi tertentu terpenuhi, maka lakukan tindakan tertentu." Mesin inferensi dalam arsitektur ini menggunakan aturan-aturan tersebut untuk melakukan

penalaran dan menghasilkan output berdasarkan fakta atau input yang diberikan.

2. *Arsitektur Berbasis Kasus (Case-Based Architecture)*

Arsitektur ini menggunakan basis kasus yang berisi kumpulan kasus-kasus yang relevan dalam domain tertentu. Kasus-kasus dalam basis kasus berisi informasi tentang situasi masalah, solusi yang diberikan, dan hasil yang diperoleh. Ketika dihadapkan pada masalah baru, sistem mencocokkan masalah tersebut dengan kasus yang serupa dalam basis kasus, dan mengadopsi solusi dari kasus yang paling cocok.

3. *Arsitektur Berbasis Jaringan Semantik (Semantic Network Architecture)*

Arsitektur ini menggunakan representasi grafis yang berupa jaringan hubungan antara konsep-konsep dalam domain. Konsep-konsep tersebut dihubungkan oleh relasi yang mewakili hubungan antara konsep-konsep tersebut. Mesin inferensi dalam arsitektur ini menggunakan jaringan semantik untuk melakukan penalaran dan menghasilkan output berdasarkan konsep-konsep dan relasi di dalamnya.

4. *Arsitektur Berbasis Logika (Logic-Based Architecture)*

Arsitektur ini menggunakan representasi formal dalam bentuk logika matematika, seperti logika proposisional atau predikat. Aturan-aturan dan fakta-fakta dalam domain dinyatakan menggunakan logika matematika. Mesin inferensi dalam arsitektur ini menggunakan aturan-aturan logika untuk melakukan penalaran dan menghasilkan output berdasarkan input yang diberikan.

5. *Arsitektur Berbasis Ontologi*

Arsitektur ini menggunakan ontologi yang merupakan representasi formal dari konsep, hubungan, dan sifat dalam domain tertentu. Ontologi menyediakan struktur

yang formal dan terdefinisi dengan jelas untuk mewakili pengetahuan dalam domain. Mesin inferensi dalam arsitektur ini menggunakan ontologi untuk melakukan penalaran dan menghasilkan output berdasarkan pengetahuan yang terkandung dalam ontologi.

Setiap arsitektur dan model memiliki kelebihan dan kekurangan yang berbeda tergantung pada kebutuhan dan karakteristik domain yang digunakan. Pilihan arsitektur dan model yang tepat sangat tergantung pada sifat dan kompleksitas masalah yang ingin diselesaikan dalam sistem berbasis pengetahuan.

10.2.3 Representasi Pengetahuan

Representasi Pengetahuan dalam konteks sistem berbasis pengetahuan adalah cara untuk menggambarkan, menyimpan, dan mengorganisasi pengetahuan dalam sistem tersebut. Representasi pengetahuan bertujuan untuk membuat pengetahuan dapat dipahami oleh mesin atau sistem komputer, sehingga dapat digunakan untuk pemrosesan, penalaran, dan pengambilan keputusan.

Ada berbagai macam bentuk representasi pengetahuan yang digunakan dalam sistem berbasis pengetahuan, tergantung pada jenis pengetahuan yang diwakili dan karakteristik dari domain aplikasi. Beberapa bentuk representasi pengetahuan umum meliputi:

1. Aturan (*Rules*)

Representasi pengetahuan berbasis aturan menggunakan himpunan aturan yang terdiri dari kondisi (*premises*) dan tindakan (*conclusions*). Aturan-aturan ini menyatakan hubungan logis antara kondisi yang diberikan dengan tindakan yang harus diambil.

Contoh: Jika suhu lebih dari 30 derajat Celsius, maka hidupkan AC.

2. Fakta (*Facts*)

Representasi pengetahuan berbasis fakta menggunakan pernyataan yang menyatakan keadaan atau hubungan antara entitas dalam domain. Fakta-fakta ini berupa pernyataan yang benar atau tidak benar.

Contoh: Jakarta adalah ibu kota Indonesia.

3. Jaringan Semantik (*Semantic Networks*)

Representasi pengetahuan berbasis jaringan semantik menggunakan grafik yang terdiri dari simpul (*node*) dan hubungan (*edge*) untuk menggambarkan konsep dan hubungan di antara mereka.

Jaringan semantik menunjukkan hubungan semantik antara konsep-konsep dalam domain.

Contoh: [Grafik] Manusia -- merupakan --> Makhluk Hidup.

4. Grafik Konseptual (*Conceptual Graphs*)

Representasi pengetahuan berbasis grafik konseptual menggunakan grafik yang menggambarkan konsep, entitas, atribut, dan hubungan di antara mereka. Grafik konseptual menggabungkan elemen-elemen semantik dan sintaksis untuk merepresentasikan pengetahuan.

Contoh: [Grafik] [Manusia] memiliki [Nama] yang merupakan [String].

5. Logika (*Logic*)

Representasi pengetahuan berbasis logika menggunakan aturan dan pernyataan dalam logika matematika, seperti logika proposisional, logika predikat, atau logika deskriptif. Logika menyediakan formalisme untuk menggambarkan hubungan logis dan inferensi antara pernyataan.

Contoh: Pernyataan: Semua manusia adalah makhluk hidup.
Pernyataan: Seseorang adalah manusia. Konklusi: Seseorang adalah makhluk hidup.

Representasi pengetahuan yang dipilih akan tergantung pada sifat pengetahuan yang ingin diwakili, kompleksitas domain, dan kemampuan sistem dalam memahami dan memanipulasi representasi tersebut. Pemilihan representasi pengetahuan yang tepat sangat penting untuk memastikan sistem berbasis pengetahuan.

10.2.4 Bentuk-Bentuk Representasi Pengetahuan

Bentuk-bentuk representasi pengetahuan dalam sistem berbasis pengetahuan dapat bervariasi tergantung pada jenis pengetahuan yang ingin diwakili dan karakteristik dari domain aplikasi. Berikut ini adalah beberapa bentuk representasi pengetahuan yang umum digunakan:

1. Simbolik

Representasi pengetahuan simbolik menggunakan simbol, aturan, dan struktur yang terdefinisi dengan jelas. Contoh bentuk representasi pengetahuan simbolik adalah logika proposisional, logika predikat, dan aturan berbasis if-then. Pada representasi ini, pengetahuan diungkapkan dalam bentuk simbol-simbol dan relasi antara simbol-simbol tersebut.

2. Grafis

Representasi pengetahuan grafis menggunakan grafik, diagram, atau jaringan untuk menggambarkan konsep, entitas, dan hubungan antara mereka. Contoh bentuk representasi pengetahuan grafis adalah jaringan semantik, grafik konseptual, dan diagram aliran proses. Representasi ini memberikan visualisasi yang jelas tentang struktur pengetahuan dan hubungan antara elemen-elemen dalam domain.

3. Berbasis Kasus

Representasi pengetahuan berbasis kasus menggunakan kumpulan kasus-kasus yang relevan dalam

domain tertentu. Setiap kasus berisi informasi tentang situasi masalah, solusi yang diberikan, dan hasil yang diperoleh. Kasus-kasus ini digunakan untuk melakukan penalaran analogi dan menemukan solusi untuk masalah baru yang mirip dengan kasus yang sudah ada.

4. Berbasis Model

Representasi pengetahuan berbasis model menggunakan model matematika atau model formal lainnya untuk menggambarkan domain atau fenomena yang diamati. Contoh bentuk representasi pengetahuan berbasis model adalah model aliran data, model jaringan saraf, atau model probabilitas. Model ini digunakan untuk mewakili pengetahuan yang rumit dan mencerminkan hubungan yang kompleks antara elemen-elemen dalam domain.

5. Berbasis Ontologi

Representasi pengetahuan berbasis ontologi menggunakan ontologi yang merupakan representasi formal dari konsep, hubungan, dan sifat dalam domain tertentu. Ontologi menyediakan struktur yang formal dan terdefinisi dengan jelas untuk mewakili pengetahuan dalam domain. Representasi ini memungkinkan pengorganisasian yang sistematis dan konsisten dari pengetahuan dalam domain.

6. Statistik dan Data

Representasi pengetahuan berbasis statistik dan data menggunakan teknik-teknik analisis data dan statistik untuk mengidentifikasi pola, tren, atau hubungan dalam data. Representasi ini berfokus pada pemodelan statistik, pengklasifikasian, regresi, atau metode prediktif lainnya untuk mewakili pengetahuan dalam bentuk data yang terukur.

Pemilihan bentuk representasi pengetahuan yang tepat tergantung pada sifat pengetahuan yang ingin diwakili, kompleksitas domain, kemampuan sistem, serta tujuan dan kebutuhan aplikasi yang ingin dicapai.

10.2.5 Logika Proposisional dan Logika Predikat

Logika proposisional dan logika predikat adalah dua jenis logika matematika yang digunakan dalam representasi pengetahuan. Meskipun keduanya berfokus pada pemodelan hubungan logis antara pernyataan, mereka memiliki perbedaan dalam lingkup dan ekspresivitas.

1. Logika Proposisional
 - a. Logika proposisional, juga dikenal sebagai logika kalimat, berkaitan dengan hubungan logis antara pernyataan atomik atau proposisi tunggal.
 - b. Pernyataan dalam logika proposisional tidak terpecah menjadi bagian-bagian yang lebih kecil dan tidak terdapat variabel atau kuantifikasi.
 - c. Pernyataan dalam logika proposisional direpresentasikan menggunakan simbol-simbol logika seperti konjungsi ($\wedge$), disjungsi ($\vee$), implikasi ($\rightarrow$), dan negasi ($\neg$).
 - d. Contoh: Jika A adalah pernyataan "Cuaca cerah" dan B adalah pernyataan "Saya pergi bermain", maka pernyataan "Jika cuaca cerah, maka saya pergi bermain" dapat direpresentasikan sebagai $A \rightarrow B$ dalam logika proposisional.
2. Logika Predikat
 - a. Logika predikat, juga dikenal sebagai logika kuantor, digunakan untuk memodelkan hubungan logis antara objek atau entitas dalam domain.
 - b. Pernyataan dalam logika predikat terdiri dari predikat, konstanta, variabel, dan kuantor.

- c. Predikat menggambarkan sifat atau hubungan antara objek, sedangkan kuantor digunakan untuk menggeneralisasi pernyataan.
- d. Contoh: Jika $P(x)$ adalah predikat "x adalah seorang mahasiswa", dan $Q(x)$ adalah predikat "x pintar", maka pernyataan "Beberapa mahasiswa pintar" dapat direpresentasikan sebagai $\exists x (P(x) \wedge Q(x))$ dalam logika predikat.

Perbedaan utama antara logika proposisional dan logika predikat terletak pada tingkat ekspresivitas dan kemampuan untuk memodelkan hubungan yang kompleks. Logika proposisional lebih sederhana dan cocok untuk menggambarkan hubungan logis antara pernyataan atomik, sementara logika predikat lebih fleksibel dan dapat menggambarkan hubungan antara objek atau entitas dalam domain yang lebih kompleks. Pemilihan antara logika proposisional dan logika predikat tergantung pada tingkat kompleksitas dan kebutuhan representasi pengetahuan yang ingin dicapai.

10.2.6 Representasi Pengetahuan dengan Aturan

Representasi pengetahuan dengan aturan adalah salah satu bentuk representasi pengetahuan dalam sistem berbasis pengetahuan. Representasi ini menggunakan himpunan aturan yang terdiri dari kondisi (*premises*) dan tindakan (*conclusions*) yang dipicu oleh kondisi-kondisi tersebut. Aturan-aturan ini digunakan untuk melakukan penalaran dan pengambilan keputusan dalam sistem.

Berikut adalah contoh representasi pengetahuan dengan aturan:

1. Bentuk Umum Aturan

Jika [kondisi], maka [tindakan].

Contoh: Jika suhu lebih dari 30 derajat Celsius, maka hidupkan AC.

2. Aturan dengan Kondisi Majemuk

Jika [kondisi 1] dan [kondisi 2], maka [tindakan].

Contoh: Jika suhu lebih dari 30 derajat Celsius dan kelembaban lebih dari 70%, maka hidupkan AC.

3. Aturan dengan Kondisi Alternatif

Jika [kondisi 1] atau [kondisi 2], maka [tindakan].

Contoh: Jika suhu lebih dari 30 derajat Celsius atau kelembaban lebih dari 70%, maka hidupkan AC.

4. Aturan dengan Konjungsi dan Disjungsi

Jika [kondisi 1] dan ([kondisi 2] atau [kondisi 3]), maka [tindakan].

Contoh: Jika suhu lebih dari 30 derajat Celsius dan (kelembaban lebih dari 70% atau cuaca panas), maka hidupkan AC.

5. Aturan dengan Aksi Prioritas

Jika [kondisi 1], maka [tindakan 1].

Jika [kondisi 2], maka [tindakan 2].

Contoh: Jika suhu lebih dari 30 derajat Celsius, maka hidupkan kipas angin.

Jika suhu lebih dari 35 derajat Celsius, maka hidupkan AC.

Representasi pengetahuan dengan aturan memungkinkan sistem untuk mengambil keputusan berdasarkan kondisi yang diberikan. Mesin inferensi dalam sistem berbasis aturan melakukan penalaran dengan menerapkan aturan-aturan tersebut pada fakta-fakta atau kondisi-kondisi yang diberikan untuk menghasilkan tindakan yang sesuai. Aturan-aturan ini dapat direpresentasikan dalam bentuk bahasa alami, notasi logika, atau bahasa pemrograman khusus yang digunakan dalam implementasi sistem berbasis pengetahuan.

10.3 Perancangan Sistem Berbasis Pengetahuan

10.3.1 Tahap Analisis

Tahap analisis sistem berbasis pengetahuan melibatkan pemahaman mendalam tentang domain aplikasi, tujuan sistem, dan kebutuhan dan persyaratan pengguna. Tujuan dari tahap analisis adalah untuk mengumpulkan informasi yang diperlukan untuk merancang dan mengembangkan sistem berbasis pengetahuan yang efektif. Selain itu, tahap ini bertujuan untuk mengidentifikasi pengetahuan yang relevan, merancang struktur pengetahuan, dan memahami bagaimana sistem akan digunakan. Berikut adalah tahap-tahap dalam analisis sistem berbasis pengetahuan:

1. Identifikasi Kebutuhan
 - a. Menentukan tujuan sistem berbasis pengetahuan dan kebutuhan pengguna.
 - b. Mengidentifikasi area domain yang akan dicakup oleh sistem dan pengetahuan yang relevan untuk mencapai tujuan tersebut.
2. Analisis Domain
 - a. Melakukan studi mendalam tentang domain aplikasi yang akan direpresentasikan dalam sistem berbasis pengetahuan.
 - b. Menganalisis berbagai sumber pengetahuan yang tersedia, seperti dokumen, ahli domain, atau basis pengetahuan yang ada.
3. Ekstraksi Pengetahuan
 - a. Mengidentifikasi pengetahuan yang dapat diekstraksi dari sumber-sumber yang ada.
 - b. Mewawancarai ahli domain untuk mengumpulkan pengetahuan eksplisit dan tacit yang relevan.
 - c. Menggunakan teknik-teknik seperti analisis dokumen, observasi, atau wawancara untuk mendapatkan

pemahaman yang komprehensif tentang pengetahuan domain.

4. Organisasi Pengetahuan
 - a. Membangun struktur pengetahuan yang logis dan terorganisasi.
 - b. Mengidentifikasi hubungan antara konsep-konsep dalam domain dan merancang skema representasi pengetahuan yang sesuai, seperti aturan, jaringan semantik, atau ontologi.
5. Validasi Pengetahuan
 - a. Memvalidasi pengetahuan yang telah dikumpulkan dan direpresentasikan dalam sistem.
 - b. Melibatkan proses verifikasi dan evaluasi pengetahuan dengan melibatkan ahli domain atau pengguna sistem.
 - c. Mengidentifikasi dan mengatasi kesalahan atau inkonsistensi dalam pengetahuan yang terkumpul.
6. Perancangan Infrastruktur
 - a. Merancang infrastruktur sistem berbasis pengetahuan, termasuk mekanisme penalaran, basis pengetahuan, dan antarmuka pengguna.
 - b. Memilih teknologi dan alat yang sesuai untuk mendukung representasi dan pengolahan pengetahuan.

Tahap analisis sistem berbasis pengetahuan ini membentuk dasar untuk pengembangan dan implementasi sistem yang efektif. Dalam tahap ini, pengetahuan yang diperoleh dan direpresentasikan secara tepat akan menjadi dasar untuk proses penalaran dan pengambilan keputusan dalam sistem berbasis pengetahuan.

1. Identifikasi masalah yang dapat diselesaikan dengan sistem berbasis pengetahuan

Sistem berbasis pengetahuan dapat digunakan untuk menyelesaikan berbagai masalah di berbagai bidang. Beberapa contoh masalah yang dapat diselesaikan dengan menggunakan sistem berbasis pengetahuan adalah sebagai berikut:

a. Diagnosis Penyakit

Sistem berbasis pengetahuan dapat digunakan untuk membantu dokter dalam mendiagnosis penyakit dengan memanfaatkan basis pengetahuan medis. Sistem dapat menganalisis gejala yang dilaporkan oleh pasien dan membandingkannya dengan pengetahuan yang ada untuk menghasilkan diagnosis yang akurat.

b. Pengambilan Keputusan Bisnis

Dalam konteks bisnis, sistem berbasis pengetahuan dapat membantu dalam pengambilan keputusan yang kompleks. Misalnya, sistem dapat menggunakan pengetahuan tentang strategi pemasaran, perilaku konsumen, dan kondisi pasar untuk memberikan rekomendasi dalam pengembangan produk baru atau perencanaan strategi pemasaran

c. Sistem Pakar

Sistem berbasis pengetahuan dapat digunakan untuk mengembangkan sistem pakar yang dapat memberikan pengetahuan dan solusi dalam bidang spesifik. Contohnya, sistem pakar di bidang hukum dapat memberikan nasihat hukum berdasarkan pengetahuan hukum yang ada.

d. Pengelolaan Pengetahuan

Sistem berbasis pengetahuan dapat membantu mengelola pengetahuan organisasi, seperti basis pengetahuan pelanggan, prosedur operasional, atau

kebijakan internal. Sistem dapat menyimpan, mengatur, dan membagikan pengetahuan tersebut kepada karyawan, sehingga memfasilitasi kolaborasi dan pengambilan keputusan yang lebih baik.

e. Analisis Data dan Prediksi

Dalam bidang analisis data, sistem berbasis pengetahuan dapat digunakan untuk mengidentifikasi pola atau tren dalam data yang kompleks. Sistem dapat menggunakan pengetahuan tentang relasi antar variabel untuk melakukan analisis prediktif atau pengambilan keputusan berdasarkan data historis.

f. Manajemen Rantai Pasokan

Sistem berbasis pengetahuan dapat membantu dalam perencanaan, pengelolaan, dan pengoptimasian rantai pasokan. Sistem dapat menggunakan pengetahuan tentang permintaan pelanggan, ketersediaan stok, dan faktor-faktor lainnya untuk mengoptimalkan pengiriman dan manajemen persediaan.

g. Sistem Pendukung Keputusan Klinis

Dalam bidang kesehatan, sistem berbasis pengetahuan dapat membantu dokter dalam pengambilan keputusan klinis. Sistem dapat memanfaatkan pengetahuan medis terkini untuk memberikan rekomendasi tentang diagnosa, terapi, atau tindakan medis yang paling tepat.

Contoh tersebut merupakan masalah yang dapat diselesaikan dengan sistem berbasis pengetahuan. Penerapan sistem berbasis pengetahuan dapat sangat bervariasi tergantung pada domain aplikasinya dan masalah yang ingin diselesaikan.

2. Penentuan tujuan dan kebutuhan sistem

Penentuan tujuan dan kebutuhan sistem merupakan langkah awal yang krusial dalam pengembangan sistem berbasis pengetahuan. Tujuan dan kebutuhan yang jelas akan membimbing seluruh proses pengembangan sistem untuk mencapai hasil yang diinginkan. Berikut adalah langkah-langkah untuk menentukan tujuan dan kebutuhan sistem:

a. Identifikasi Masalah atau Tantangan

Identifikasi masalah atau tantangan yang ingin diselesaikan oleh sistem. Analisis masalah secara menyeluruh dan pahami bagaimana sistem berbasis pengetahuan dapat membantu dalam mengatasi masalah tersebut.

b. Libatkan Pihak Terkait

Libatkan semua pemangku kepentingan yang relevan, seperti pengguna, manajemen, ahli domain, dan tim pengembang. Lakukan diskusi dan wawancara untuk memahami kebutuhan dan harapan mereka terhadap sistem.

c. Definisikan Tujuan Utama

Tentukan tujuan utama sistem berbasis pengetahuan, yang menggambarkan hasil yang ingin dicapai. Tujuan dapat berupa meningkatkan efisiensi, meningkatkan kualitas layanan, mengurangi kesalahan, atau meningkatkan pengambilan keputusan.

d. Identifikasi Kebutuhan Fungsional

Identifikasi kebutuhan fungsional, yaitu apa yang sistem harus lakukan atau fasilitasi. Misalnya, sistem dapat membutuhkan kemampuan untuk menerima input pengetahuan, melakukan penalaran, menghasilkan rekomendasi, atau menyimpan dan mengelola pengetahuan.

e. Identifikasi Kebutuhan Non-Fungsional

Identifikasi kebutuhan non-fungsional, yang berkaitan dengan atribut atau karakteristik sistem, seperti keamanan, skalabilitas, performa, atau antarmuka pengguna yang intuitif. Pastikan kebutuhan ini didefinisikan dengan jelas untuk memastikan sistem dapat memenuhi standar kualitas yang diharapkan.

f. Prioritaskan Kebutuhan

Prioritaskan kebutuhan berdasarkan urgensi dan dampaknya terhadap keberhasilan sistem. Identifikasi kebutuhan kritis yang harus dipenuhi di awal pengembangan, serta kebutuhan yang dapat ditangani di tahap selanjutnya.

g. Validasi Kebutuhan

Diskusikan dan validasi kebutuhan dengan pemangku kepentingan terkait. Pastikan kebutuhan yang telah ditentukan mencerminkan harapan dan kebutuhan pengguna serta memperoleh persetujuan dari semua pihak terkait.

Proses penentuan tujuan dan kebutuhan sistem merupakan langkah penting untuk memastikan kesesuaian dan kesuksesan pengembangan sistem berbasis pengetahuan. Keterlibatan pihak terkait dan pemahaman yang mendalam tentang masalah yang ingin diselesaikan akan membantu merumuskan tujuan yang tepat dan kebutuhan yang harus dipenuhi oleh sistem.

3. Analisis pengetahuan yang diperlukan

Analisis pengetahuan yang diperlukan merupakan langkah penting dalam pengembangan sistem berbasis pengetahuan. Melalui analisis ini, pengetahuan yang relevan untuk sistem dapat diidentifikasi, diekstraksi, dan

dikelompokkan. Berikut adalah langkah-langkah dalam analisis pengetahuan yang diperlukan:

a. Identifikasi Sumber Pengetahuan

- 1) Tentukan sumber-sumber pengetahuan yang relevan dalam domain aplikasi.
- 2) Sumber-sumber pengetahuan dapat berupa dokumen, buku, jurnal, basis pengetahuan yang ada, atau ahli domain yang dapat memberikan pengetahuan tacit.

b. Pemetaan Domain

- 1) Pahami domain aplikasi secara menyeluruh.
- 2) Identifikasi konsep, istilah, dan hubungan antar konsep dalam domain.
- 3) Buat pemetaan domain untuk memvisualisasikan elemen-elemen utama dalam domain dan hubungan antar mereka.

c. Identifikasi Jenis-Jenis Pengetahuan

- 1) Identifikasi jenis-jenis pengetahuan yang relevan dalam domain.
- 2) Misalnya, pengetahuan dapat berupa fakta, aturan, prosedur, heuristik, atau pengetahuan berbasis kasus.

d. Analisis Dokumen

- 1) Lakukan analisis mendalam terhadap dokumen atau sumber-sumber pengetahuan yang telah diidentifikasi.
- 2) Pahami isi dokumen, identifikasi fakta-fakta penting, aturan, dan prosedur yang terkandung di dalamnya.

e. Wawancara dengan Ahli Domain

- 1) Wawancarai ahli domain untuk mengumpulkan pengetahuan tacit yang tidak terdokumentasi.

- 2) Ajukan pertanyaan terperinci dan berfokus pada pengetahuan yang relevan untuk sistem berbasis pengetahuan yang sedang dikembangkan.
- f. Pemodelan Pengetahuan
- 1) Gunakan hasil analisis untuk memodelkan pengetahuan dalam sistem berbasis pengetahuan.
 - 2) Buat struktur pengetahuan yang sesuai, seperti aturan, jaringan semantik, atau ontologi, untuk merepresentasikan pengetahuan dengan cara yang dapat dimanfaatkan oleh sistem.
- g. Verifikasi dan Validasi
- 1) Verifikasi dan validasi pengetahuan yang telah dikumpulkan dan direpresentasikan dalam sistem.
 - 2) Melibatkan ahli domain dan pemangku kepentingan terkait dalam memverifikasi keakuratan dan konsistensi pengetahuan yang terkumpul.

Analisis pengetahuan yang diperlukan membantu dalam memahami dan merepresentasikan pengetahuan yang relevan dalam sistem berbasis pengetahuan. Melalui langkah-langkah ini, pengetahuan dapat diekstraksi dari berbagai sumber, baik yang terdokumentasi maupun tacit, dan direpresentasikan dalam bentuk yang dapat dimanfaatkan oleh sistem untuk penalaran dan pengambilan keputusan.

10.3.2 Tahap Perancangan

Tahap perancangan merupakan salah satu tahap penting dalam pengembangan sistem berbasis pengetahuan. Pada tahap ini, desain sistem yang akan dibangun dibuat berdasarkan analisis kebutuhan dan pengetahuan yang telah diidentifikasi sebelumnya. Serta, desain sistem dan pengetahuan yang telah diidentifikasi dikonseptualisasikan

menjadi bentuk yang akan diimplementasikan. Rancangan ini akan menjadi panduan dalam tahap implementasi dan pengembangan lebih lanjut dari sistem berbasis pengetahuan tersebut.

1. Perancangan Struktur Pengetahuan.

Perancangan struktur pengetahuan merupakan langkah penting dalam pengembangan sistem berbasis pengetahuan. Struktur pengetahuan menggambarkan cara pengetahuan disusun, terorganisir, dan direpresentasikan dalam sistem. Berikut adalah beberapa pendekatan yang umum digunakan dalam perancangan struktur pengetahuan:

- a. Aturan (*Rule-based*)

- 1) Pendekatan ini menggunakan aturan sebagai bentuk utama pengetahuan dalam sistem.
 - 2) Aturan terdiri dari kondisi (if) dan tindakan (then) yang menyatakan hubungan antara fakta dan tindakan yang harus diambil.
 - 3) Struktur pengetahuan dalam bentuk aturan memungkinkan sistem untuk melakukan penalaran berbasis logika dan menghasilkan rekomendasi atau keputusan.

- b. Jaringan Semantik (*Semantic Networks*)

- 1) Jaringan semantik menggunakan graf untuk menggambarkan hubungan antara konsep-konsep dalam domain.
 - 2) Konsep-konsep direpresentasikan sebagai simpul dalam graf, sedangkan hubungan antara konsep-konsep direpresentasikan sebagai busur atau edge.
 - 3) Struktur pengetahuan dalam jaringan semantik memudahkan sistem dalam menghubungkan konsep-konsep terkait dan menemukan jalur pemikiran yang relevan.

c. Ontologi

- 1) Ontologi adalah representasi formal dari pengetahuan dalam suatu domain, yang mencakup konsep, relasi, dan properti yang terkait.
- 2) Ontologi memberikan struktur dan klasifikasi yang lebih rinci pada pengetahuan dalam domain, memungkinkan sistem untuk melakukan penalaran berbasis semantik yang lebih kompleks.
- 3) Ontologi sering digunakan dalam aplikasi yang kompleks dan memerlukan representasi pengetahuan yang lebih formal dan terstruktur.

d. Pengetahuan Berbasis Kasus (*Case-based*)

- 1) Pendekatan ini menggunakan pengetahuan yang diperoleh dari kasus-kasus sebelumnya sebagai basis pengetahuan.
- 2) Setiap kasus direpresentasikan sebagai himpunan fakta, aturan, atau tindakan yang relevan dengan masalah yang serupa.
- 3) Struktur pengetahuan dalam pengetahuan berbasis kasus memungkinkan sistem untuk mencocokkan masalah yang ada dengan kasus yang relevan dan mengadopsi solusi yang telah diterapkan sebelumnya.

Pendekatan-pendekatan di atas dapat digunakan secara individu atau dikombinasikan tergantung pada kebutuhan dan karakteristik aplikasi yang sedang dikembangkan. Penting untuk merancang struktur pengetahuan yang sesuai dengan domain aplikasi dan tujuan sistem berbasis pengetahuan tersebut.

2. Perancangan Basis Pengetahuan

Perancangan basis pengetahuan merupakan langkah penting dalam pengembangan sistem berbasis

pengetahuan. Basis pengetahuan merupakan tempat penyimpanan pengetahuan yang akan digunakan oleh sistem untuk penalaran dan pengambilan keputusan. Berikut adalah beberapa hal yang perlu dipertimbangkan dalam perancangan basis pengetahuan:

a. Identifikasi Jenis Pengetahuan

Identifikasi jenis-jenis pengetahuan yang akan disimpan dalam basis pengetahuan, seperti fakta, aturan, prosedur, heuristik, atau pengetahuan berbasis kasus. Jenis pengetahuan yang dipilih akan mempengaruhi cara penyimpanan dan penggunaan pengetahuan dalam sistem.

b. Struktur Penyimpanan

Pilih struktur penyimpanan yang sesuai untuk basis pengetahuan, tergantung pada jenis pengetahuan yang akan disimpan. Contoh struktur penyimpanan meliputi tabel, graf, pohon, atau format file khusus.

c. Organisasi Pengetahuan

Organisasikan pengetahuan dalam basis pengetahuan agar mudah diakses dan dimanfaatkan oleh sistem. Buat kategori atau kelompok berdasarkan topik atau tema tertentu untuk mengelompokkan pengetahuan yang serupa.

d. Representasi Pengetahuan

Tentukan cara representasi pengetahuan dalam basis pengetahuan. Misalnya, jika menggunakan aturan, tentukan format aturan yang digunakan, seperti format IF-THEN atau format lainnya. Jika menggunakan jaringan semantik, tentukan bagaimana konsep-konsep dan hubungan antara mereka akan direpresentasikan dalam basis pengetahuan.

e. Integrasi dengan Sistem Lain (Opsional)

Jika diperlukan, perancangan basis pengetahuan dapat melibatkan integrasi dengan sistem lain yang ada. Pertimbangkan cara basis pengetahuan akan berintegrasi dengan sistem manajemen basis data atau sistem lain yang digunakan dalam aplikasi.

f. Manajemen Basis Pengetahuan

Pertimbangkan bagaimana basis pengetahuan akan dikelola, termasuk proses pemeliharaan, pembaruan, dan penghapusan pengetahuan. Pastikan ada mekanisme untuk mengelola versi pengetahuan dan melacak perubahan yang dilakukan pada basis pengetahuan.

g. Validasi dan Verifikasi

Validasi dan verifikasi basis pengetahuan untuk memastikan keakuratan, konsistensi, dan relevansi pengetahuan yang disimpan. Melibatkan ahli domain dan pemangku kepentingan terkait dalam proses validasi dan verifikasi.

Perancangan basis pengetahuan memastikan bahwa pengetahuan yang diperlukan tersimpan dengan baik dan dapat diakses oleh sistem secara efisien. Struktur penyimpanan, organisasi, representasi, dan manajemen basis pengetahuan yang baik akan memudahkan sistem dalam menggunakan pengetahuan untuk penalaran dan pengambilan keputusan.

3. Perancangan aturan dan mekanisme inferensi

Perancangan aturan dan mekanisme inferensi merupakan langkah penting dalam pengembangan sistem berbasis pengetahuan. Aturan-aturan yang telah dirancang akan digunakan oleh mekanisme inferensi untuk melakukan penalaran dan pengambilan keputusan berdasarkan pengetahuan yang ada. Berikut adalah

beberapa hal yang perlu dipertimbangkan dalam perancangan aturan dan mekanisme inferensi:

a. Identifikasi Kondisi (*If*)

- 1) Identifikasi kondisi-kondisi yang harus terpenuhi agar aturan dapat diterapkan.
- 2) Kondisi-kondisi ini dapat berupa fakta-fakta atau pernyataan-pernyataan yang ada dalam basis pengetahuan.

b. Tentukan Tindakan (*Then*)

- 1) Tentukan tindakan atau langkah-langkah yang harus diambil jika kondisi-kondisi tertentu terpenuhi.
- 2) Tindakan ini bisa berupa pengambilan keputusan, menjalankan prosedur tertentu, atau menghasilkan rekomendasi.

c. Bentuk Aturan

- 1) Pilih format atau notasi yang sesuai untuk merancang aturan, seperti format IF-THEN atau format lainnya.
- 2) Pastikan aturan dirancang dengan jelas dan dapat dipahami oleh sistem dan pengguna.

d. Mekanisme Inferensi

- 1) Pilih jenis mekanisme inferensi yang sesuai untuk sistem, seperti forward chaining, backward chaining, atau kombinasi keduanya.
- 2) *Forward chaining* digunakan untuk menemukan tindakan-tindakan yang dapat diambil berdasarkan fakta-fakta yang ada.
- 3) *Backward chaining* digunakan untuk menemukan fakta-fakta yang perlu dipenuhi agar tindakan tertentu dapat diambil.

- e. Penanganan Konflik dan Prioritas
 - 1) Tentukan cara menangani konflik jika terdapat beberapa aturan yang bisa diterapkan pada satu waktu.
 - 2) Berikan prioritas pada aturan-aturan yang lebih penting atau lebih spesifik jika ada konflik.
- f. Kesimpulan dan Justifikasi
 - 1) Pertimbangkan bagaimana sistem dapat menghasilkan kesimpulan dan justifikasi yang dapat dipahami oleh pengguna.
 - 2) Sistem dapat menjelaskan bagaimana suatu keputusan atau rekomendasi diperoleh berdasarkan aturan-aturan yang diterapkan.
- g. Evaluasi dan Pembaruan
 - 1) Evaluasi dan uji mekanisme inferensi untuk memastikan korektness dan efisiensi.
 - 2) Jika diperlukan, lakukan pembaruan atau penyesuaian aturan dan mekanisme inferensi berdasarkan hasil evaluasi.

Perancangan aturan dan mekanisme inferensi yang baik akan memastikan sistem dapat melakukan penalaran dan pengambilan keputusan yang akurat dan efektif berdasarkan pengetahuan yang ada. Aturan-aturan yang dirancang harus memperhitungkan kondisi-kondisi yang relevan dan tindakan yang tepat untuk mencapai tujuan sistem. Mekanisme inferensi yang dipilih harus sesuai dengan karakteristik sistem dan dapat menghasilkan hasil yang diharapkan.

10.3.3 Tahap Implementasi

Tahap implementasi adalah langkah penting dalam pengembangan sistem berbasis pengetahuan. Pada tahap ini, desain dan spesifikasi yang telah dibuat dalam tahap

sebelumnya diubah menjadi bentuk yang dapat dijalankan oleh komputer. Berikut adalah langkah-langkah yang biasanya dilakukan dalam tahap implementasi:

1. Implementasi Aturan dan Pengetahuan
 - a. Mengimplementasikan aturan-aturan dan pengetahuan ke dalam bahasa pemrograman atau alat pengembangan yang dipilih.
 - b. Menyusun aturan-aturan dan pengetahuan sesuai dengan struktur dan format yang telah dirancang sebelumnya.
2. Pengembangan Mekanisme Inferensi
 - a. Membangun mekanisme inferensi yang akan digunakan oleh sistem untuk melakukan penalaran dan pengambilan keputusan.
 - b. Mengimplementasikan algoritma dan metode yang diperlukan untuk menerapkan aturan-aturan dan melakukan inferensi.
3. Integrasi dengan Komponen Sistem Lain (Opsional)
 - a. Jika sistem berbasis pengetahuan akan diintegrasikan dengan komponen sistem lain, seperti antarmuka pengguna atau sistem manajemen basis data, maka tahap ini melibatkan integrasi tersebut.
 - b. Menghubungkan sistem berbasis pengetahuan dengan komponen lain dan memastikan interaksi yang mulus antara mereka.
4. Pengujian dan Validasi
 - a. Melakukan pengujian dan validasi terhadap sistem berbasis pengetahuan yang telah diimplementasikan.
 - b. Memastikan bahwa sistem berperilaku sesuai dengan harapan dan dapat menghasilkan hasil yang akurat dan konsisten.

5. Pemeliharaan dan Peningkatan

- a. Setelah implementasi, sistem berbasis pengetahuan akan memasuki tahap pemeliharaan dan peningkatan.
- b. Mengidentifikasi dan memperbaiki bug atau kesalahan dalam implementasi.
- c. Menangani perubahan atau penambahan pengetahuan baru yang diperlukan seiring waktu.

Pada tahap implementasi, penting untuk memastikan bahwa implementasi sesuai dengan desain yang telah dirancang dan spesifikasi yang telah ditetapkan. Pengujian dan validasi sistem berbasis pengetahuan juga sangat penting untuk memastikan kinerjanya yang baik dan keandalannya. Selain itu, pemeliharaan dan peningkatan berkelanjutan perlu dilakukan untuk menjaga sistem berbasis pengetahuan tetap relevan dan efektif seiring berjalannya waktu.

1. Pemilihan teknologi dan bahasa pemrograman

Pemilihan teknologi dan bahasa pemrograman dalam implementasi sistem berbasis pengetahuan sangat penting untuk keberhasilan proyek. Berikut adalah beberapa faktor yang perlu dipertimbangkan dalam pemilihan teknologi dan bahasa pemrograman:

a. Kebutuhan dan Fitur Sistem

Pertimbangkan fitur-fitur yang diperlukan oleh sistem berbasis pengetahuan. Periksa apakah teknologi dan bahasa pemrograman yang dipertimbangkan dapat mendukung kebutuhan tersebut, seperti kemampuan penalaran logika, dukungan terhadap aturan dan inferensi, dan pemrosesan pengetahuan.

b. Ketersediaan dan Keahlian

Pertimbangkan ketersediaan dan keahlian tim pengembang terkait dengan teknologi dan bahasa pemrograman yang akan digunakan. Jika tim

pengembang memiliki pengalaman dan keahlian dalam teknologi atau bahasa tertentu, itu dapat menjadi pertimbangan penting.

c. Ketersediaan Sumber Daya dan Dukungan

Periksa ketersediaan sumber daya dan dukungan untuk teknologi atau bahasa pemrograman yang akan digunakan. Pertimbangkan apakah ada dokumentasi, tutorial, forum diskusi, atau komunitas yang dapat membantu dalam pengembangan sistem berbasis pengetahuan.

d. Integrasi dengan Komponen Lain

Jika sistem berbasis pengetahuan perlu diintegrasikan dengan komponen lain, seperti basis data atau antarmuka pengguna, pastikan bahwa teknologi dan bahasa pemrograman yang dipilih dapat berintegrasi dengan baik dengan komponen tersebut.

e. Kinerja dan Skalabilitas

Pertimbangkan kinerja sistem berbasis pengetahuan dan skalabilitas yang diinginkan. Periksa apakah teknologi dan bahasa pemrograman yang dipilih dapat memberikan kinerja yang memadai dan mampu menangani jumlah pengetahuan yang besar jika diperlukan.

f. Keberlanjutan dan Ketersediaan Tenaga Kerja

Pertimbangkan keberlanjutan teknologi atau bahasa pemrograman yang dipilih. Periksa apakah teknologi atau bahasa pemrograman tersebut masih aktif dan akan terus didukung ke depannya. Pastikan juga ada ketersediaan tenaga kerja yang memiliki keahlian dalam teknologi atau bahasa pemrograman tersebut.

Setiap proyek dapat memiliki kebutuhan dan konteks yang berbeda, sehingga pemilihan teknologi dan bahasa pemrograman harus disesuaikan dengan kebutuhan

spesifik proyek tersebut. Penting untuk melakukan riset dan evaluasi yang teliti sebelum membuat keputusan final mengenai pemilihan teknologi dan bahasa pemrograman.

2. Pembangunan Basis Pengetahuan

Pembangunan basis pengetahuan adalah tahap dalam pengembangan sistem berbasis pengetahuan di mana pengetahuan yang relevan dikumpulkan, dianalisis, dan diorganisir untuk membentuk basis pengetahuan yang akan digunakan oleh sistem. Berikut adalah langkah-langkah umum dalam pembangunan basis pengetahuan:

a. Identifikasi Sumber Pengetahuan

- 1) Identifikasi sumber-sumber pengetahuan yang relevan yang akan digunakan dalam sistem.
- 2) Sumber pengetahuan dapat berasal dari ahli domain, dokumen referensi, literatur, basis data, atau pengalaman praktis.

b. Kumpulkan Pengetahuan

- 1) Kumpulkan pengetahuan dari sumber-sumber yang telah diidentifikasi.
- 2) Proses pengumpulan pengetahuan dapat melibatkan wawancara dengan ahli, studi literatur, observasi, atau penelusuran basis data.

c. Analisis dan Seleksi Pengetahuan

- 1) Analisis pengetahuan yang telah dikumpulkan untuk menentukan relevansi, keakuratan, dan kualitasnya.
- 2) Pilih pengetahuan yang paling relevan dan berharga untuk dimasukkan ke dalam basis pengetahuan.

d. Organisasi Pengetahuan

- 1) Organisasikan pengetahuan yang telah dipilih ke dalam struktur yang teratur dan sistematis.

- 2) Gunakan format atau notasi yang sesuai untuk mewakili pengetahuan, seperti aturan berbasis teks, grafik, atau representasi formal lainnya.
- e. Validasi dan Verifikasi
 - 1) Lakukan validasi dan verifikasi terhadap pengetahuan yang telah dikumpulkan dan diorganisir.
 - 2) Pastikan bahwa pengetahuan yang ada dalam basis pengetahuan akurat, konsisten, dan sesuai dengan kebutuhan sistem.
- f. Dokumentasi
 - 1) Dokumentasikan basis pengetahuan dengan jelas dan komprehensif.
 - 2) Buat dokumentasi yang mencakup struktur basis pengetahuan, aturan-aturan, fakta-fakta, dan justifikasi di balik pengetahuan tersebut.
- g. Pemeliharaan dan Pembaruan
 - 1) Selama penggunaan sistem, basis pengetahuan harus tetap diperbarui dan dipelihara.
 - 2) Tinjau dan perbarui basis pengetahuan secara berkala untuk mencerminkan perubahan dalam domain atau pengetahuan yang ada.

Pembangunan basis pengetahuan merupakan proses yang berkelanjutan dan iteratif. Pada tahap ini, kerja sama antara ahli domain, analis sistem, dan pengembang sangat penting untuk memastikan pengetahuan yang tepat dan relevan dimasukkan ke dalam basis pengetahuan.

3. Tahap Pengujian dan validasi sistem

Tahap pengujian dan validasi sistem adalah tahap penting dalam pengembangan sistem berbasis pengetahuan. Pada tahap ini, sistem yang telah diimplementasikan dan basis pengetahuannya diuji untuk

memastikan bahwa sistem berperilaku sebagaimana yang diharapkan dan menghasilkan hasil yang akurat. Berikut adalah beberapa langkah yang biasanya dilakukan dalam tahap pengujian dan validasi:

a. Perencanaan Pengujian

- 1) Rencanakan strategi pengujian yang mencakup tujuan pengujian, lingkup pengujian, jenis pengujian, data pengujian, dan sumber daya yang diperlukan.
- 2) Identifikasi skenario pengujian yang mencakup berbagai situasi dan kasus uji untuk menguji fungsionalitas dan performa sistem.

b. Pengujian Fungsional

- 1) Lakukan pengujian untuk memastikan bahwa sistem berbasis pengetahuan berperilaku sesuai dengan aturan dan pengetahuan yang telah ditentukan.
- 2) Verifikasi bahwa aturan dan inferensi berfungsi dengan benar dan menghasilkan hasil yang diharapkan.

c. Pengujian Performa

- 1) Uji kinerja sistem untuk memastikan bahwa sistem mampu menangani volume pengetahuan yang besar dan memberikan respons yang cepat.
- 2) Lakukan pengujian beban (*load testing*) dan pengujian stres (*stress testing*) untuk menguji kinerja sistem dalam situasi yang menguji batasannya.

d. Pengujian Kesalahan dan Pengecualian

- 1) Identifikasi kemungkinan kesalahan atau pengecualian yang dapat terjadi dalam sistem.
- 2) Uji sistem dengan skenario pengujian yang mencakup situasi yang memicu kesalahan atau

- pengecualian dan pastikan sistem menangani mereka dengan benar.
- e. Validasi dengan Ahli Domain
 - 1) Melibatkan ahli domain untuk memvalidasi basis pengetahuan dan keputusan yang dihasilkan oleh sistem.
 - 2) Diskusikan hasil pengujian dengan ahli domain dan pastikan bahwa sistem memberikan hasil yang tepat dan relevan.
 - f. Evaluasi Kualitas
 - 1) Evaluasi kualitas sistem berdasarkan kriteria yang telah ditentukan, seperti akurasi, keandalan, efisiensi, dan kepuasan pengguna.
 - 2) Identifikasi dan perbaiki kekurangan yang ditemukan selama evaluasi.
 - g. Pengujian Integrasi
 - 1) Jika sistem berbasis pengetahuan diintegrasikan dengan komponen lain, lakukan pengujian integrasi untuk memastikan interaksi yang mulus antara sistem berbasis pengetahuan dan komponen lainnya.
 - h. Pembaruan dan Peningkatan
 - 1) Jika ada masalah atau kekurangan yang ditemukan selama pengujian, perbaiki dan tingkatkan sistem berdasarkan hasil pengujian tersebut.
 - 2) Pastikan bahwa sistem telah melewati pengujian dan validasi sebelum melanjutkan ke tahap produksi atau implementasi.

Tahap pengujian dan validasi yang komprehensif dan teliti akan membantu memastikan bahwa sistem berbasis pengetahuan bekerja dengan baik dan menghasilkan hasil yang diharapkan. Penting untuk mengikuti metode dan prosedur

yang teliti dalam melakukan pengujian dan validasi. Selain itu, kolaborasi yang baik antara tim pengembang, ahli domain, dan pengguna juga penting untuk memperoleh masukan dan umpan balik yang berharga selama proses pengujian.

Selama tahap pengujian dan validasi, dokumentasi yang baik juga harus diperhatikan. Catat hasil pengujian, kesalahan yang ditemukan, langkah-langkah pemecahan masalah, dan perbaikan yang telah dilakukan. Dokumentasi ini akan membantu tim pengembang dalam memahami dan melacak proses pengujian serta memfasilitasi pemeliharaan dan pembaruan sistem di masa mendatang.

Selain itu, penting juga untuk memperhatikan keamanan sistem berbasis pengetahuan selama tahap pengujian dan validasi. Lakukan pengujian keamanan untuk mengidentifikasi dan mengatasi kerentanan yang mungkin ada dalam sistem. Terakhir, lakukan pengujian dengan menggunakan data realistis dan skenario pengujian yang mencakup berbagai situasi yang mungkin terjadi dalam kehidupan nyata. Hal ini akan membantu memastikan bahwa sistem berbasis pengetahuan dapat berkinerja baik dalam berbagai kondisi dan memberikan hasil yang dapat diandalkan. Setelah tahap pengujian dan validasi selesai, evaluasi hasil pengujian dan umpan balik dari pengguna dan ahli domain. Gunakan informasi ini untuk melakukan perbaikan dan penyempurnaan terakhir sebelum melanjutkan ke tahap implementasi dan penggunaan sistem berbasis pengetahuan. Penting untuk diingat bahwa tahap pengujian dan validasi tidak hanya satu kali, tetapi bisa menjadi siklus iteratif. Setelah implementasi sistem berbasis pengetahuan, terus pantau kinerjanya, terima masukan dari pengguna, dan lakukan pembaruan dan perbaikan sesuai kebutuhan.

Dengan menjalankan tahap pengujian dan validasi dengan cermat, Anda dapat memastikan bahwa sistem berbasis

pengetahuan yang dikembangkan bekerja dengan baik, memberikan hasil yang akurat, dan membantu memecahkan masalah dalam domain yang relevan.

10.4 Aplikasi Sistem Berbasis Pengetahuan

10.4.1 Sistem Pakar

1. Definisi sistem pakar

Sistem pakar adalah jenis sistem berbasis pengetahuan yang menggunakan pengetahuan yang dikhususkan oleh para ahli dalam suatu domain tertentu untuk mengambil keputusan atau memberikan solusi atas masalah yang kompleks. Sistem pakar memanfaatkan pengetahuan yang telah diakuisisi dari para ahli manusia dan menyimpannya dalam bentuk aturan-aturan dan fakta-fakta yang dapat digunakan untuk melakukan penalaran dan inferensi.

Sistem pakar dirancang untuk meniru kemampuan para ahli manusia dalam menganalisis masalah, membuat keputusan, dan memberikan solusi. Mereka menggunakan pengetahuan yang terstruktur dan algoritma pemrosesan pengetahuan untuk menghasilkan jawaban yang berdasarkan pada situasi atau masalah yang diberikan. Keunggulan utama dari sistem pakar adalah kemampuannya dalam mengatasi masalah yang kompleks dan memerlukan pengetahuan yang mendalam dalam domain tertentu. Mereka dapat memberikan solusi yang konsisten, akurat, dan cepat, serta dapat mengatasi situasi yang tidak umum atau langka. Sistem pakar telah diterapkan dalam berbagai bidang, termasuk kedokteran, keuangan, manufaktur, hukum, dan banyak lagi. Mereka digunakan untuk membantu diagnosis penyakit, pengambilan keputusan investasi, perencanaan produksi, analisis hukum, dan berbagai tugas yang membutuhkan pemahaman mendalam dan keahlian domain khusus.

Dalam sistem pakar, pengetahuan yang dimiliki oleh ahli manusia dikodekan dalam bentuk aturan, basis pengetahuan, atau jaringan semantik, yang kemudian digunakan oleh mesin untuk melakukan inferensi dan memberikan solusi. Sistem pakar juga dapat melibatkan antarmuka pengguna yang memungkinkan pengguna berinteraksi dengan sistem dan memberikan masukan tambahan. Secara keseluruhan, sistem pakar merupakan aplikasi yang kuat dalam bidang kecerdasan buatan yang memanfaatkan pengetahuan manusia yang spesifik dan pengolahan pengetahuan yang canggih untuk menghadapi masalah yang kompleks dan memberikan solusi yang dapat diandalkan.

Selain itu, Sistem pakar dapat diartikan sebagai aplikasi yang memanfaatkan pengetahuan dan keahlian dari para ahli dalam suatu domain tertentu untuk menyelesaikan masalah yang kompleks. Mereka dirancang untuk meniru pemikiran dan penalaran manusia dalam mengambil keputusan. Dalam sistem pakar, pengetahuan yang dimiliki oleh para ahli manusia direpresentasikan secara formal dan digunakan untuk melakukan inferensi dan penalaran.

Berikut adalah beberapa komponen penting dalam sebuah sistem pakar:

a. Basis Pengetahuan (*Knowledge Base*)

Basis pengetahuan adalah komponen utama dalam sistem pakar. Ini berisi pengetahuan yang dikumpulkan dari para ahli manusia dan disusun dalam bentuk yang dapat digunakan oleh sistem. Basis pengetahuan terdiri dari aturan-aturan dan fakta-fakta yang menjelaskan hubungan antara data masukan dan solusi yang dihasilkan. Aturan-aturan ini ditulis dalam bahasa yang dapat dimengerti oleh sistem dan digunakan untuk melakukan penalaran.

b. Mesin Inferensi (*Inference Engine*)

Mesin inferensi adalah komponen yang bertanggung jawab untuk melakukan penalaran dan inferensi berdasarkan pada pengetahuan yang ada dalam basis pengetahuan. Mesin inferensi menerapkan aturan-aturan yang ada dan menggunakan data masukan untuk menghasilkan solusi atau kesimpulan. Berbagai teknik inferensi dapat digunakan, seperti *forward chaining* (penalaran maju) atau *backward chaining* (penalaran mundur), untuk mencapai tujuan yang diinginkan.

c. Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna menyediakan cara bagi pengguna untuk berinteraksi dengan sistem pakar. Ini dapat berupa antarmuka grafis yang intuitif atau antarmuka berbasis teks yang memungkinkan pengguna memasukkan data masukan, mendapatkan solusi, atau memberikan informasi tambahan kepada sistem. Antarmuka pengguna juga dapat digunakan untuk menjelaskan solusi yang dihasilkan oleh sistem dan memberikan justifikasi atau penjelasan tentang hasil tersebut.

d. Pengetahuan Ahli (*Expert Knowledge*)

Pengetahuan ahli adalah pengetahuan yang dimiliki oleh para ahli manusia dalam domain yang relevan. Ini mencakup pemahaman mendalam tentang masalah, pengalaman praktis, heuristik, dan pengetahuan berbasis aturan. Pengetahuan ahli digunakan sebagai landasan untuk mengembangkan basis pengetahuan dalam sistem pakar. Ahli domain bekerja sama dengan tim pengembang untuk mentransfer pengetahuan mereka ke dalam sistem dan memvalidasi kebenaran solusi yang dihasilkan.

Proses pengembangan sistem pakar melibatkan tahap analisis, perancangan, implementasi, pengujian, dan validasi. Pada tahap analisis, masalah yang akan diselesaikan dan pengetahuan yang dibutuhkan diidentifikasi. Kemudian, pada tahap perancangan, struktur pengetahuan dan aturan dibangun. Tahap implementasi melibatkan implementasi basis pengetahuan dan mesin inferensi ke dalam sistem komputer. Setelah itu, tahap pengujian dan validasi dilakukan untuk memastikan sistem memberikan solusi yang benar dan memenuhi kebutuhan pengguna. Pada tahap pengujian, sistem pakar diuji dengan skenario pengujian yang mencakup berbagai situasi dan kasus uji untuk memastikan kinerjanya yang baik. Selama tahap validasi, sistem dievaluasi dengan melibatkan ahli domain dan pengguna akhir untuk memastikan bahwa solusi yang dihasilkan sesuai dengan harapan dan memenuhi standar yang ditetapkan.

Selain komponen utama yang disebutkan di atas, terdapat juga beberapa aspek penting dalam perancangan dan implementasi sistem pakar, seperti:

a. Representasi Pengetahuan

Representasi pengetahuan adalah cara di mana pengetahuan ahli diwakili dalam sistem pakar. Ini dapat mencakup aturan produksi (*rule-based*), jaringan semantik (*semantic networks*), logika predikat (*predicate logic*), atau bentuk lainnya. Representasi pengetahuan dipilih berdasarkan kebutuhan domain dan tujuan sistem pakar.

b. Proses Pemerolehan Pengetahuan

Proses pemerolehan pengetahuan melibatkan pengumpulan, analisis, dan pengestrakan pengetahuan dari para ahli manusia. Proses ini mencakup wawancara dengan ahli, pengamatan, studi

literatur, dan penggunaan alat bantu seperti sistem pengumpulan pengetahuan. Tujuan dari proses ini adalah untuk mengidentifikasi pengetahuan yang relevan dan mengubahnya menjadi format yang dapat dimengerti oleh sistem pakar.

c. Manajemen Basis Pengetahuan

Manajemen basis pengetahuan melibatkan pengorganisasian, pemeliharaan, dan pembaruan basis pengetahuan dalam sistem pakar. Hal ini termasuk pengelolaan aturan-aturan, fakta-fakta, dan hubungan antara mereka. Manajemen yang baik dari basis pengetahuan memastikan bahwa sistem pakar tetap konsisten, terkini, dan dapat diandalkan dalam menghasilkan solusi.

d. Evaluasi dan Peningkatan

Setelah sistem pakar diimplementasikan, evaluasi dan pembaruan terus dilakukan untuk memastikan kinerjanya yang optimal. Evaluasi dilakukan melalui pengumpulan umpan balik dari pengguna dan ahli domain, serta dengan menggunakan metrik kualitas yang telah ditentukan sebelumnya. Berdasarkan hasil evaluasi, sistem dapat ditingkatkan dan diperbaiki melalui penambahan aturan baru, peningkatan performa, atau penyesuaian lainnya.

Dalam rangka mengembangkan sistem pakar yang efektif, penting untuk melibatkan ahli domain dan tim pengembang yang memiliki pemahaman mendalam tentang masalah dan kebutuhan pengguna. Kolaborasi yang baik antara ahli domain dan tim teknis akan memastikan bahwa sistem pakar dirancang dan dikembangkan dengan baik, serta memberikan solusi yang akurat dan relevan. Secara keseluruhan, sistem pakar adalah alat yang berharga dalam menghadapi masalah yang kompleks dan

memerlukan pengetahuan domain spesifik. Dengan menggunakan pengetahuan ahli manusia dan teknik inferensi yang canggih, sistem pakar mampu memberikan solusi yang akurat dan dapat mendukung pengambilan keputusan yang tepat. Tahap pengujian dan validasi yang cermat serta evaluasi yang berkelanjutan juga penting untuk memastikan kualitas dan kinerja sistem pakar. Selain itu, penting juga untuk mempertimbangkan aspek pengelolaan pengetahuan dalam perancangan sistem pakar. Ini mencakup pemeliharaan basis pengetahuan, manajemen pengetahuan yang diperoleh dari para ahli, dan pembaruan secara teratur untuk mengikuti perkembangan dalam domain yang relevan. Pengelolaan pengetahuan yang efektif akan memastikan bahwa sistem pakar tetap relevan dan dapat diandalkan dalam memberikan solusi.

Selama tahap implementasi, pemilihan teknologi dan bahasa pemrograman yang sesuai dengan kebutuhan sistem pakar juga penting. Berbagai teknologi dan bahasa pemrograman seperti Python, Java, atau Prolog dapat digunakan tergantung pada karakteristik dan kompleksitas sistem yang dibangun. Pemilihan teknologi yang tepat akan memudahkan pengembangan, pengujian, dan pemeliharaan sistem pakar. Tahap pengujian dan validasi merupakan langkah penting dalam memastikan kualitas dan kinerja sistem pakar. Pengujian harus mencakup skenario pengujian yang mencakup berbagai situasi dan kasus uji yang mewakili kondisi yang berbeda dalam domain yang relevan. Pengujian ini akan membantu mengidentifikasi kelemahan dan kesalahan dalam sistem pakar, serta memperbaikinya sebelum diterapkan secara penuh.

Validasi sistem pakar melibatkan evaluasi oleh ahli domain yang kompeten dalam memverifikasi keakuratan

solusi yang diberikan oleh sistem. Hal ini dapat dilakukan melalui perbandingan solusi sistem dengan solusi yang direkomendasikan oleh ahli manusia. Feedback dan masukan dari ahli domain sangat berharga dalam meningkatkan kualitas sistem pakar. Dalam tahap pengujian dan validasi, juga penting untuk memperhatikan aspek keamanan sistem pakar. Melindungi basis pengetahuan dan data pengguna dari akses yang tidak sah atau manipulasi adalah prioritas utama. Keamanan sistem pakar harus diperhatikan secara menyeluruh untuk menjaga kepercayaan dan integritas sistem.

Terakhir, dokumentasi sistem pakar juga merupakan komponen penting dalam tahap implementasi. Dokumentasi yang lengkap dan terstruktur akan membantu pengguna dan pengembang memahami sistem dengan baik. Ini mencakup dokumentasi tentang basis pengetahuan, aturan-aturan, proses inferensi, serta panduan penggunaan dan pemeliharaan sistem. Dalam kesimpulannya, tahap implementasi sistem pakar melibatkan pemilihan teknologi yang tepat, pengujian dan validasi yang cermat, serta perhatian terhadap pengelolaan pengetahuan dan keamanan sistem. Dengan pendekatan yang hati-hati dan metodologi yang baik, sistem pakar dapat diimplementasikan dengan sukses dan memberikan manfaat yang signifikan dalam menyelesaikan masalah kompleks dalam domain yang spesifik.

2. Contoh aplikasi sistem pakar

Berikut adalah beberapa contoh aplikasi sistem pakar dalam berbagai domain:

a. Diagnosa Medis

Sistem pakar dapat digunakan dalam bidang medis untuk membantu dalam diagnosis penyakit. Sistem

pakar dapat mengumpulkan informasi tentang gejala yang dialami oleh pasien, melakukan penalaran berdasarkan basis pengetahuan medis, dan memberikan rekomendasi diagnosis yang akurat. Contoh aplikasi yang terkenal adalah MYCIN, sebuah sistem pakar yang dikembangkan untuk diagnosis penyakit infeksi.

b. Perencanaan Perjalanan

Sistem pakar dapat digunakan dalam perencanaan perjalanan untuk memberikan rekomendasi tentang tempat wisata, akomodasi, dan rute perjalanan. Sistem pakar dapat mempertimbangkan preferensi pengguna, anggaran, dan informasi tentang tujuan wisata. Contoh aplikasi yang populer adalah TripAdvisor, yang menggunakan sistem pakar untuk memberikan rekomendasi perjalanan berdasarkan preferensi pengguna.

c. Perencanaan Keuangan

Sistem pakar dapat digunakan dalam perencanaan keuangan untuk membantu pengguna dalam mengelola investasi, perencanaan pensiun, dan pengelolaan keuangan pribadi. Sistem pakar dapat menganalisis profil risiko pengguna, mempertimbangkan kondisi pasar, dan memberikan saran tentang alokasi aset yang tepat. Contoh aplikasi yang terkenal adalah Wealthfront, sebuah platform investasi yang menggunakan sistem pakar untuk memberikan saran investasi yang disesuaikan dengan profil risiko pengguna.

d. Manajemen Rantai Pasokan

Sistem pakar dapat digunakan dalam manajemen rantai pasokan untuk membantu dalam pengambilan keputusan terkait persediaan, produksi, dan distribusi. Sistem pakar dapat menganalisis data dari berbagai sumber, mempertimbangkan faktor-faktor seperti

permintaan, kapasitas produksi, dan ketersediaan bahan baku, dan memberikan rekomendasi untuk mengoptimalkan operasi rantai pasokan. Contoh aplikasi yang umum adalah sistem pakar yang digunakan dalam manajemen persediaan dan perencanaan produksi.

e. Pengenalan Pola dan Pengenalan Suara

Sistem pakar dapat digunakan dalam pengenalan pola dan pengenalan suara untuk aplikasi seperti pengenalan wajah, pengenalan tulisan tangan, atau pengenalan suara. Sistem pakar dapat menggunakan pengetahuan yang dikumpulkan dari para ahli untuk mengenali pola dan membandingkannya dengan contoh yang ada dalam basis pengetahuan. Contoh aplikasi yang terkenal adalah sistem pengenalan suara Siri dari Apple atau sistem pengenalan wajah pada perangkat smartphone.

f. Perawatan Kesehatan

Sistem pakar dapat digunakan dalam perawatan kesehatan untuk memberikan saran tentang pengobatan, manajemen penyakit, dan perawatan pasien. Sistem pakar dapat menganalisis data medis, menggabungkannya dengan pengetahuan medis yang ada, dan memberikan rekomendasi yang personal. Contoh aplikasi termasuk sistem pakar yang digunakan dalam manajemen diabetes atau manajemen penyakit kronis lainnya.

g. Sistem Navigasi

Sistem pakar dapat digunakan dalam sistem navigasi untuk memberikan panduan rute, informasi lalu lintas, dan rekomendasi navigasi yang optimal. Sistem pakar dapat mempertimbangkan faktor-faktor seperti jarak, waktu tempuh, kondisi lalu lintas, dan

preferensi pengguna untuk memberikan petunjuk navigasi yang akurat. Contoh aplikasi yang umum adalah Google Maps, yang menggunakan sistem pakar untuk memberikan petunjuk rute yang optimal berdasarkan kondisi lalu lintas dan preferensi pengguna.

h. Perawatan Tanaman dan Pertanian

Sistem pakar dapat digunakan dalam bidang pertanian untuk memberikan saran tentang perawatan tanaman, manajemen hama, dan pengelolaan tanah. Sistem pakar dapat menganalisis data seperti jenis tanaman, kondisi cuaca, dan gejala penyakit, dan memberikan rekomendasi yang spesifik. Contoh aplikasi termasuk sistem pakar yang digunakan dalam manajemen kebun raya atau pertanian organik.

i. Manajemen Proyek

Sistem pakar dapat digunakan dalam manajemen proyek untuk membantu dalam perencanaan, pengawasan, dan pengambilan keputusan. Sistem pakar dapat mempertimbangkan faktor-faktor seperti sumber daya, jadwal, dan risiko proyek, dan memberikan rekomendasi untuk mengoptimalkan pelaksanaan proyek. Contoh aplikasi termasuk sistem pakar yang digunakan dalam manajemen konstruksi atau pengembangan perangkat lunak.

j. Diagnosa Kerusakan dan Perbaikan

Sistem pakar dapat digunakan dalam bidang perbaikan dan pemeliharaan untuk membantu dalam mendiagnosis kerusakan dan memberikan solusi perbaikan. Sistem pakar dapat mempertimbangkan gejala kerusakan, riwayat perbaikan, dan pengetahuan ahli untuk memberikan rekomendasi pemecahan masalah. Contoh aplikasi termasuk sistem pakar yang

digunakan dalam perbaikan perangkat elektronik atau kendaraan.

Dalam setiap contoh aplikasi sistem pakar, tujuan utamanya adalah membantu dalam pengambilan keputusan, mengatasi masalah kompleks, dan memberikan solusi yang akurat dan relevan dalam domain yang spesifik. Sistem pakar menjadi alat yang berharga dalam memanfaatkan pengetahuan ahli dan teknik inferensi untuk memberikan dukungan yang efektif dan efisien dalam berbagai bidang.

3. Proses Pengembangan Sistem Pakar

Proses pengembangan sistem pakar melibatkan serangkaian tahapan yang terstruktur dan metodologis. Proses pengembangan sistem pakar dapat berulang dan iteratif, di mana setelah implementasi awal, sistem pakar dapat diperbaiki atau ditingkatkan berdasarkan umpan balik dan pengalaman pengguna. Proses ini memastikan bahwa sistem pakar terus berkembang dan tetap relevan dalam mengatasi masalah dalam domain yang spesifik. Penting untuk mencatat bahwa setiap tahap dalam pengembangan sistem pakar harus dilakukan dengan cermat dan dengan melibatkan para ahli domain yang memiliki pengetahuan mendalam tentang masalah yang ingin diselesaikan.

Kolaborasi yang baik antara ahli domain dan tim pengembang sistem pakar akan memastikan bahwa sistem pakar dapat memberikan solusi yang efektif dan bernilai tambah dalam memecahkan masalah yang kompleks. Berikut adalah langkah-langkah umum yang terlibat dalam pengembangan sistem pakar:

a. Identifikasi Masalah

Tahap pertama dalam pengembangan sistem pakar adalah mengidentifikasi masalah atau domain spesifik yang akan diselesaikan oleh sistem pakar. Identifikasi masalah ini melibatkan pemahaman yang mendalam tentang masalah yang ingin dipecahkan dan batasan serta tujuan yang ingin dicapai dengan menggunakan sistem pakar.

b. Perencanaan

Tahap perencanaan melibatkan pengumpulan persyaratan sistem dan perencanaan strategi pengembangan. Hal ini melibatkan identifikasi pengguna target, analisis kebutuhan sistem, dan penentuan batasan dan skala proyek. Selain itu, tahap perencanaan juga melibatkan penentuan sumber daya yang diperlukan, jadwal pengembangan, dan alokasi anggaran.

c. Pengumpulan Pengetahuan

Tahap ini melibatkan pengumpulan pengetahuan dari para ahli domain. Pengetahuan ini dapat diperoleh melalui wawancara, diskusi, observasi, atau studi literatur. Tujuan dari tahap ini adalah memahami pengetahuan yang diperlukan untuk memecahkan masalah dalam domain yang spesifik.

d. Representasi Pengetahuan

Setelah pengetahuan terkumpul, langkah selanjutnya adalah merepresentasikan pengetahuan tersebut dalam bentuk yang dapat diproses oleh sistem pakar. Representasi pengetahuan dapat dilakukan menggunakan aturan-aturan (rule-based), jaringan semantik, atau bahasa formal lainnya, tergantung pada jenis pengetahuan yang diwakili.

e. Mekanisme Inferensi

Tahap ini melibatkan pengembangan mekanisme inferensi yang akan digunakan oleh sistem pakar untuk menarik kesimpulan dan memberikan solusi. Mekanisme inferensi ini dapat berupa aturan-aturan logika, pemrograman logika, atau teknik inferensi lainnya seperti jaringan semantik atau pendekatan berbasis kasus.

f. Pengembangan Antarmuka Pengguna

Tahap ini melibatkan pengembangan antarmuka pengguna yang interaktif dan mudah digunakan. Antarmuka pengguna akan memungkinkan pengguna berinteraksi dengan sistem pakar, memasukkan informasi, dan menerima solusi atau rekomendasi yang diberikan oleh sistem pakar. Antarmuka pengguna harus dirancang dengan memperhatikan kebutuhan pengguna dan kemudahan penggunaan.

g. Implementasi dan Integrasi

Tahap ini melibatkan implementasi sistem pakar berdasarkan desain yang telah dikembangkan. Implementasi melibatkan pengkodean aturan-aturan pengetahuan, mekanisme inferensi, dan antarmuka pengguna. Selain itu, sistem pakar juga perlu diintegrasikan dengan sistem yang ada jika diperlukan.

h. Pengujian dan Validasi

Tahap ini melibatkan pengujian sistem pakar untuk memastikan keakuratan, kinerja, dan fungsionalitas

sistem pakar harus diuji untuk memastikan bahwa ia berfungsi sesuai dengan yang diharapkan dan memberikan hasil yang akurat dan konsisten. Pengujian dilakukan dengan menggunakan kasus-kasus uji yang mencakup berbagai situasi dan skenario dalam domain yang relevan. Pengujian ini akan membantu

mengidentifikasi kesalahan atau kekurangan dalam sistem pakar dan memastikan bahwa solusi atau rekomendasi yang diberikan sesuai dengan pengetahuan domain.

Validasi sistem pakar juga penting untuk memastikan bahwa sistem pakar dapat diandalkan dan dapat diterima oleh pengguna. Validasi melibatkan evaluasi sistem pakar oleh para ahli domain atau pengguna yang memiliki pengetahuan dan pengalaman yang relevan. Feedback dari validasi ini dapat digunakan untuk meningkatkan kualitas dan kinerja sistem pakar sebelum diimplementasikan secara penuh.

i. Penerapan dan Pemeliharaan

Setelah melalui tahap pengujian dan validasi, sistem pakar siap untuk diimplementasikan secara penuh. Sistem pakar dapat diintegrasikan ke dalam lingkungan produksi atau digunakan sebagai alat bantu dalam proses pengambilan keputusan. Selama penerapan, pemeliharaan rutin juga penting untuk memastikan bahwa sistem pakar tetap berfungsi dengan baik dan diperbarui sesuai dengan perkembangan dalam domain atau perubahan pengetahuan yang mendasarinya.

j. Pelatihan Sistem Pakar

Setelah sistem pakar diimplementasikan, tahap selanjutnya adalah melatih sistem pakar. Pelatihan ini melibatkan memberikan pengetahuan domain kepada sistem pakar sehingga dapat membuat inferensi yang akurat dan memberikan solusi yang tepat. Pelatihan sistem pakar dapat dilakukan dengan mengumpulkan lebih banyak kasus yang relevan dan menggunakan metode pembelajaran mesin seperti pembelajaran berbasis aturan atau pembelajaran berbasis kasus.

k. Evaluasi dan Penilaian

Tahap evaluasi dan penilaian dilakukan untuk mengukur kinerja sistem pakar dan mengevaluasi apakah sistem pakar mencapai tujuan yang telah ditetapkan. Evaluasi dapat melibatkan pengukuran akurasi, kecepatan, efisiensi, dan kepuasan pengguna. Hasil evaluasi digunakan untuk menilai keberhasilan sistem pakar dan mengidentifikasi area di mana perbaikan atau peningkatan dapat dilakukan.

l. Dokumentasi

Dokumentasi yang komprehensif tentang sistem pakar harus disusun selama tahap pengembangan. Dokumentasi ini berisi penjelasan rinci tentang tujuan sistem pakar, pengetahuan yang diperoleh, representasi pengetahuan, aturan-aturan, mekanisme inferensi, dan catatan pengembangan lainnya. Dokumentasi ini berguna untuk pemeliharaan dan pengembangan masa depan serta untuk memfasilitasi pemahaman yang lebih baik tentang sistem pakar oleh para pengguna atau pengembang lainnya.

m. Penyempurnaan dan Pengembangan Lanjutan

Setelah sistem pakar diimplementasikan dan dievaluasi, tahap terakhir adalah melakukan penyempurnaan dan pengembangan lanjutan. Feedback dari pengguna dan pemangku kepentingan dapat digunakan untuk meningkatkan sistem pakar, mengidentifikasi kekurangan, dan menambahkan pengetahuan baru yang relevan. Pengembangan lanjutan dapat melibatkan perbaikan performa, penambahan fitur baru, atau pembaruan pengetahuan yang diperlukan.

Proses pengembangan sistem pakar adalah suatu rangkaian yang iteratif, di mana tahapan-tahapan tersebut

dapat diulang dan disesuaikan sesuai dengan kebutuhan dan kemajuan pengembangan. Selama seluruh proses ini, kolaborasi yang baik antara tim pengembang sistem pakar, ahli domain, dan pengguna sangat penting untuk mencapai hasil yang optimal.

10.4.2 Penunjang Keputusan

Penunjang keputusan adalah suatu konsep atau pendekatan yang digunakan untuk membantu pengambilan keputusan yang kompleks dan penting. Tujuan dari sistem penunjang keputusan adalah untuk menyediakan informasi yang relevan, analisis data, dan alat-alat yang diperlukan agar pengambil keputusan dapat membuat keputusan yang lebih baik.

Berikut adalah beberapa alat atau teknik yang digunakan dalam sistem penunjang keputusan:

1. **Sistem Informasi Manajemen (SIM)**

Sistem Informasi Manajemen adalah sistem yang dirancang untuk mengumpulkan, mengolah, menyimpan, dan mengelola informasi yang relevan dalam suatu organisasi. SIM menyediakan data dan laporan yang dibutuhkan oleh pengambil keputusan untuk memahami situasi, melacak kinerja, dan memprediksi tren di berbagai bidang operasional.

2. **Sistem Pendukung Keputusan (SPK)**

Sistem Pendukung Keputusan adalah sistem yang menggunakan data, model matematis, dan metode analisis untuk membantu pengambil keputusan dalam memecahkan masalah yang kompleks. SPK menyediakan alat untuk menganalisis alternatif, melakukan perhitungan, dan menyajikan hasil yang berguna bagi pengambil keputusan.

3. Analisis Data dan Data Mining

Analisis data dan data mining melibatkan penggunaan teknik dan algoritma untuk mengeksplorasi data yang ada, mengidentifikasi pola atau hubungan yang relevan, dan mengambil informasi yang berharga dari data tersebut. Hal ini dapat membantu pengambil keputusan dalam memahami tren, mengidentifikasi peluang, dan mengungkap wawasan yang mendukung pengambilan keputusan yang lebih baik.

4. Model Matematis

Model matematis adalah representasi formal dari situasi atau masalah yang kompleks dalam bentuk persamaan atau aturan matematika. Model matematis digunakan untuk menganalisis situasi, menguji skenario, dan memprediksi hasil dari berbagai keputusan yang mungkin diambil. Model ini dapat membantu pengambil keputusan dalam memahami implikasi dari keputusan yang berbeda dan melakukan analisis sensitivitas.

5. Sistem Berbasis Pengetahuan

Sistem berbasis pengetahuan menggabungkan pengetahuan eksplisit dan aturan-aturan yang diperoleh dari para ahli domain ke dalam sistem komputer. Sistem ini dapat memberikan rekomendasi atau solusi berdasarkan pengetahuan yang terkandung dalam sistem, sehingga membantu pengambil keputusan dalam memecahkan masalah kompleks yang memerlukan pengetahuan khusus.

6. Visualisasi Data

Visualisasi data adalah representasi grafis dari data yang kompleks untuk membantu pengambil keputusan dalam memahami pola, hubungan, dan tren yang ada dalam data. Visualisasi data dapat berupa grafik, diagram, peta, atau tampilan interaktif lainnya yang memungkinkan pengguna

melihat dan menganalisis data dengan cara yang lebih intuitif dan efektif.

Penggunaan alat-alat dan teknik penunjang keputusan dapat membantu pengambil keputusan dalam mengelola kompleksitas, mengurangi ketidakpastian, dan meningkatkan efisiensi dalam pengambilan keputusan.

Berikut adalah beberapa manfaat penggunaan penunjang keputusan:

1. Peningkatan Kecepatan dan Efisiensi: Sistem penunjang keputusan dapat mengotomatisasi proses pengumpulan data, analisis, dan penyajian informasi yang diperlukan untuk pengambilan keputusan. Hal ini memungkinkan pengambil keputusan untuk mengakses informasi secara cepat dan efisien, menghemat waktu dan sumber daya yang diperlukan untuk analisis manual.
2. Akurasi dan Konsistensi: Dengan menggunakan algoritma dan model matematis, sistem penunjang keputusan dapat menghasilkan analisis yang akurat dan konsisten. Ini membantu pengambil keputusan dalam menghindari kesalahan manusia dan membuat keputusan yang lebih tepat berdasarkan fakta dan data yang terukur.
3. Analisis Mendalam: Melalui teknik analisis data dan visualisasi, sistem penunjang keputusan memungkinkan pengambil keputusan untuk melihat gambaran yang lebih lengkap dan mendalam tentang situasi atau masalah yang dihadapi. Mereka dapat menjelajahi data, mengidentifikasi pola atau tren yang relevan, dan membuat inferensi yang mendukung pengambilan keputusan yang informasional.
4. Pengambilan Keputusan yang Terinformasi: Dengan menyediakan informasi yang relevan dan terkini, sistem penunjang keputusan membantu pengambil keputusan dalam mengurangi ketidakpastian dan risiko. Mereka dapat

mengakses data historis, memprediksi hasil berdasarkan skenario yang berbeda, dan mengambil keputusan yang lebih terinformasi dan berdasarkan bukti.

5. Pembagian Pengetahuan: Sistem penunjang keputusan memungkinkan pengambil keputusan untuk berbagi pengetahuan dan informasi dengan pengguna lainnya. Hal ini memungkinkan kolaborasi dan komunikasi yang lebih baik antara para pemangku kepentingan dalam pengambilan keputusan yang melibatkan banyak pihak.
6. Peningkatan Produktivitas dan Kinerja: Dengan mempercepat proses pengambilan keputusan, meningkatkan akurasi, dan memberikan akses ke informasi yang relevan, sistem penunjang keputusan dapat meningkatkan produktivitas dan kinerja organisasi secara keseluruhan. Ini membantu organisasi dalam mengambil keputusan yang lebih baik, mengoptimalkan penggunaan sumber daya, dan mencapai tujuan bisnis yang ditetapkan.

Penting untuk memahami bahwa sistem penunjang keputusan hanyalah alat atau pendekatan yang digunakan untuk membantu pengambil keputusan dalam proses pengambilan keputusan. Keputusan akhir tetap merupakan tanggung jawab dan kebijakan dari pengambil keputusan itu sendiri.

Adapun Tahap Pengembangan Sistem Penunjang Keputusan, sebagai berikut:

1. Identifikasi Kebutuhan Pengguna

Tahap pertama dalam pengembangan sistem penunjang keputusan adalah mengidentifikasi kebutuhan pengguna. Ini melibatkan pemahaman yang mendalam tentang masalah atau situasi yang akan diselesaikan, pemahaman tentang kebutuhan informasi pengguna, dan tujuan yang ingin dicapai melalui sistem penunjang keputusan.

2. Perancangan Sistem

Setelah kebutuhan pengguna teridentifikasi, langkah selanjutnya adalah merancang sistem penunjang keputusan. Ini melibatkan pemilihan alat dan teknologi yang sesuai, perancangan antarmuka pengguna, serta perancangan struktur dan komponen sistem penunjang keputusan.

3. Pengumpulan dan Integrasi Data

Untuk mendukung pengambilan keputusan yang informasional, data yang relevan harus dikumpulkan dan diintegrasikan dalam sistem penunjang keputusan. Ini melibatkan pengumpulan data dari berbagai sumber, pembersihan data, pengolahan data, dan pengintegrasian data ke dalam sistem.

4. Pengembangan Model dan Analisis

Tahap ini melibatkan pengembangan model matematis, teknik analisis data, dan algoritma yang diperlukan untuk menganalisis data dan memberikan informasi yang berguna bagi pengambil keputusan. Model dan analisis ini digunakan untuk menggali wawasan dari data, memprediksi hasil, dan mendukung proses pengambilan keputusan.

5. Implementasi dan Integrasi

Setelah sistem penunjang keputusan dikembangkan, langkah selanjutnya adalah mengimplementasikannya dalam lingkungan operasional organisasi. Ini melibatkan instalasi sistem, integrasi dengan sistem yang ada, serta pelatihan pengguna dalam penggunaan sistem penunjang keputusan.

6. Uji Coba dan Evaluasi

Sebelum sistem penunjang keputusan dapat digunakan secara penuh, perlu dilakukan uji coba dan evaluasi untuk memastikan kinerjanya sesuai dengan harapan. Uji coba ini melibatkan pengujian fungsionalitas, pengujian performa,

dan evaluasi oleh pengguna untuk mengidentifikasi masalah potensial dan memperbaikinya.

7. Pemeliharaan dan Peningkatan

Setelah sistem penunjang keputusan diimplementasikan, pemeliharaan dan peningkatan berkala diperlukan untuk memastikan kinerja sistem yang optimal. Ini melibatkan pemantauan sistem, penanganan masalah yang muncul, pembaruan data dan model, serta pengembangan lanjutan untuk memenuhi kebutuhan yang berkembang.

8. Pelatihan dan Dukungan Pengguna

Selama dan setelah implementasi, pelatihan dan dukungan pengguna penting untuk memastikan penggunaan yang efektif dan pemahaman yang tepat tentang sistem penunjang keputusan. Pelatihan diberikan kepada pengguna agar mereka dapat mengoperasikan sistem dengan baik, memahami output yang dihasilkan, dan menggunakan informasi yang diberikan untuk pengambilan keputusan yang lebih baik.

9. Evaluasi dan Peningkatan Berkelanjutan

Tahap terakhir dalam pengembangan sistem penunjang keputusan adalah evaluasi dan peningkatan berkelanjutan. Evaluasi dilakukan untuk mengevaluasi kinerja sistem penunjang keputusan terhadap tujuan yang ditetapkan dan kebutuhan pengguna. Evaluasi ini melibatkan pengumpulan umpan balik dari pengguna, analisis kinerja sistem, dan identifikasi area peningkatan yang mungkin.

Berdasarkan hasil evaluasi, langkah-langkah perbaikan dan peningkatan dapat diimplementasikan. Ini dapat meliputi peningkatan fungsionalitas sistem, pembaruan data yang digunakan, penyesuaian algoritma dan model, atau peningkatan antarmuka pengguna. Peningkatan berkelanjutan perlu dilakukan untuk memastikan sistem penunjang

keputusan tetap relevan dan efektif dalam mendukung pengambilan keputusan di masa depan.

Penting untuk mencatat bahwa pengembangan sistem penunjang keputusan adalah proses iteratif. Setelah tahap evaluasi dan peningkatan, siklus pengembangan dapat kembali dimulai dengan identifikasi kebutuhan baru atau perubahan lingkungan yang memerlukan pembaruan sistem penunjang keputusan.

Dalam keseluruhan proses pengembangan sistem penunjang keputusan, kolaborasi yang baik antara pengambil keputusan, analis, dan pengembang sistem sangat penting. Pengambil keputusan harus secara aktif terlibat dalam semua tahap pengembangan untuk memastikan bahwa sistem penunjang keputusan memenuhi kebutuhan dan tujuan mereka dengan tepat.

1. Pendekatan dalam sistem penunjang keputusan

Dalam membangun sistem penunjang keputusan, terdapat beberapa pendekatan yang dapat digunakan, di antaranya:

- a. Model pengambilan keputusan: Pendekatan ini mengacu pada model pengambilan keputusan secara umum, seperti model rasional, model pengambilan keputusan terbatas, atau model politik. Dalam pendekatan ini, sistem penunjang keputusan didesain untuk membantu pengambil keputusan dalam menerapkan model pengambilan keputusan yang sesuai.
- b. Sistem berbasis aturan: Pendekatan ini menggunakan aturan-aturan yang sudah tersusun sebelumnya untuk membuat keputusan. Aturan tersebut dibuat berdasarkan pengetahuan yang telah ada dan dapat diubah atau diperbaharui sesuai kebutuhan. Contohnya adalah sistem pakar.
- c. Sistem berbasis pengetahuan: Pendekatan ini menggunakan basis pengetahuan yang diperoleh dari

pakar atau sumber pengetahuan lainnya. Basis pengetahuan ini kemudian diolah dengan menggunakan teknik-teknik logika atau heuristik untuk membantu pengambil keputusan.

- d. Data mining: Pendekatan ini menggunakan teknik-teknik statistik dan matematika untuk mengekstraksi pola dari data yang tersedia, sehingga dapat digunakan untuk membuat keputusan.
- e. Sistem manajemen basis data: Pendekatan ini mengelola basis data dalam organisasi dan menyediakan informasi yang diperlukan untuk membuat keputusan. Sistem ini memudahkan akses data dan memberikan kemampuan untuk melihat data dalam berbagai perspektif.
- f. Sistem informasi geografis: Pendekatan ini menggunakan informasi geografis dalam pengambilan keputusan, misalnya dalam perencanaan tata kota, manajemen sumber daya alam, atau dalam analisis risiko bencana. Pendekatan-pendekatan tersebut dapat digunakan secara terpisah atau dikombinasikan dalam pengembangan sistem penunjang keputusan, tergantung pada kebutuhan dan karakteristik masalah yang dihadapi.
- g. Analisis multi-kriteria: Pendekatan ini digunakan ketika pengambil keputusan harus mempertimbangkan beberapa kriteria atau faktor dalam pengambilan keputusan. Metode seperti Analisis Hierarchy Process (AHP) atau Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) dapat digunakan untuk menggabungkan dan membandingkan kriteria yang berbeda dalam menghasilkan rekomendasi keputusan.
- h. Optimasi matematis: Pendekatan ini melibatkan penggunaan teknik-teknik optimasi matematis untuk

memecahkan masalah pengambilan keputusan yang melibatkan pembagian sumber daya yang terbatas atau penentuan solusi optimal dari berbagai alternatif yang mungkin. Metode linear programming, integer programming, atau dynamic programming sering digunakan dalam pendekatan ini.

- i. Jaringan syaraf tiruan: Pendekatan ini mengadopsi konsep dari struktur dan fungsi jaringan syaraf manusia dalam mengambil keputusan. Jaringan syaraf tiruan dapat digunakan untuk mengidentifikasi pola, mempelajari hubungan kompleks antara variabel, dan melakukan prediksi berdasarkan data yang ada.
- j. Metode simulasi: Pendekatan ini melibatkan penggunaan model matematis atau komputer untuk mensimulasikan situasi atau skenario tertentu dalam pengambilan keputusan. Dengan menggunakan simulasi, pengambil keputusan dapat menganalisis dampak keputusan pada sistem secara virtual sebelum mengambil keputusan yang sebenarnya.
- k. Teknik pengambilan keputusan berbasis web: Pendekatan ini memanfaatkan teknologi web dan internet untuk memberikan akses mudah dan cepat ke informasi dan alat pengambilan keputusan kepada pengguna. Sistem penunjang keputusan berbasis web memungkinkan pengguna untuk mengakses sistem dari mana saja dan kapan saja, serta memfasilitasi kolaborasi antara pengambil keputusan yang berbeda.

Pilihan pendekatan dalam pengembangan sistem penunjang keputusan akan sangat tergantung pada jenis masalah yang dihadapi, karakteristik data yang tersedia, serta kebutuhan dan preferensi pengguna. Penting untuk memilih pendekatan yang paling sesuai dan efektif untuk memastikan sistem penunjang keputusan dapat

memberikan manfaat yang optimal dalam mendukung pengambilan keputusan yang tepat.

2. Contoh Aplikasi Sistem Penunjang Keputusan

Berikut adalah beberapa contoh aplikasi sistem penunjang keputusan dalam berbagai bidang:

- a. Keuangan dan Investasi: Sistem penunjang keputusan dapat digunakan dalam analisis portofolio investasi, perencanaan keuangan, pengelolaan risiko, dan prediksi pergerakan pasar. Contohnya adalah sistem yang memberikan rekomendasi investasi berdasarkan analisis data historis dan prediksi pasar.
- b. Manufaktur: Dalam industri manufaktur, sistem penunjang keputusan dapat digunakan untuk perencanaan persediaan, perencanaan produksi, pengelolaan rantai pasok, dan pemeliharaan peralatan. Sistem tersebut dapat membantu dalam pengambilan keputusan yang efisien terkait pengaturan jadwal produksi, pengelolaan persediaan, dan pemeliharaan yang optimal.
- c. Logistik dan Transportasi: Sistem penunjang keputusan dapat membantu dalam pengaturan rute pengiriman, pemilihan metode transportasi yang efisien, manajemen persediaan, dan pemantauan lalu lintas. Contohnya adalah sistem yang menggunakan data lalu lintas aktual untuk merencanakan rute pengiriman yang paling cepat dan efisien.
- d. Sumber Daya Manusia: Dalam manajemen sumber daya manusia, sistem penunjang keputusan dapat digunakan untuk pengelolaan kinerja, rekrutmen dan seleksi karyawan, analisis gaji dan tunjangan, dan pengembangan karir. Contohnya adalah sistem yang membantu dalam pengambilan keputusan terkait promosi karyawan berdasarkan kinerja dan kompetensi.

- e. Kesehatan: Dalam bidang kesehatan, sistem penunjang keputusan dapat digunakan untuk diagnosis medis, pengobatan, pengaturan jadwal operasi, dan manajemen data pasien. Contohnya adalah sistem yang menggunakan data gejala dan riwayat medis pasien untuk membantu dokter dalam membuat diagnosis dan rencana pengobatan.
- f. Pemasaran dan Periklanan: Dalam pemasaran dan periklanan, sistem penunjang keputusan dapat membantu dalam analisis pasar, segmentasi pelanggan, pengelolaan kampanye iklan, dan penetapan harga. Contohnya adalah sistem yang menggunakan data perilaku konsumen dan preferensi untuk merancang kampanye pemasaran yang tepat sasaran.
- g. Perencanaan Kota dan Lingkungan: Dalam perencanaan kota dan lingkungan, sistem penunjang keputusan dapat digunakan untuk analisis dampak lingkungan, pengaturan tata ruang, dan pengelolaan sumber daya alam. Contohnya adalah sistem yang membantu dalam pengambilan keputusan terkait penempatan fasilitas publik, pengembangan kawasan, dan pengurangan dampak lingkungan.
- h. Pendidikan: Sistem penunjang keputusan dapat digunakan dalam bidang pendidikan untuk membantu pengambilan keputusan terkait perencanaan kurikulum, penempatan siswa, evaluasi kinerja guru, dan pengembangan program pembelajaran. Contohnya adalah sistem yang menggunakan data akademik dan psikometrik untuk memberikan rekomendasi penempatan siswa ke kelas atau program yang paling sesuai.
- i. Perbankan: Dalam industri perbankan, sistem penunjang keputusan dapat digunakan untuk analisis

kredit, deteksi penipuan, pengelolaan risiko, dan perencanaan keuangan. Contohnya adalah sistem yang menggunakan data historis nasabah dan analisis kredit untuk membantu dalam pengambilan keputusan pemberian pinjaman.

- j. Ritel dan E-commerce: Sistem penunjang keputusan dapat digunakan dalam industri ritel dan e-commerce untuk analisis perilaku konsumen, manajemen persediaan, pengaturan harga, dan personalisasi pengalaman pelanggan. Contohnya adalah sistem yang menggunakan data transaksi dan preferensi pelanggan untuk merekomendasikan produk yang relevan atau menyesuaikan harga secara dinamis.
- k. Energi dan Lingkungan: Dalam industri energi dan lingkungan, sistem penunjang keputusan dapat digunakan untuk manajemen penggunaan energi, analisis efisiensi energi, pengelolaan limbah, dan pemantauan dampak lingkungan. Contohnya adalah sistem yang membantu dalam pengambilan keputusan terkait penggunaan energi yang optimal atau pemilihan teknologi yang ramah lingkungan.
- l. Keamanan dan Kejahatan: Sistem penunjang keputusan dapat digunakan dalam bidang keamanan dan kejahatan untuk analisis pola kejahatan, prediksi risiko keamanan, dan pengambilan keputusan penegakan hukum. Contohnya adalah sistem yang menggunakan data kejahatan dan analisis pola untuk membantu dalam pengambilan keputusan terkait penempatan polisi atau pengaturan patroli.

Ini hanya beberapa contoh aplikasi sistem penunjang keputusan, dan penggunaannya dapat bervariasi tergantung pada kebutuhan dan konteks organisasi atau industri tertentu. Sistem penunjang keputusan dapat digunakan

dalam berbagai bidang untuk membantu pengambil keputusan dalam membuat keputusan yang lebih efektif dan efisien.

Setiap industri atau bidang memiliki kebutuhan unik dalam pengambilan keputusan, dan sistem penunjang keputusan dapat disesuaikan dengan kebutuhan tersebut. Dengan memanfaatkan teknologi dan algoritma yang tepat, sistem penunjang keputusan dapat memberikan analisis data yang lebih baik, meminimalkan kesalahan manusia, dan meningkatkan efisiensi dalam pengambilan keputusan.

10.4.3 Sistem Rekomendasi

1. Definisi sistem rekomendasi

Sistem rekomendasi adalah sistem komputer yang menggunakan teknik analisis data dan algoritma untuk menghasilkan rekomendasi atau saran kepada pengguna tentang item atau konten yang mungkin mereka sukai atau temukan berguna. Tujuan utama dari sistem rekomendasi adalah membantu pengguna dalam menemukan item atau konten yang relevan, meningkatkan pengalaman pengguna, dan memfasilitasi proses pengambilan keputusan.

Sistem rekomendasi bekerja dengan menganalisis data mengenai preferensi, perilaku, dan karakteristik pengguna, serta informasi tentang item atau konten yang tersedia. Data ini dapat berupa riwayat pembelian, penilaian, ulasan, preferensi yang dideklarasikan oleh pengguna, atau data perilaku lainnya. Berdasarkan analisis data tersebut, sistem rekomendasi dapat mengidentifikasi pola dan kesamaan di antara pengguna atau item, dan menggunakan informasi ini untuk menghasilkan rekomendasi yang sesuai.

Sistem rekomendasi dapat diterapkan dalam berbagai domain, termasuk e-commerce, hiburan (misalnya, film atau musik), berita dan konten media, sosial media, aplikasi

perjalanan, dan banyak lagi. Contoh aplikasi sistem rekomendasi termasuk rekomendasi produk berdasarkan riwayat pembelian pengguna, rekomendasi film berdasarkan preferensi penonton sebelumnya, atau rekomendasi teman atau grup di media sosial berdasarkan kesamaan minat.

Pengembangan sistem rekomendasi melibatkan penggunaan teknik seperti filtrasi kolaboratif, filtrasi berbasis konten, pengolahan bahasa alami, atau metode pembelajaran mesin lainnya. Tujuan utama adalah memberikan rekomendasi yang personal dan relevan kepada pengguna, serta meningkatkan tingkat kepuasan dan keterlibatan pengguna dengan platform atau layanan yang diberikan.

Namun, penting untuk memperhatikan bahwa sistem rekomendasi tidak sepenuhnya memahami preferensi atau kebutuhan pengguna dengan sempurna. Pengguna juga harus mempertimbangkan rekomendasi dengan bijak dan melakukan evaluasi sendiri terhadap rekomendasi yang diberikan, mengingat bahwa preferensi dan kebutuhan setiap individu dapat berbeda.

2. Metode dan teknik dalam sistem rekomendasi

Dalam pengembangan sistem rekomendasi, terdapat berbagai metode dan teknik yang digunakan untuk menghasilkan rekomendasi yang relevan dan akurat. Berikut ini adalah beberapa metode dan teknik yang umum digunakan dalam sistem rekomendasi:

- a. *Filtrasi Kolaboratif (Collaborative Filtering)*: Metode ini mencari kesamaan atau pola di antara pengguna atau item berdasarkan riwayat preferensi atau perilaku mereka. Terdapat dua pendekatan dalam filtrasi kolaboratif, yaitu filtrasi kolaboratif berbasis pengguna (*user-based collaborative filtering*) dan filtrasi

kolaboratif berbasis item (*item-based collaborative filtering*). Metode ini tidak memerlukan informasi tentang atribut atau konten item.

- b. Filtrasi Berbasis Konten (*Content-Based Filtering*): Metode ini menggunakan atribut atau konten item untuk membuat rekomendasi yang serupa. Metode ini menganalisis atribut atau konten item, seperti deskripsi, tag, atau atribut lainnya, dan mencocokkan preferensi pengguna dengan atribut tersebut. Misalnya, jika seorang pengguna menyukai film aksi tertentu, sistem rekomendasi berbasis konten dapat merekomendasikan film aksi lain yang memiliki atribut serupa.
- c. Filtrasi Berbasis Hibrida (*Hybrid Filtering*): Pendekatan ini menggabungkan dua atau lebih metode rekomendasi, seperti filtrasi kolaboratif dan filtrasi berbasis konten, untuk meningkatkan akurasi dan keberagaman rekomendasi. Dengan menggabungkan metode-metode ini, sistem rekomendasi dapat memanfaatkan kelebihan masing-masing metode dan mengatasi keterbatasan mereka.
- d. Filtrasi Berbasis Pengetahuan (*Knowledge-Based Filtering*): Pendekatan ini menggunakan basis pengetahuan yang didefinisikan sebelumnya tentang preferensi, aturan bisnis, atau pengetahuan domain untuk membuat rekomendasi. Basis pengetahuan ini dapat berupa aturan yang ditetapkan oleh ahli domain atau preferensi yang dideklarasikan oleh pengguna. Metode ini memanfaatkan pengetahuan eksplisit untuk memberikan rekomendasi yang sesuai.
- e. Sistem Rekomendasi Berbasis Pengetahuan (*Knowledge-Based Recommendation Systems*): Pendekatan ini memanfaatkan pengetahuan tentang preferensi pengguna, pengetahuan domain, dan aturan bisnis

untuk membuat rekomendasi. Metode ini sering digunakan dalam konteks domain yang kompleks, di mana pengetahuan ekspert diperlukan untuk memberikan rekomendasi yang tepat.

- f. Metode Pembelajaran Mesin (*Machine Learning Methods*): Metode ini melibatkan penggunaan algoritma pembelajaran mesin untuk mempelajari pola dan hubungan dalam data pengguna dan item. Algoritma pembelajaran mesin seperti algoritma pembelajaran mendalam (*deep learning*), naive Bayes, atau pohon keputusan dapat digunakan untuk menghasilkan rekomendasi.
- g. Analisis Sekarang (*Context-Aware Analysis*): Pendekatan ini mempertimbangkan konteks atau situasi pengguna saat memberikan rekomendasi. Konteks seperti lokasi, waktu, preferensi musiman, atau preferensi yang berubah-ubah dapat digunakan untuk meningkatkan relevansi dan personalisasi rekomendasi. Misalnya, dalam sistem rekomendasi musik, analisis konteks dapat mempertimbangkan lokasi pengguna saat ini, waktu hari, dan suasana hati untuk merekomendasikan genre musik yang sesuai.
- h. Metode Pemfilteran Demografi (*Demographic Filtering*): Metode ini menggunakan informasi demografis pengguna, seperti usia, jenis kelamin, lokasi, atau preferensi yang dideklarasikan, untuk membuat rekomendasi. Pendekatan ini berfokus pada kesamaan demografis antara pengguna dalam menghasilkan rekomendasi yang relevan.
- i. Metode Pemfilteran Berbasis Peringkat (*Ranking-Based Filtering*): Metode ini menggunakan peringkat atau penilaian yang diberikan oleh pengguna untuk item untuk menghasilkan rekomendasi. Pendekatan ini

mengutamakan item dengan peringkat tertinggi atau penilaian positif yang sesuai dengan preferensi pengguna.

- j. Metode Pemfilteran Temporal (*Temporal Filtering*): Pendekatan ini mempertimbangkan faktor waktu saat membuat rekomendasi. Misalnya, dalam sistem rekomendasi berita, konten yang relevan dan baru lebih diutamakan daripada konten lama. Metode ini memastikan bahwa rekomendasi tetap up-to-date dan sesuai dengan perubahan kebutuhan pengguna.
- k. Sistem Rekomendasi Berbasis Jejak (*Sequential Recommendation Systems*): Pendekatan ini mempertimbangkan urutan atau jejak perilaku pengguna dalam memberikan rekomendasi. Metode ini mengambil keuntungan dari informasi sekuensial, seperti riwayat pembelian atau riwayat penelusuran, untuk memprediksi preferensi pengguna di masa depan.
- l. Metode Pemfilteran Sosial (*Social Filtering*): Pendekatan ini menggunakan informasi dari jejaring sosial atau interaksi sosial pengguna untuk membuat rekomendasi. Misalnya, sistem rekomendasi dapat memanfaatkan informasi tentang teman atau kontak pengguna, ulasan yang diberikan oleh orang lain, atau rekomendasi dari pengguna dengan minat serupa.
- m. Pemfilteran Berbasis Konteks (*Context-Based Filtering*): Pendekatan ini mempertimbangkan konteks yang lebih luas, seperti kondisi lingkungan, situasi sosial, atau faktor eksternal lainnya dalam memberikan rekomendasi. Misalnya, dalam sistem rekomendasi perjalanan, konteks seperti cuaca, musim liburan, atau acara khusus dapat mempengaruhi rekomendasi tempat tujuan yang tepat.

- n. Metode Pemfilteran Berbasis Perilaku (*Behavior-Based Filtering*): Pendekatan ini mengamati perilaku pengguna secara langsung, seperti klik, interaksi, atau preferensi yang diperlihatkan saat ini, untuk membuat rekomendasi. Metode ini berfokus pada informasi terbaru dan perilaku saat ini pengguna.
- o. Metode Pemfilteran Berbasis Kecocokan (*Match-Based Filtering*): Pendekatan ini mencari kesamaan atau kecocokan antara pengguna dengan item atau pengguna dengan pengguna lain untuk menghasilkan rekomendasi. Misalnya, sistem rekomendasi dapat mencocokkan pengguna dengan profil serupa atau dengan minat yang serupa.

Perlu dicatat bahwa metode dan teknik yang digunakan dalam sistem rekomendasi dapat bervariasi tergantung pada konteks dan tujuan aplikasi. Selain itu, kombinasi dari beberapa metode juga dapat diterapkan untuk menghasilkan sistem rekomendasi yang lebih baik dan lebih terpersonalisasi. Selain itu, tidak ada pendekatan yang satu ukuran cocok untuk semua sistem rekomendasi. Pemilihan metode dan teknik yang tepat harus didasarkan pada konteks aplikasi, data yang tersedia, dan tujuan yang ingin dicapai. Kombinasi atau variasi dari pendekatan tersebut juga dapat digunakan untuk menghasilkan sistem rekomendasi yang lebih canggih dan sesuai dengan kebutuhan pengguna.

Selain pendekatan di atas, terdapat juga teknik lanjutan yang digunakan dalam pengembangan sistem rekomendasi, seperti metode pembelajaran mesin (*machine learning*), pemrosesan bahasa alami (*natural language processing*), atau metode optimisasi. Penggunaan teknik-teknik ini tergantung pada kebutuhan dan kompleksitas sistem rekomendasi yang dikembangkan.

Penting untuk diingat bahwa pengembangan sistem rekomendasi melibatkan pengumpulan dan analisis data pengguna. Oleh karena itu, privasi dan keamanan data harus diperhatikan dalam merancang dan mengimplementasikan sistem rekomendasi, serta mematuhi peraturan dan kebijakan yang berlaku.

Setelah memilih pendekatan yang sesuai, Tahapan pengembangan sistem rekomendasi meliputi:

- a. Pengumpulan dan Pemrosesan Data: Tahap ini melibatkan pengumpulan data tentang pengguna, item, dan interaksi antara pengguna dan item. Data ini dapat mencakup riwayat pembelian, penilaian, ulasan, preferensi yang dideklarasikan oleh pengguna, atau data perilaku lainnya. Data kemudian diproses untuk menghilangkan kebisingan, mengubahnya ke dalam format yang sesuai, dan menyiapkan data untuk analisis lebih lanjut.
- b. Analisis Data: Tahap ini melibatkan analisis data untuk mengidentifikasi pola dan kesamaan antara pengguna dan item. Berbagai teknik analisis data dapat digunakan, seperti clustering, analisis faktor, atau metode pembelajaran mesin seperti algoritma pembelajaran mendalam (deep learning) atau algoritma pemfilteran kolaboratif. Tujuan analisis data adalah untuk menemukan hubungan atau preferensi yang tersembunyi dalam data yang dapat digunakan untuk menghasilkan rekomendasi yang relevan.
- c. Perancangan Model: Setelah analisis data, model rekomendasi dikembangkan berdasarkan pendekatan yang dipilih. Model ini mencakup algoritma dan aturan yang digunakan untuk menghasilkan rekomendasi. Desain model harus mempertimbangkan kebutuhan dan karakteristik sistem rekomendasi yang diinginkan,

seperti tingkat personalisasi, keakuratan, dan keberagaman rekomendasi.

- d. Implementasi: Tahap ini melibatkan implementasi model rekomendasi ke dalam sistem atau platform yang relevan. Hal ini dapat melibatkan pengembangan perangkat lunak, integrasi dengan basis data, dan penerapan algoritma rekomendasi yang telah dirancang. Pada tahap ini juga diperlukan pengujian untuk memastikan bahwa sistem rekomendasi berfungsi dengan baik dan menghasilkan rekomendasi yang sesuai.
- e. Evaluasi dan Peningkatan: Setelah implementasi, sistem rekomendasi harus dievaluasi untuk memastikan kinerjanya sesuai dengan tujuan yang ditetapkan. Evaluasi dilakukan dengan menggunakan metrik yang relevan, seperti tingkat keberhasilan rekomendasi, kepuasan pengguna, atau tingkat konversi. Jika diperlukan, perbaikan dan penyesuaian dilakukan untuk meningkatkan kinerja sistem rekomendasi.
- f. Pemeliharaan dan Pembaruan: Sistem rekomendasi perlu dipelihara dan diperbarui secara berkala. Ini melibatkan pemantauan kinerja sistem, pembaruan data, dan penyesuaian model rekomendasi sesuai dengan perubahan preferensi atau kondisi pengguna. Pemeliharaan dan pembaruan yang tepat waktu diperlukan untuk memastikan bahwa sistem rekomendasi tetap relevan dan efektif.

Pengembangan sistem rekomendasi adalah proses iteratif yang membutuhkan pemahaman yang baik tentang kebutuhan pengguna dan data yang tersedia. Selain itu, penting juga untuk memperhatikan privasi dan keamanan data, serta mengikuti regulasi yang berlaku terkait dengan pengumpulan dan penggunaan data pengguna.

Pengembangan sistem rekomendasi juga harus melibatkan interaksi dengan pengguna dan mendengarkan umpan balik mereka untuk terus meningkatkan pengalaman rekomendasi.

Selain itu, perlu juga mempertimbangkan faktor-faktor seperti skalabilitas sistem untuk mengatasi jumlah pengguna dan item yang besar, efisiensi komputasi untuk menghasilkan rekomendasi dalam waktu yang wajar, dan adaptabilitas sistem untuk mengakomodasi perubahan lingkungan atau preferensi pengguna.

Dalam proses pengembangan sistem rekomendasi, penting untuk memiliki tim yang terampil dan berpengetahuan dalam bidang analisis data, pembelajaran mesin, pemrosesan bahasa alami, dan rekayasa perangkat lunak. Kolaborasi antara ahli domain, analis data, dan pengembang perangkat lunak sangat penting untuk memastikan pengembangan sistem rekomendasi yang efektif dan efisien.

Selama seluruh proses pengembangan, perhatian juga harus diberikan pada etika dalam penggunaan sistem rekomendasi. Hal ini termasuk kebijakan privasi yang jelas, transparansi dalam algoritma rekomendasi, dan perlindungan terhadap bias yang tidak disengaja atau diskriminasi dalam rekomendasi.

Dengan pengembangan yang tepat, sistem rekomendasi dapat memberikan manfaat yang signifikan, baik bagi pengguna maupun bisnis. Pengguna dapat mendapatkan rekomendasi yang relevan dan personal, yang memudahkan mereka dalam menemukan item atau konten yang mereka minati. Di sisi bisnis, sistem rekomendasi dapat meningkatkan retensi pengguna, meningkatkan konversi, dan memperluas penjualan melalui rekomendasi produk yang tepat waktu dan relevan.

Dalam konteks teknologi informasi yang terus berkembang, sistem rekomendasi terus menjadi bidang penelitian dan pengembangan yang menarik. Dengan kemajuan dalam bidang kecerdasan buatan dan analisis data, diharapkan sistem rekomendasi dapat terus berkembang untuk memberikan rekomendasi yang lebih baik, akurat, dan personal sesuai dengan preferensi dan kebutuhan pengguna.

3. Contoh aplikasi sistem rekomendasi

Ada banyak contoh aplikasi sistem rekomendasi yang digunakan dalam berbagai bidang. Berikut ini adalah beberapa contoh populer:

- a. Sistem Rekomendasi E-commerce: Contoh paling umum adalah sistem rekomendasi yang digunakan oleh platform e-commerce seperti Amazon, eBay, dan Alibaba. Sistem ini memberikan rekomendasi produk kepada pengguna berdasarkan riwayat pembelian mereka, preferensi produk, atau perilaku penelusuran.
- b. Sistem Rekomendasi Musik dan Video: Platform streaming musik dan video seperti Spotify, Netflix, dan YouTube menggunakan sistem rekomendasi untuk memberikan rekomendasi lagu, artis, film, atau video kepada pengguna berdasarkan preferensi musik mereka, riwayat putar, atau pola tontonan sebelumnya.
- c. Sistem Rekomendasi Perjalanan: Situs perjalanan seperti Booking.com, TripAdvisor, dan Airbnb menggunakan sistem rekomendasi untuk merekomendasikan hotel, restoran, tujuan wisata, atau akomodasi kepada pengguna berdasarkan preferensi mereka, ulasan pengguna, atau kriteria pencarian.
- d. Sistem Rekomendasi Buku: Situs e-commerce buku seperti Goodreads atau Amazon Books menggunakan

sistem rekomendasi untuk memberikan rekomendasi buku kepada pengguna berdasarkan minat baca mereka, penilaian buku sebelumnya, atau preferensi genre.

- e. Sistem Rekomendasi Sosial Media: Platform media sosial seperti Facebook, Instagram, dan Twitter menggunakan sistem rekomendasi untuk memberikan rekomendasi teman, halaman, atau konten yang relevan kepada pengguna berdasarkan jejaring sosial mereka, minat, atau interaksi sebelumnya.
- f. Sistem Rekomendasi Restoran: Aplikasi seperti Yelp atau Zomato menggunakan sistem rekomendasi untuk memberikan rekomendasi restoran atau tempat makan kepada pengguna berdasarkan preferensi kuliner, lokasi, atau ulasan pengguna lainnya.
- g. Sistem Rekomendasi Karir: Platform pencarian kerja atau pengembangan karir menggunakan sistem rekomendasi untuk memberikan rekomendasi pekerjaan atau peluang karir kepada pengguna berdasarkan profil mereka, pengalaman kerja, atau preferensi industri.
- h. Sistem Rekomendasi Penelitian dan Literatur: Dalam bidang akademik, sistem rekomendasi digunakan untuk memberikan rekomendasi artikel penelitian, publikasi, atau literatur yang relevan kepada peneliti berdasarkan minat riset mereka, riwayat publikasi, atau citasi.
- i. Sistem Rekomendasi Mode dan Pakaian: Aplikasi e-commerce dan platform fashion seperti ASOS, Zalando, atau Polyvore menggunakan sistem rekomendasi untuk memberikan rekomendasi pakaian atau gaya kepada pengguna berdasarkan preferensi fashion mereka, gaya pakaian, atau trend terkini.
- j. Sistem Rekomendasi Makanan dan Resep: Aplikasi dan situs web seperti Allrecipes atau Tasty menggunakan

sistem rekomendasi untuk memberikan rekomendasi makanan, resep, atau masakan kepada pengguna berdasarkan preferensi kuliner, diet, atau bahan yang tersedia.

Ini hanya beberapa contoh aplikasi sistem rekomendasi yang dapat ditemukan dalam berbagai industri. Sistem rekomendasi menjadi sangat penting dalam membantu pengguna menavigasi melalui jumlah besar informasi yang tersedia dan menyediakan rekomendasi yang relevan dan personalisasi. Dengan memanfaatkan data pengguna, preferensi, dan pola perilaku, sistem rekomendasi dapat membantu pengguna menemukan konten, produk, atau layanan yang sesuai dengan kebutuhan dan minat mereka.

Selain itu, sistem rekomendasi juga memberikan manfaat bagi penyedia layanan atau platform yang mengimplementasikannya. Dengan memberikan rekomendasi yang tepat kepada pengguna, mereka dapat meningkatkan pengalaman pengguna, meningkatkan tingkat keterlibatan, dan mempromosikan produk atau layanan tambahan. Selain itu, sistem rekomendasi dapat membantu dalam meningkatkan retensi pengguna, menghasilkan penjualan tambahan, dan meningkatkan loyalitas pelanggan.

Pengembangan sistem rekomendasi melibatkan analisis data, penggunaan algoritma dan teknik pemfilteran, serta pemahaman yang mendalam tentang preferensi dan perilaku pengguna. Metode dan teknik yang digunakan dalam sistem rekomendasi terus berkembang dengan kemajuan dalam bidang kecerdasan buatan dan analisis data.

Penting untuk dicatat bahwa sistem rekomendasi harus menjaga keseimbangan antara relevansi dan privasi pengguna. Keberhasilan sistem rekomendasi tergantung pada kemampuannya untuk memahami dan menginterpretasikan

preferensi pengguna dengan benar tanpa melanggar privasi mereka. Pengguna juga harus memiliki kontrol dan transparansi atas data mereka yang digunakan dalam membuat rekomendasi. Dalam kesimpulan, sistem rekomendasi adalah alat yang kuat dalam membantu pengguna menemukan informasi yang relevan, produk, atau layanan yang sesuai dengan preferensi mereka. Dengan menggabungkan data pengguna, analisis cerdas, dan teknik pemfilteran yang canggih, sistem rekomendasi dapat memberikan pengalaman yang dipersonalisasi dan meningkatkan keterlibatan pengguna.

10.5 Tantangan dan Tren Masa Depan

10.5.1 Tantangan dalam Sistem Berbasis Pengetahuan

Pengembangan dan implementasi sistem berbasis pengetahuan juga melibatkan beberapa tantangan yang perlu diatasi. Berikut adalah beberapa tantangan utama yang sering dihadapi dalam sistem berbasis pengetahuan:

1. **Pemerolehan Pengetahuan:** Salah satu tantangan utama adalah mendapatkan pengetahuan yang relevan dan benar. Pemerolehan pengetahuan melibatkan identifikasi, pengumpulan, dan representasi pengetahuan dari sumber yang berbeda, seperti ahli domain, dokumen, basis data, atau pengalaman. Tantangan ini melibatkan pemahaman yang mendalam tentang domain yang bersangkutan dan memastikan integritas dan kualitas pengetahuan yang diperoleh.
2. **Penanganan Pengetahuan yang Tidak Tepat:** Sistem berbasis pengetahuan harus dapat mengatasi ketidakpastian, keambiguan, atau ketidaklengkapan dalam pengetahuan yang diperoleh. Tantangan ini melibatkan pengembangan metode dan teknik untuk mengelola pengetahuan yang tidak sempurna, termasuk penanganan

- pengecualian, perbedaan pendapat, atau pemrosesan bahasa alami yang ambigu.
3. **Pemeliharaan dan Evolusi Pengetahuan:** Pengetahuan dalam sistem berbasis pengetahuan tidak statis dan harus diperbarui atau diperluas seiring berjalannya waktu. Tantangan ini termasuk dalam pemeliharaan pengetahuan yang ada, penambahan pengetahuan baru, dan penghapusan pengetahuan yang usang atau tidak relevan. Pembaruan pengetahuan juga membutuhkan pemahaman yang baik tentang dampak perubahan pada sistem secara keseluruhan.
 4. **Skalabilitas:** Tantangan lain adalah menghadapi pertumbuhan jumlah pengetahuan dan kompleksitas sistem. Sistem berbasis pengetahuan harus mampu menangani volume pengetahuan yang besar, baik dalam hal pengelolaan dan penyimpanan pengetahuan, maupun dalam pemrosesan dan inferensi yang efisien.
 5. **Interpretasi dan Pemahaman Konteks:** Sistem berbasis pengetahuan harus mampu memahami dan menginterpretasikan konteks dalam menghasilkan pengetahuan atau memberikan rekomendasi. Tantangan ini melibatkan pemahaman yang baik tentang situasi atau lingkungan di mana sistem beroperasi, serta kemampuan untuk beradaptasi dengan perubahan konteks yang mungkin terjadi.
 6. **Validasi dan Evaluasi:** Evaluasi dan validasi sistem berbasis pengetahuan adalah tantangan penting untuk memastikan keandalan, kualitas, dan kinerja sistem. Diperlukan metode dan kriteria evaluasi yang sesuai untuk mengukur efektivitas sistem, termasuk pengujian dan verifikasi pengetahuan, serta evaluasi pengguna.
 7. **Ketergantungan pada Ahli Domain:** Tantangan lain adalah ketergantungan pada keahlian ahli domain dalam

pengembangan dan pemeliharaan sistem berbasis pengetahuan. Pengetahuan ahli domain sering kali sulit didapatkan, dan ada risiko ketergantungan pada individu tertentu. Tantangan ini membutuhkan strategi untuk mengelola pengetahuan yang terkait dengan ahli domain, termasuk dokumentasi yang baik, pembagian pengetahuan dengan anggota tim yang lebih luas, atau pendekatan kolaboratif yang melibatkan berbagai pemangku kepentingan.

8. Integrasi dengan Sistem Lain: Sistem berbasis pengetahuan sering kali perlu diintegrasikan dengan sistem lain dalam organisasi atau lingkungan yang lebih luas. Tantangan ini melibatkan pemahaman yang baik tentang infrastruktur yang ada, interoperabilitas dengan sistem lain dan integrasi data yang efektif.
9. Pengelolaan Pengetahuan yang Kompleks: Sistem berbasis pengetahuan sering menghadapi pengetahuan yang kompleks, termasuk aturan yang saling terkait, pengetahuan temporal, atau pengetahuan dengan variasi tinggi. Tantangan ini melibatkan pengembangan model representasi yang tepat dan algoritma inferensi yang efisien untuk menangani kompleksitas ini.
10. Dukungan Pengguna: Dalam beberapa kasus, tantangan terkait dukungan pengguna dalam menggunakan sistem berbasis pengetahuan dapat muncul. Pengguna mungkin membutuhkan bantuan dalam memahami rekomendasi atau penjelasan dari sistem, atau dalam berinteraksi dengan antarmuka sistem. Penting untuk menyediakan dokumentasi, pelatihan, atau bantuan pengguna yang memadai.
11. Keamanan dan Privasi: Pengelolaan keamanan dan privasi data dalam sistem berbasis pengetahuan adalah tantangan penting. Perlindungan data sensitif dan privasi pengguna

harus menjadi prioritas utama, termasuk kebijakan akses yang tepat, enkripsi data, dan kepatuhan terhadap peraturan privasi yang berlaku.

Menghadapi tantangan-tantangan ini memerlukan pendekatan yang cermat dalam pengembangan dan implementasi sistem berbasis pengetahuan. Penggunaan metode dan teknik yang tepat, pemahaman yang mendalam tentang domain dan pengguna, serta kolaborasi antara ahli domain, pengembang, dan pemangku kepentingan lainnya dapat membantu mengatasi tantangan-tantangan ini dan memastikan keberhasilan sistem berbasis pengetahuan.

a. Keterbatasan dalam pengembangan sistem berbasis pengetahuan

Pengembangan sistem berbasis pengetahuan juga memiliki beberapa keterbatasan yang perlu diperhatikan. Berikut adalah beberapa keterbatasan umum yang mungkin muncul:

1. Keterbatasan Pengetahuan: Sistem berbasis pengetahuan sangat tergantung pada kualitas dan kuantitas pengetahuan yang ada. Jika pengetahuan yang diperoleh tidak cukup atau tidak memadai, maka kualitas rekomendasi atau hasil inferensi sistem dapat terpengaruh. Dalam beberapa kasus, mendapatkan pengetahuan yang akurat dan lengkap bisa menjadi tantangan tersendiri.
2. Ketidakmampuan untuk Menghadapi Informasi Baru: Sistem berbasis pengetahuan mungkin sulit untuk mengakomodasi informasi baru atau pengetahuan yang berkembang dengan cepat. Jika sistem tidak dirancang dengan fleksibilitas yang cukup, maka dapat sulit untuk memperbarui atau memasukkan pengetahuan baru ke dalam sistem secara efisien.

3. Kurangnya Kemampuan Pemahaman Konteks: Sistem berbasis pengetahuan mungkin mengalami kesulitan dalam memahami dan menginterpretasikan konteks yang kompleks. Pengetahuan yang dimiliki oleh sistem mungkin tidak cukup untuk sepenuhnya memahami situasi atau lingkungan di mana sistem beroperasi. Hal ini dapat menghasilkan rekomendasi yang kurang tepat atau pemahaman yang terbatas terhadap preferensi pengguna.
4. Keterbatasan dalam Kemampuan Pemrosesan Data: Sistem berbasis pengetahuan yang kompleks dan berukuran besar dapat memerlukan pemrosesan data yang intensif. Keterbatasan dalam sumber daya komputasi atau infrastruktur dapat membatasi kinerja dan skalabilitas sistem. Pemrosesan yang lambat atau tidak responsif dapat mempengaruhi pengalaman pengguna dan efektivitas sistem secara keseluruhan.
5. Tantangan dalam Validasi dan Evaluasi: Mengevaluasi dan mengukur kinerja sistem berbasis pengetahuan dapat menjadi sulit. Memperoleh data evaluasi yang relevan dan membandingkan kinerja sistem dengan standar atau metrik yang sesuai adalah tantangan yang nyata. Validasi dan evaluasi yang tidak memadai dapat menghambat pengembangan dan peningkatan sistem.
6. Ketergantungan pada Ahli Domain: Pengembangan sistem berbasis pengetahuan sering kali bergantung pada keahlian ahli domain. Tantangan ini melibatkan keterbatasan pengetahuan yang tersedia dari ahli domain yang mungkin terbatas dalam jumlah atau ketersediaannya. Ketergantungan pada individu tertentu juga dapat menjadi masalah jika pengetahuan tidak dibagikan secara luas atau tidak terdokumentasi dengan baik.

7. **Resolusi Konflik atau Perbedaan Pendapat:** Dalam beberapa kasus, sistem berbasis pengetahuan dapat menghadapi situasi di mana ada konflik atau perbedaan pendapat antara pengetahuan yang ada. Penyelesaian konflik atau pengelolaan perbedaan pendapat dapat menjadi tantangan yang rumit dan memerlukan pendekatan yang tepat untuk memastikan konsistensi dan kualitas hasil inferensi atau rekomendasi yang diberikan oleh sistem.
8. **Tidak Mampu Menggantikan Kecerdasan Manusia:** Meskipun sistem berbasis pengetahuan dapat memiliki pengetahuan yang kaya dan kemampuan inferensi yang baik, mereka masih memiliki keterbatasan dalam hal pemahaman yang serba jitu, intuisi, dan penyesuaian dengan situasi yang kompleks seperti yang dilakukan oleh manusia. Sistem berbasis pengetahuan mungkin tidak mampu menggantikan sepenuhnya kecerdasan manusia dalam beberapa konteks.
9. **Kendala dalam Ekstraksi Pengetahuan Tersirat:** Ada beberapa jenis pengetahuan yang sulit diekstraksi secara langsung dari sumbernya, seperti pengetahuan yang tersirat dalam teks yang ambigu atau pengetahuan yang tersembunyi dalam pola data yang kompleks. Mengidentifikasi dan mengakuisisi pengetahuan semacam ini dapat menjadi tantangan tersendiri dan memerlukan teknik dan algoritma yang canggih.
10. **Keterbatasan Domain:** Sistem berbasis pengetahuan biasanya beroperasi dalam domain atau bidang tertentu. Mereka mungkin memiliki keterbatasan dalam beradaptasi dengan domain yang berbeda atau mengatasi pertanyaan atau masalah di luar cakupan

domain mereka. Hal ini memerlukan perencanaan yang hati-hati dalam perancangan sistem untuk memastikan bahwa sistem dapat diterapkan dengan efektif dalam domain yang dituju.

11. Tantangan Integrasi dengan Sistem Lain: Integrasi sistem berbasis pengetahuan dengan sistem lain dalam lingkungan yang kompleks dapat menjadi keterbatasan. Sistem tersebut mungkin perlu berinteraksi dengan sumber data eksternal, infrastruktur IT yang ada, atau sistem lain yang digunakan dalam organisasi. Memastikan interoperabilitas dan integrasi yang mulus dapat menjadi tantangan teknis.

Meskipun sistem berbasis pengetahuan memiliki keterbatasan tersebut, dengan pendekatan yang tepat dan pemecahan masalah yang inovatif, banyak tantangan tersebut dapat diatasi. Terus berkembangnya teknologi seperti pembelajaran mesin, pemrosesan bahasa alami, dan komputasi awan juga memberikan potensi untuk mengatasi beberapa keterbatasan tersebut dan meningkatkan kinerja dan fleksibilitas sistem berbasis pengetahuan.

- b. Etika dan privasi dalam penggunaan sistem berbasis pengetahuan

Etika dan privasi adalah dua aspek penting yang perlu diperhatikan dalam penggunaan sistem berbasis pengetahuan. Dalam pengumpulan, pengolahan, dan penggunaan data serta pengetahuan pengguna, beberapa prinsip etika dan privasi berikut harus diikuti:

1. Privasi Data: Penting untuk melindungi privasi data pengguna yang dikumpulkan dalam sistem berbasis pengetahuan. Hal ini melibatkan kebijakan dan praktik

yang memastikan bahwa data pribadi tidak diungkapkan kepada pihak ketiga tanpa persetujuan pengguna, dan bahwa langkah-langkah keamanan yang tepat diambil untuk melindungi data dari akses yang tidak sah.

2. **Transparansi:** Pengguna sistem berbasis pengetahuan harus diberikan informasi yang jelas tentang bagaimana data mereka dikumpulkan, diproses, dan digunakan dalam sistem. Mereka harus diberi pemahaman yang jelas tentang bagaimana pengetahuan dan informasi mereka digunakan untuk memberikan rekomendasi atau hasil tertentu.
3. **Keterbukaan:** Sistem berbasis pengetahuan yang etis harus memberikan pengguna kemampuan untuk mengakses dan mengelola data pribadi mereka. Pengguna harus dapat memperbarui, menghapus, atau mengatur preferensi mereka terkait dengan penggunaan data mereka dalam sistem.
4. **Keberpihakan dan Keadilan:** Sistem berbasis pengetahuan harus memperlakukan pengguna dengan adil dan tidak mendiskriminasi. Tidak boleh ada bias dalam pengumpulan atau penggunaan data yang dapat menghasilkan perlakuan yang tidak adil atau diskriminatif terhadap individu atau kelompok tertentu.
5. **Pertimbangan Etis dalam Pengetahuan dan Rekomendasi:** Pengetahuan yang digunakan dalam sistem berbasis pengetahuan harus diperoleh dengan cara yang etis dan legal. Rekomendasi atau hasil yang diberikan oleh sistem juga harus mempertimbangkan nilai-nilai etis, kepentingan pengguna, dan dampak sosial yang mungkin timbul.

6. Kejelasan Tujuan dan Batasan Sistem: Pengguna harus diberikan pemahaman yang jelas tentang tujuan dan batasan sistem berbasis pengetahuan. Mereka harus mengetahui apa yang diharapkan dari sistem dan apa yang tidak dapat diandalkan. Hal ini membantu menghindari kesalahpahaman atau ekspektasi yang tidak realistis.
7. Evaluasi dan Pemantauan: Sistem berbasis pengetahuan perlu secara teratur dievaluasi dan dipantau untuk memastikan kepatuhan terhadap prinsip etika dan privasi. Jika terjadi pelanggaran atau kekhawatiran etika, langkah-langkah perbaikan atau perubahan perlu diambil.

Penting untuk melibatkan ahli etika dan privasi dalam pengembangan dan penggunaan sistem berbasis pengetahuan untuk memastikan bahwa aspek-aspek ini diperhatikan dan diimplementasikan dengan baik. Selain prinsip etika dan privasi yang telah disebutkan sebelumnya, berikut adalah beberapa pertimbangan penting dalam penggunaan sistem berbasis pengetahuan:

1. Kejelasan dan Transparansi: Sistem berbasis pengetahuan harus memiliki dokumentasi dan penjelasan yang jelas tentang cara kerjanya, alasan di balik rekomendasi atau hasil yang diberikan, dan bagaimana pengetahuan diperoleh atau dipelajari. Ini membantu pengguna dan pemangku kepentingan lainnya memahami proses yang terlibat dan meningkatkan kepercayaan terhadap sistem.
2. Kesalahan dan Akuntabilitas: Sistem berbasis pengetahuan harus dapat mengakui dan mengelola kesalahan atau ketidakpastian dalam proses inferensi atau rekomendasi. Jika terjadi kesalahan atau masalah, sistem harus memiliki mekanisme untuk melacak dan memperbaiki kesalahan

- tersebut. Selain itu, ada kebutuhan untuk mengidentifikasi dan mempertanggungjawabkan pemilik sistem berbasis pengetahuan jika terjadi dampak negatif yang signifikan.
3. Keamanan Data: Sistem berbasis pengetahuan harus melindungi data pengguna dan pengetahuan yang disimpan dari akses yang tidak sah atau penggunaan yang tidak sesuai. Keamanan data harus diutamakan dengan menerapkan praktik keamanan yang kuat seperti enkripsi, akses terbatas, dan pengelolaan hak akses.
 4. Bias dan Diskriminasi: Sistem berbasis pengetahuan harus diuji secara cermat untuk mengidentifikasi dan mengatasi bias atau diskriminasi yang tidak disengaja. Bias dapat muncul dalam pengumpulan dan analisis data serta dalam proses inferensi. Upaya harus dilakukan untuk mengurangi bias dan memastikan bahwa sistem memberikan rekomendasi yang adil dan merata kepada semua pengguna.
 5. Penggunaan yang Bertanggung Jawab: Penggunaan sistem berbasis pengetahuan harus dilakukan dengan pertimbangan etis dan tanggung jawab. Sistem tidak boleh digunakan untuk tujuan yang merugikan atau melanggar hak privasi individu. Ada kebutuhan untuk menjaga integritas sistem dan memastikan bahwa penggunaannya sejalan dengan nilai-nilai etis dan regulasi yang berlaku.
 6. Pembaruan dan Perubahan: Sistem berbasis pengetahuan harus memiliki mekanisme untuk pembaruan dan perubahan yang diperlukan seiring waktu. Pengetahuan dan aturan sistem harus diperbarui sesuai dengan perkembangan baru dalam domain atau lingkungan yang relevan. Ini memastikan keberlanjutan dan relevansi sistem dalam jangka panjang.

Penting untuk melibatkan ahli etika, pakar privasi, dan pemangku kepentingan lainnya dalam pengembangan dan implementasi sistem berbasis pengetahuan. Ini akan membantu memastikan bahwa semua pertimbangan etika dan privasi yang relevan diambil dengan serius dan diterapkan dengan baik.

10.5.2 Tren Masa Depan

Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan. Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan (*artificial intelligence*/AI) memiliki potensi besar untuk meningkatkan kinerja dan kemampuan sistem tersebut. Berikut adalah beberapa aspek penting dalam penggabungan kedua teknologi ini:

1. Pembelajaran Mesin: Kecerdasan buatan memanfaatkan teknik pembelajaran mesin untuk memungkinkan sistem berbasis pengetahuan untuk belajar dari data dan pengalaman. Dengan menggunakan algoritma pembelajaran mesin, sistem dapat mengenali pola, mengidentifikasi tren, dan meningkatkan kinerja berdasarkan umpan balik yang diterima.
2. Pengambilan Keputusan Otomatis: Dengan memanfaatkan kecerdasan buatan, sistem berbasis pengetahuan dapat meningkatkan kemampuan pengambilan keputusan secara otomatis. Kecerdasan buatan dapat memberikan penilaian dan rekomendasi yang lebih canggih berdasarkan data yang dikumpulkan dan pengetahuan yang dimiliki oleh sistem.
3. Pemrosesan Bahasa Alami: Kecerdasan buatan juga memungkinkan integrasi sistem berbasis pengetahuan dengan pemrosesan bahasa alami. Ini memungkinkan pengguna untuk berinteraksi dengan sistem menggunakan

- bahasa alami, seperti bertanya atau memberikan instruksi, dan sistem dapat memahami dan merespons dengan tepat.
4. **Pengoptimalan dan Penyesuaian Otomatis:** Dengan kecerdasan buatan, sistem berbasis pengetahuan dapat secara otomatis mengoptimalkan pengetahuan dan aturan yang dimilikinya. Sistem dapat mempelajari perilaku pengguna, menyesuaikan preferensi, dan meningkatkan kinerja berdasarkan umpan balik yang diterima.
 5. **Analisis Data yang Lebih Lanjut:** Kecerdasan buatan dapat membantu sistem berbasis pengetahuan dalam melakukan analisis data yang lebih lanjut. Dengan teknik pengolahan data yang canggih, sistem dapat mengidentifikasi pola yang lebih kompleks, menemukan wawasan baru, dan menghasilkan pengetahuan yang lebih mendalam.
 6. **Pengambilan Keputusan yang Berbasis Pengetahuan:** Kecerdasan buatan dapat digunakan untuk memperkuat proses pengambilan keputusan dengan menggabungkan pengetahuan eksplisit dalam sistem berbasis pengetahuan dengan kemampuan analisis dan inferensi AI. Hal ini memungkinkan sistem untuk memberikan rekomendasi yang lebih cerdas dan mendalam.

Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan membawa manfaat besar dalam meningkatkan kemampuan sistem, memungkinkan pembelajaran dan adaptasi yang lebih baik, serta memberikan keputusan yang lebih cerdas. Namun, juga penting untuk mempertimbangkan aspek-etika dan privasi dalam penggunaan teknologi ini dan memastikan bahwa sistem dijalankan dengan integritas dan tanggung jawab yang tepat.

Terdapat beberapa aspek yang perlu diperhatikan dalam penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan:

1. Integrasi Pengetahuan dan Pembelajaran: Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan memungkinkan pengetahuan yang ada dalam sistem berbasis pengetahuan digunakan sebagai dasar untuk pembelajaran mesin. Pembelajaran mesin dapat memperluas pengetahuan sistem berbasis pengetahuan dengan mengidentifikasi pola, menggali wawasan baru, dan mengoptimalkan proses pengambilan keputusan.
2. Penyimpanan dan Manajemen Pengetahuan: Sistem berbasis pengetahuan membutuhkan struktur dan metode penyimpanan pengetahuan yang efisien. Dalam penggabungan dengan kecerdasan buatan, penting untuk mengembangkan infrastruktur yang memadai untuk menyimpan dan mengelola pengetahuan yang diperoleh dari pembelajaran mesin. Ini mencakup pemilihan teknik penyimpanan, pengindeksan yang efisien, dan metode penelusuran yang cepat.
3. Algoritma Inferensi yang Efektif: Dalam penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan, perlu dipertimbangkan algoritma inferensi yang efektif untuk menerapkan pengetahuan dan aturan yang ada dalam sistem. Algoritma ini harus mampu mengambil keputusan yang cerdas, mengatasi ketidakpastian, dan menghasilkan hasil yang dapat dijelaskan dan dipertanggungjawabkan.
4. Skalabilitas: Sistem berbasis pengetahuan yang dikombinasikan dengan kecerdasan buatan harus memiliki kemampuan untuk skalabilitas. Artinya, sistem harus mampu mengelola dan memproses volume data yang besar, serta dapat menangani permintaan yang meningkat seiring waktu. Skalabilitas penting untuk memastikan kinerja yang baik dan penggunaan yang efisien dari sistem.

5. Evaluasi dan Pengujian: Penting untuk melakukan evaluasi dan pengujian yang cermat pada sistem yang dikembangkan. Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan memerlukan metode evaluasi yang memadai untuk mengukur keberhasilan sistem, baik dalam hal kinerja, akurasi, maupun kepuasan pengguna. Evaluasi ini dapat melibatkan pengujian fungsionalitas sistem, pengujian terhadap skenario penggunaan yang berbeda, serta pengujian kinerja untuk mengukur waktu respons dan skalabilitas.

Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan dapat menghasilkan sistem yang lebih cerdas, adaptif, dan mampu mengambil keputusan yang lebih baik. Namun, penting untuk memperhatikan aspek-etika dan privasi dalam pengembangan dan penggunaan sistem ini serta memastikan perlindungan data dan integritas sistem yang tepat.

Selain aspek yang telah disebutkan sebelumnya, ada beberapa hal lain yang perlu dipertimbangkan dalam penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan:

1. Interpretabilitas: Kecerdasan buatan cenderung menggunakan algoritma yang kompleks dan sulit diinterpretasikan oleh manusia. Namun, dalam sistem berbasis pengetahuan, interpretasi pengetahuan oleh manusia sangat penting. Oleh karena itu, dalam penggabungan ini, diperlukan upaya untuk menjaga interpretasi dan pemahaman yang baik terhadap pengetahuan yang dimiliki oleh sistem.
2. Pemeliharaan dan Pembaruan Pengetahuan: Sistem berbasis pengetahuan umumnya membutuhkan pemeliharaan dan pembaruan yang berkelanjutan untuk

menjaga kebaruan dan relevansi pengetahuan. Dalam penggabungan dengan kecerdasan buatan, penting untuk memastikan bahwa pengetahuan yang diperoleh dari pembelajaran mesin dapat diintegrasikan dan diperbarui secara efisien ke dalam sistem berbasis pengetahuan.

3. Manajemen Kesalahan dan Ketidakpastian: Kecerdasan buatan dapat memperkenalkan ketidakpastian dalam pengambilan keputusan dan analisis data. Oleh karena itu, dalam penggabungan ini, perlu ada mekanisme untuk mengelola dan menangani kesalahan serta ketidakpastian yang mungkin muncul dalam proses inferensi dan pengambilan keputusan.
4. Pengamanan dan Privasi: Dalam penggunaan sistem berbasis pengetahuan yang dikombinasikan dengan kecerdasan buatan, perlindungan data dan privasi pengguna sangat penting. Penggunaan data yang sensitif harus dilakukan dengan hati-hati, dan mekanisme keamanan yang tepat harus diterapkan untuk melindungi integritas dan kerahasiaan informasi yang disimpan dalam sistem.
5. Penggunaan Eksplanabilitas: Dalam keputusan yang dibuat oleh sistem berbasis pengetahuan yang dikombinasikan dengan kecerdasan buatan, eksplanabilitas menjadi faktor penting. Pengguna atau pihak yang terkait harus dapat memahami alasan di balik keputusan yang diambil oleh sistem. Oleh karena itu, diperlukan upaya untuk menjelaskan dan mempertanggungjawabkan proses inferensi dan logika yang digunakan dalam pengambilan keputusan.

Penggabungan sistem berbasis pengetahuan dengan kecerdasan buatan memberikan peluang untuk menciptakan sistem yang lebih pintar dan adaptif. Namun, tantangan seperti

interpretabilitas, pemeliharaan pengetahuan, manajemen kesalahan, keamanan, dan eksplanabilitas perlu diperhatikan dengan cermat agar penggunaan sistem dapat dilakukan dengan aman, efektif, dan sesuai dengan etika yang berlaku.

10.5.3 Penggunaan big data dalam sistem berbasis pengetahuan

Penggunaan Big Data dalam sistem berbasis pengetahuan memberikan potensi yang besar dalam menghasilkan wawasan yang lebih mendalam dan pengambilan keputusan yang lebih cerdas. Big Data merujuk pada volume besar data yang dihasilkan secara terus-menerus dari berbagai sumber seperti sensor, perangkat IoT, media sosial, transaksi bisnis, dan banyak lagi. Dengan memanfaatkan Big Data, sistem berbasis pengetahuan dapat menggali informasi yang berharga, menemukan pola yang tersembunyi, dan menghasilkan pengetahuan baru yang dapat digunakan untuk pengambilan keputusan yang lebih baik.

Berikut ini adalah beberapa cara penggunaan Big Data dalam sistem berbasis pengetahuan:

1. **Pemrosesan Data yang Skala Besar:** Sistem berbasis pengetahuan dapat memanfaatkan teknologi Big Data untuk mengelola dan menganalisis volume besar data yang dihasilkan. Dengan menggunakan teknik-teknik seperti pemrosesan paralel, komputasi terdistribusi, dan algoritma pengindeksan yang efisien, sistem dapat mengatasi tantangan skalabilitas dalam mengelola Big Data.
2. **Analisis Data yang Mendalam:** Big Data menyediakan sumber data yang beragam dan lengkap. Dengan menerapkan teknik analisis data lanjutan seperti data mining, machine learning, dan analisis prediktif, sistem berbasis pengetahuan dapat menggali wawasan yang lebih mendalam dari data yang ada. Ini memungkinkan sistem

untuk mengidentifikasi pola, tren, dan hubungan yang kompleks dalam data yang dapat digunakan untuk pengambilan keputusan yang lebih cerdas.

3. Rekomendasi yang Personalized: Dengan menggunakan data pengguna yang kaya dan beragam, sistem berbasis pengetahuan dapat memberikan rekomendasi yang lebih personalized. Dalam konteks sistem rekomendasi, penggunaan Big Data memungkinkan sistem untuk mempelajari preferensi pengguna secara mendalam dan menghasilkan rekomendasi yang lebih akurat dan relevan berdasarkan data historis pengguna dan data dari pengguna serupa.
4. Pemodelan yang Lebih Kompleks: Big Data memungkinkan sistem berbasis pengetahuan untuk memodelkan situasi yang lebih kompleks dan realistis. Dengan menggabungkan data yang beragam dan heterogen, sistem dapat membangun model yang lebih komprehensif tentang domain pengetahuan yang sedang dipelajari. Hal ini membantu sistem dalam memahami konteks yang lebih luas dan menghasilkan keputusan yang lebih baik dalam situasi yang kompleks.
5. Deteksi Anomali dan Prediksi: Dengan menggunakan teknik analisis data yang kuat, sistem berbasis pengetahuan dapat menggunakan Big Data untuk mendeteksi anomali, pola yang tidak biasa, atau perubahan yang signifikan dalam data. Ini dapat digunakan untuk tujuan seperti deteksi fraud, prediksi risiko, dan pemantauan real-time. Dengan demikian, sistem dapat memberikan keputusan yang lebih tepat waktu dan mengurangi risiko yang mungkin timbul.

Penggunaan Big Data dalam sistem berbasis pengetahuan memberikan keuntungan besar dalam hal

menggali wawasan yang mendalam, pengambilan keputusan yang lebih cerdas, dan rekomendasi yang lebih personal. Namun, penggunaan Big Data dalam sistem berbasis pengetahuan juga memiliki tantangan dan pertimbangan yang perlu diperhatikan:

1. **Infrastruktur dan Teknologi:** Mengelola dan menganalisis Big Data memerlukan infrastruktur yang kuat dan teknologi yang sesuai. Sistem harus dapat mengakses, menyimpan, dan memproses volume besar data dengan efisien. Pemilihan teknologi seperti sistem manajemen basis data yang sesuai, alat analisis data, dan framework pengolahan data skala besar menjadi penting untuk memastikan penggunaan Big Data yang sukses dalam sistem berbasis pengetahuan.
2. **Kualitas Data:** Big Data sering kali kompleks dan bervariasi dalam kualitas. Tantangan dalam mengatasi kecacatan, ketidakakuratan, dan ketidaklengkapannya menjadi penting. Upaya perlu dilakukan untuk membersihkan dan memperbaiki data agar dapat diandalkan dan akurat dalam proses analisis dan pengambilan keputusan.
3. **Privasi dan Kepatuhan:** Big Data sering kali mencakup data pribadi dan sensitif. Perlindungan privasi pengguna dan kepatuhan terhadap regulasi perlindungan data menjadi hal yang penting dalam penggunaan Big Data. Sistem harus memastikan bahwa data digunakan dengan mematuhi persyaratan privasi yang berlaku dan menerapkan mekanisme keamanan yang tepat untuk melindungi data.
4. **Kompleksitas Analisis:** Big Data sering kali membutuhkan algoritma analisis yang kompleks dan memakan waktu. Proses analisis dapat menjadi rumit dan memerlukan sumber daya komputasi yang signifikan. Pemilihan teknik analisis yang tepat, skalabilitas komputasi, dan optimisasi

kinerja menjadi penting dalam memproses Big Data secara efisien.

5. Interpretasi dan Penggunaan yang Efektif: Mengelola Big Data dalam sistem berbasis pengetahuan memerlukan kemampuan interpretasi yang baik. Data yang besar dan kompleks harus diubah menjadi pengetahuan yang berguna dan dipahami oleh pengguna. Representasi pengetahuan yang tepat dan mekanisme penyampaian informasi yang efektif menjadi kunci dalam memanfaatkan Big Data secara maksimal.

Dengan mempertimbangkan tantangan dan pertimbangan di atas, penggunaan Big Data dalam sistem berbasis pengetahuan dapat memberikan manfaat yang signifikan dalam pengambilan keputusan yang lebih cerdas dan rekomendasi yang lebih personal. Namun, perlu dilakukan pendekatan yang terencana, pengelolaan data yang cermat, dan implementasi teknologi yang tepat untuk mengatasi tantangan yang mungkin timbul dan memastikan penggunaan Big Data yang sukses.

10.5.4 Penerapan sistem berbasis pengetahuan dalam industri dan sektor lainnya

Sistem berbasis pengetahuan telah diterapkan di berbagai industri dan sektor dengan tujuan meningkatkan efisiensi, produktivitas, dan pengambilan keputusan yang lebih baik. Berikut adalah beberapa contoh penerapan sistem berbasis pengetahuan dalam beberapa industri dan sektor:

1. Kesehatan: Dalam industri kesehatan, sistem berbasis pengetahuan dapat digunakan untuk diagnosis medis, perencanaan perawatan, rekomendasi pengobatan, dan pemantauan pasien. Sistem ini dapat menggabungkan pengetahuan medis, data pasien, hasil tes, dan literatur

- medis untuk membantu dokter dan tenaga medis dalam pengambilan keputusan yang lebih akurat.
2. Keuangan: Sistem berbasis pengetahuan dapat digunakan dalam industri keuangan untuk analisis risiko, manajemen investasi, dan pengambilan keputusan dalam transaksi keuangan. Sistem ini dapat mengintegrasikan data pasar keuangan, regulasi, dan pengetahuan ahli dalam menghasilkan rekomendasi investasi yang lebih baik dan mengelola risiko secara efektif.
 3. Manufaktur: Dalam industri manufaktur, sistem berbasis pengetahuan dapat digunakan untuk pemeliharaan mesin, manajemen rantai pasokan, dan perencanaan produksi. Sistem ini dapat memanfaatkan data sensor, data historis, dan pengetahuan ahli untuk mendeteksi kerusakan pada mesin, mengoptimalkan jadwal produksi, dan meningkatkan efisiensi operasional.
 4. Perdagangan Elektronik: Dalam sektor perdagangan elektronik, sistem berbasis pengetahuan dapat digunakan untuk sistem rekomendasi, personalisasi pengalaman pengguna, dan deteksi penipuan. Sistem ini dapat menganalisis data perilaku pengguna, preferensi pembelian, dan data transaksi untuk memberikan rekomendasi produk yang relevan, meningkatkan konversi penjualan, dan mengurangi risiko penipuan.
 5. Transportasi dan Logistik: Dalam industri transportasi dan logistik, sistem berbasis pengetahuan dapat digunakan untuk perencanaan rute, manajemen pengiriman, dan optimasi jadwal. Sistem ini dapat mengintegrasikan data lalu lintas, data cuaca, data inventaris, dan pengetahuan logistik untuk mengoptimalkan pengiriman barang, mengurangi biaya, dan meningkatkan efisiensi.
 6. Pendidikan: Sistem berbasis pengetahuan juga dapat diterapkan dalam sektor pendidikan untuk menyediakan

dukungan pembelajaran yang personal dan rekomendasi kurikulum. Sistem ini dapat menganalisis data kinerja siswa, preferensi belajar, dan kurikulum yang tersedia untuk memberikan rekomendasi pembelajaran yang sesuai dengan kebutuhan individu.

7. Layanan Pelanggan: Sistem berbasis pengetahuan dapat digunakan dalam industri layanan pelanggan untuk memberikan dukungan pelanggan yang lebih baik. Sistem ini dapat mengintegrasikan basis pengetahuan tentang produk, kebijakan perusahaan, dan solusi masalah umum untuk memberikan jawaban cepat dan akurat kepada pelanggan melalui pusat kontak, live chat, atau platform self-service.
8. Manajemen Sumber Daya Manusia: Dalam pengelolaan sumber daya manusia, sistem berbasis pengetahuan dapat digunakan untuk memfasilitasi proses rekrutmen, pengembangan karyawan, dan manajemen pengetahuan. Sistem ini dapat menggabungkan pengetahuan tentang keahlian yang diperlukan, pengalaman kerja, dan preferensi karyawan untuk memfasilitasi pengambilan keputusan terkait perekrutan, pelatihan, dan penempatan karyawan.
9. Sistem Pakar dalam Bidang Teknis: Sistem berbasis pengetahuan juga banyak digunakan dalam bidang teknis, seperti bidang perawatan mesin industri, jaringan komputer, dan teknologi informasi. Sistem ini dapat digunakan untuk melakukan diagnosis masalah, perbaikan, dan pemeliharaan dengan memanfaatkan pengetahuan ahli dan data sensor.
10. Pengenalan Pola dan Deteksi Anomali: Sistem berbasis pengetahuan dapat digunakan untuk pengenalan pola dan deteksi anomali dalam berbagai aplikasi, seperti keamanan jaringan, kecurangan keuangan, dan pengawasan lalu

- lintas. Sistem ini dapat menggunakan pengetahuan tentang pola yang terdefinisi dan aturan logika untuk mengidentifikasi perilaku yang mencurigakan atau anomali yang tidak biasa.
11. Penelitian dan Pengembangan: Dalam bidang penelitian dan pengembangan, sistem berbasis pengetahuan dapat digunakan untuk mengelola dan mengintegrasikan pengetahuan ilmiah, literatur, dan data eksperimental. Sistem ini dapat membantu peneliti dalam menganalisis data, membangun model prediktif, dan mengidentifikasi tren atau pola yang penting dalam penelitian.
 12. Sistem Berbasis Pengetahuan untuk Pengambilan Keputusan Strategis: Sistem berbasis pengetahuan juga dapat digunakan untuk pengambilan keputusan strategis dalam organisasi, seperti perencanaan bisnis, analisis pasar, dan pengembangan strategi. Sistem ini dapat mengintegrasikan pengetahuan bisnis, data pasar, dan alat analisis untuk memberikan wawasan yang lebih baik kepada manajemen dalam pengambilan keputusan yang berkaitan dengan pertumbuhan dan strategi perusahaan.

Ini hanya beberapa contoh penerapan sistem berbasis pengetahuan dalam industri dan sektor lainnya. Dalam praktiknya, sistem berbasis pengetahuan dapat diadaptasi dan diterapkan dalam berbagai domain untuk memecahkan masalah khusus dan meningkatkan kinerja bisnis atau layanan. Penerapan sistem berbasis pengetahuan dalam berbagai industri dan sektor ini memberikan manfaat seperti peningkatan efisiensi, pengurangan kesalahan, pemecahan masalah yang lebih cepat, dan pengambilan keputusan yang lebih akurat. Dalam setiap aplikasi, sistem berbasis pengetahuan dirancang dan dikembangkan sesuai dengan

kebutuhan dan karakteristik unik dari industri atau sektor yang bersangkutan.

DAFTAR PUSTAKA

- Brachman, R.J., & Levesque, H.J. 2004. Knowledge Representation and Reasoning. San Francisco, CA: Morgan Kaufmann Publishers.
- Buchanan, B.G., & Shortliffe, E.H. 1984. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Reading, MA: Addison-Wesley.
- Budiharto, W. 2009. Sistem Berbasis Pengetahuan: Teori dan Aplikasi. Yogyakarta: Penerbit Andi.
- Budiharto, W., & Marzuki, I. 2009. Sistem Pakar dan Sistem Berbasis Pengetahuan. Yogyakarta: Penerbit Andi.
- Chaudhry, A.S. 2015. Expert Systems: Principles and Programming. New Delhi, India: PHI Learning Pvt. Ltd.
- Dewi, A. I., & Andri, R. 2015. Sistem Pakar dan Sistem Berbasis Pengetahuan. Yogyakarta: Penerbit Deepublish.
- Giarratano, J., & Riley, G. 2004. Expert Systems: Principles and Programming. Boston, MA: Course Technology.
- Hayes-Roth, B., & Waterman, D.A. 1990. Expert Systems: Introduction to First and Second Generation and Hybrid Knowledge-Based Systems. Reading, MA: Addison-Wesley.
- Jackson, P. 1999. Introduction to Expert Systems. Harlow, England: Addison-Wesley.
- Jackson, P., & Shoham, Y. 1994. Machine Learning and the Design of Expert Systems. *AI Magazine*, 15(1), 37-54.
- Nurdin, E., & Prasetyo, T. 2014. Sistem Pakar dan Sistem Berbasis Pengetahuan. Jakarta: Penerbit PT Rajagrafindo Persada.
- Nurhadi, E. 2010. Sistem Berbasis Pengetahuan: Konsep dan Implementasi. Bandung: Penerbit Informatika.

- Nwana, H.S. 1996. Software Agents: An Overview. *Knowledge Engineering Review*, 11(3), 205-244.
- Patel, M., & Patel, K. 2017. Expert Systems: Applications and Challenges. *International Journal of Computer Science and Mobile Computing*, 6(1), 101-107.
- Russell, S., & Norvig, P. 2016. *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Pearson Education.
- Simanjuntak, M. 2014. *Sistem Berbasis Pengetahuan: Konsep dan Implementasi*. Jakarta: Penerbit PT Gramedia Pustaka Utama.
- Simanjuntak, M., & Arief, I. 2017. *Sistem Pakar Berbasis Pengetahuan*. Jakarta: Penerbit PT Elex Media Komputindo.
- Sinaga, B. 2018. *Sistem Berbasis Pengetahuan*. Yogyakarta: Penerbit Andi.

BAB 11

JARINGAN SYARAF TIRUAN

Oleh Angga Aditya Permana

11.1 Pendahuluan

Algoritma jaringan syaraf tiruan (JST) adalah serangkaian langkah dan aturan matematis yang digunakan untuk melatih dan mengoperasikan JST. JST adalah model matematis yang terinspirasi oleh sistem saraf biologis, dan dapat digunakan untuk pemrosesan informasi dan pembelajaran mesin. (Jain, Mao and Mohiuddin, 1996)

Berikut adalah penjelasan singkat tentang algoritma JST:

1. Inisialisasi: Bobot-bobot dan bias JST diinisialisasi dengan nilai acak kecil. Bobot dan bias ini mewakili kekuatan dan ambang batas koneksi antara neuron.
2. Tahap maju (*forward propagation*): Data input diberikan ke JST, dan sinyal input diteruskan melalui lapisan-lapisan JST. Setiap neuron di lapisan input meneruskan sinyal ke neuron-neuron di lapisan berikutnya, hingga mencapai lapisan output. Setiap neuron menggunakan fungsi aktivasi untuk menghasilkan outputnya berdasarkan input dan bobotnya.
3. Perhitungan kesalahan: Output yang dihasilkan oleh JST dibandingkan dengan output yang diharapkan (target) untuk menghitung kesalahan prediksi. Metrik kesalahan yang umum digunakan adalah *mean squared error* (MSE) atau *cross-entropy loss*.
4. Tahap mundur (*backward propagation*): Kesalahan dikembalikan ke belakang melalui JST menggunakan algoritma backpropagation. Bobot-bobot dan bias diubah

berdasarkan kontribusi mereka terhadap kesalahan prediksi. Gradien kesalahan dihitung untuk setiap bobot dengan menggunakan aturan rantai (*chain rule*) dalam kalkulus.

5. Pembaruan bobot: Bobot-bobot dan bias diperbarui menggunakan metode penurunan gradien stokastik (*stochastic gradient descent*) atau variasinya. Langkah pembelajaran (*learning rate*) digunakan untuk mengontrol sejauh mana bobot diperbarui setiap iterasi. Selain itu, teknik optimisasi seperti momentum dan regularisasi dapat digunakan untuk meningkatkan pelatihan dan mencegah overfitting.
6. Iterasi dan konvergensi: Langkah-langkah 2-5 diulang secara iteratif untuk seluruh data pelatihan. Siklus iterasi ini disebut epoch. Proses ini berlanjut hingga JST mencapai kriteria penghentian yang ditentukan, seperti mencapai tingkat kesalahan yang cukup rendah atau mencapai jumlah epoch yang ditentukan.

Proses pelatihan JST dapat memakan waktu dan sumber daya komputasi yang signifikan. Oleh karena itu, teknik-teknik seperti mini-batch gradient descent, penghentian dini (*early stopping*), dan pengoptimalan parameter lainnya dapat digunakan untuk meningkatkan efisiensi dan kinerja JST.

Setelah pelatihan, JST dapat digunakan untuk melakukan prediksi atau klasifikasi pada data baru dengan meneruskan sinyal maju melalui jaringan yang telah dilatih.

Secara detail, algoritma jaringan syaraf tiruan (JST) terdiri dari beberapa komponen dan langkah-langkah yang perlu diimplementasikan oleh seorang programmer. Berikut adalah penjelasan algoritma JST dari sudut pandang seorang programmer:

1. Inisialisasi:
 - a. Tentukan jumlah lapisan (*layers*) yang akan digunakan dalam JST, termasuk lapisan input, lapisan tersembunyi (*hidden layers*), dan lapisan output.
 - b. Tetapkan jumlah neuron dalam setiap lapisan, serta fungsi aktivasi yang akan digunakan oleh masing-masing neuron.
 - c. Inisialisasi bobot dan bias secara acak untuk setiap koneksi antar neuron dalam JST.
2. Propagasi maju (*Forward Propagation*):
 - a. Ambil input dari data pelatihan dan teruskan ke lapisan input JST.
 - b. Hitung keluaran (*output*) dari setiap neuron dalam lapisan tersembunyi dan lapisan output dengan menggunakan bobot dan bias yang telah diinisialisasi sebelumnya.
 - c. Terapkan fungsi aktivasi pada setiap neuron untuk menghasilkan keluaran yang diperoleh.
3. Perhitungan kesalahan (*Error Calculation*):
 - a. Bandingkan keluaran aktual yang dihasilkan oleh JST dengan nilai target yang diharapkan dari data pelatihan.
 - b. Hitung selisih antara keluaran aktual dan nilai target untuk mengukur kesalahan (*error*).
4. Propagasi balik (*Backpropagation*):
 - a. Gunakan metode gradien untuk menghitung gradien (turunan parsial) kesalahan terhadap bobot dan bias di setiap neuron dalam JST.

- b. Sesuaikan bobot dan bias dengan menggunakan gradien tersebut untuk meminimalkan kesalahan secara iteratif.
 - c. Lanjutkan langkah ini dengan menghitung gradien dan memperbarui bobot dan bias sampai kesalahan mencapai tingkat yang dapat diterima.
5. Evaluasi performa:
- a. Gunakan data validasi yang tidak terlibat dalam pelatihan untuk mengukur performa JST.
 - b. Hitung metrik evaluasi seperti akurasi, presisi, recall, atau *mean squared error* (MSE) untuk mengevaluasi kualitas prediksi JST.
 - c. Ulangi langkah-langkah sebelumnya dengan memperbarui parameter JST jika performa tidak memenuhi harapan.
6. Pengujian:
- a. Setelah JST dilatih dan dinilai cukup baik, gunakan JST untuk melakukan prediksi pada data yang belum pernah dilihat sebelumnya.
 - b. Berikan input baru ke JST dan gunakan propagasi maju untuk menghasilkan prediksi keluaran.
7. Penyetelan parameter (*Hyperparameter Tuning*):
- a. Selama proses pengembangan JST, eksperimen dengan berbagai hyperparameter seperti jumlah lapisan, jumlah neuron, fungsi aktivasi, learning rate, dan metode inisialisasi bobot.
 - b. Cari kombinasi parameter yang menghasilkan performa terbaik pada data pelatihan dan validasi.

8. Regularisasi:

- a. Untuk menghindari overfitting, terapkan teknik regularisasi seperti L1 atau L2 *regularization*, *dropout*, atau *early stopping*.
- b. Regularisasi bertujuan untuk mengontrol kompleksitas model JST dan mencegah *overfitting*.
- c. L1 dan L2 *regularization* mengenakan biaya tambahan pada fungsi kesalahan berdasarkan norma L1 atau L2 dari bobot, sehingga mendorong bobot yang tidak signifikan atau redundan untuk menjadi nol atau mendekati nol.
- d. *Dropout* adalah teknik yang secara acak mengabaikan sebagian neuron dalam JST selama pelatihan, yang mengurangi ketergantungan pada neuron tertentu dan mendorong pembelajaran yang lebih robust.
- e. *Early stopping* adalah teknik yang menghentikan pelatihan JST lebih awal jika tidak ada peningkatan dalam performa validasi untuk menghindari overfitting.

9. Pelatihan berulang:

- a. Lakukan iterasi melalui langkah-langkah 2-7 di atas untuk melatih JST secara berulang.
- b. Saat melatih JST dengan data pelatihan, ulangi langkah-langkah propagasi maju, perhitungan kesalahan, propagasi balik, dan evaluasi performa untuk memperbaiki bobot dan bias.
- c. Lakukan iterasi ini dengan jumlah epoch yang telah ditentukan atau sampai kesalahan mencapai tingkat yang dapat diterima.

10. Validasi silang:

- a. Untuk mengukur performa JST secara obyektif, gunakan metode validasi silang (*cross-validation*) yang membagi data pelatihan menjadi subset pelatihan dan validasi.
- b. Latih JST pada subset pelatihan dan evaluasi performanya pada subset validasi.
- c. Ulangi proses ini beberapa kali dengan memvariasikan subset pelatihan dan validasi untuk mendapatkan estimasi performa yang lebih stabil.

11. Pengujian ulang:

- a. Setelah JST dilatih dengan baik, gunakan data pengujian yang terpisah dan belum pernah dilihat sebelumnya untuk menguji performa JST yang telah dilatih.
- b. Gunakan propagasi maju untuk menghasilkan prediksi JST pada data pengujian.
- c. Evaluasi performa JST berdasarkan metrik yang relevan, seperti akurasi, presisi, recall, F1-score, atau metrik khusus yang sesuai dengan masalah yang sedang diselesaikan.

12. Optimisasi dan penyempurnaan:

- a. Terus eksplorasi dan eksperimen dengan berbagai teknik, hyperparameter, atau arsitektur JST untuk meningkatkan performa dan kualitas prediksi.
- b. Perhatikan feedback dari evaluasi performa dan lakukan iterasi dalam pengembangan dan penyempurnaan JST.

Dalam prakteknya, implementasi algoritma JST dapat menggunakan berbagai library atau framework yang telah

tersedia, seperti TensorFlow, Keras, PyTorch, atau scikit-learn, yang menyediakan fungsi dan utilitas untuk memudahkan pembuatan, pelatihan, dan evaluasi JST.

11.2 Sejarah Jaringan Syaraf Tiruan

Sejarah algoritma Jaringan Saraf Tiruan (JST) dimulai pada tahun 1943 dengan model "neuron" sederhana yang dikembangkan oleh Warren McCulloch dan Walter Pitts. Model ini mencoba mensimulasikan cara kerja neuron dalam otak manusia. Pada model ini, neuron dikonseptualisasikan sebagai entitas biner yang menerima masukan, menghitung nilai ambang batas, dan menghasilkan keluaran.

Pada tahun 1950-an hingga 1960-an, konsep dasar JST diperluas oleh beberapa peneliti seperti Frank Rosenblatt dan Bernard Widrow. Rosenblatt mengembangkan model jaringan saraf tiruan yang dikenal sebagai Perceptron, yang mampu mempelajari pola-pola dari data dan melakukan klasifikasi. Perceptron menggunakan metode pelatihan yang disebut "learning rule" untuk menyesuaikan bobot dan mengoptimalkan model.

Pada tahun 1974, John Hopfield mengenalkan jaringan saraf Hopfield, yang merupakan jenis jaringan saraf tiruan yang menggunakan pendekatan yang berbeda dengan memanfaatkan konsep energi untuk memodelkan proses pemrosesan informasi.

Pada tahun 1986, seorang ilmuwan komputer bernama Geoffrey Hinton dan rekan-rekannya mengembangkan model jaringan saraf tiruan yang dikenal sebagai jaringan saraf tersembunyi (*multilayer neural network*) dengan algoritma pelatihan yang disebut backpropagation. Backpropagation memungkinkan JST untuk melatih model dengan banyak lapisan dan memperoleh tingkat akurasi yang lebih tinggi

dalam tugas-tugas seperti klasifikasi gambar dan pengenalan pola.

Pada tahun 1990-an, dukungan komputasi yang lebih kuat dan kemajuan dalam teknologi memungkinkan pengembangan JST yang lebih kompleks. Arsitektur seperti Jaringan Saraf Konvolusi (*Convolutional Neural Network/CNN*) diperkenalkan oleh Yann LeCun dan timnya pada tahun 1998, yang sangat efektif dalam memproses data gambar dan diterapkan dalam berbagai aplikasi pengenalan gambar.

Pada tahun 2000-an dan seterusnya, JST semakin populer dan digunakan secara luas dalam berbagai aplikasi seperti deteksi wajah, sistem rekomendasi, pengenalan suara, dan pemrosesan bahasa alami. Kemajuan dalam teknologi komputasi, peningkatan ketersediaan data, dan pengembangan algoritma yang lebih canggih seperti jaringan saraf rekurensi (*recurrent neural network/RNN*) dan jaringan saraf generatif (*generative neural network*) terus memperluas potensi dan kemampuan JST.

Sejak itu, penggunaan JST terus berkembang dengan cepat, termasuk penggunaan arsitektur yang lebih maju seperti Transformer dan pengembangan teknik optimasi yang lebih efisien seperti *stochastic gradient descent* (SGD) dan *adaptive learning rate*. JST juga mendapatkan perhatian besar dalam perkembangan kecerdasan buatan (*Artificial Intelligence/AI*) dan telah menjadi salah satu bidang penelitian utama dalam komunitas ilmu komputer dan teknologi informasi.

Berikut adalah penjelasan yang lebih rinci tentang sejarah algoritma Jaringan Saraf Tiruan (JST): (Ahamed and Akthar, 2016)

1. McCulloch-Pitts Neuron (1943): Warren McCulloch dan Walter Pitts mengusulkan model "neuron" sederhana yang terinspirasi dari cara kerja neuron dalam otak manusia. Model ini berupa entitas biner yang menerima masukan,

- menghitung nilai ambang batas, dan menghasilkan keluaran. Model ini menjadi dasar bagi pengembangan JST di masa depan.
2. Perceptron (1957): Frank Rosenblatt mengembangkan model jaringan saraf tiruan yang dikenal sebagai Perceptron. Perceptron adalah jaringan saraf tiruan dengan satu atau beberapa lapisan yang mampu mempelajari pola-pola dari data dan melakukan klasifikasi. Rosenblatt juga memperkenalkan algoritma pembelajaran yang disebut "learning rule" untuk menyesuaikan bobot dalam jaringan Perceptron.
 3. Jaringan Saraf Hopfield (1982): John Hopfield mengembangkan jaringan saraf Hopfield, yang menggunakan konsep energi dalam pemodelan pemrosesan informasi. Jaringan ini menggunakan pendekatan asosiasi untuk mengingat pola dan mengatasi masalah optimasi dengan mencapai titik tertentu dalam ruang solusi. Jaringan Hopfield juga dapat melakukan pemulihan data yang rusak atau tidak lengkap.
 4. Jaringan Saraf Tersembunyi dan Backpropagation (1986): Geoffrey Hinton, David Rumelhart, dan Ronald Williams memperkenalkan jaringan saraf tersembunyi, yang juga dikenal sebagai multilayer neural network. Mereka juga mengembangkan algoritma pelatihan yang disebut backpropagation. Backpropagation memungkinkan JST dengan banyak lapisan untuk dilatih dengan efisien dan memperoleh tingkat akurasi yang lebih tinggi dalam tugas-tugas seperti klasifikasi gambar dan pengenalan pola.
 5. Pengembangan Arsitektur Lanjutan: Pada tahun 1990-an, arsitektur JST semakin berkembang dengan adanya arsitektur seperti Jaringan Saraf Konvolusi (CNN) yang diperkenalkan oleh Yann LeCun dan timnya pada tahun 1998. CNN sangat efektif dalam memproses data gambar

dan telah diterapkan dalam berbagai aplikasi pengenalan gambar. Selain itu, jaringan saraf rekurensi (RNN) juga dikembangkan untuk memodelkan data sekuensial dan telah digunakan dalam pemrosesan bahasa alami dan tugas-tugas berurutan lainnya.

6. Kemajuan Komputasi dan Data yang Tersedia: Seiring dengan kemajuan teknologi komputasi dan peningkatan ketersediaan data, JST semakin populer dan digunakan secara luas dalam berbagai aplikasi. Perkembangan komputasi paralel dan GPU memungkinkan pelatihan dan penggunaan JST dalam skala yang lebih besar dan lebih cepat. Selain itu, perkembangan internet dan teknologi digital telah menghasilkan ledakan data yang dapat digunakan untuk melatih dan memvalidasi JST.
7. Pencapaian Terkini: Dalam dekade terakhir, terdapat beberapa pencapaian signifikan dalam pengembangan dan penerapan algoritma Jaringan Saraf Tiruan (JST). Beberapa perkembangan terkini termasuk:
 - a. Jaringan Saraf Konvolusi yang Semakin Mendalam: Penggunaan arsitektur Jaringan Saraf Konvolusi (CNN) semakin mendalam dengan memperluas jumlah lapisan konvolusi dan penggunaan teknik seperti residual connections. Ini telah menghasilkan peningkatan yang signifikan dalam kinerja pengenalan gambar dan tugas-tugas visi komputer lainnya.
 - b. Jaringan Saraf Rekurensi yang Lebih Canggih: Penggunaan jaringan saraf rekurensi (RNN) telah berkembang dengan munculnya variasi seperti Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU). RNN yang lebih canggih ini mampu mengatasi masalah vanishing gradient dan memodelkan hubungan jangka panjang dalam data sekuensial.

- c. **Arsitektur Transformer untuk Pemrosesan Bahasa:** Arsitektur Transformer, yang diperkenalkan pada tahun 2017, telah merevolusi pemrosesan bahasa alami. Transformer menggunakan mekanisme perhatian (*attention mechanism*) untuk mengatasi masalah ketergantungan jarak panjang dan menghasilkan hasil yang sangat baik dalam tugas-tugas seperti pemodelan bahasa, penerjemahan mesin, dan generasi teks.
- d. **Jaringan Generatif yang Lanjutan:** Jaringan saraf generatif (GAN) adalah jenis JST yang mampu menghasilkan data baru yang sangat realistis. GAN telah digunakan dalam berbagai aplikasi seperti pembuatan gambar sintetis, penggabungan wajah, dan pembangkitan konten kreatif.
- e. **JST dalam Pembelajaran Penguatan:** JST juga digunakan dalam konteks pembelajaran penguatan (*reinforcement learning*), di mana agen belajar untuk mengambil tindakan dalam lingkungan yang dinamis untuk mencapai tujuan tertentu. Dengan menggunakan kombinasi JST dan algoritma pembelajaran penguatan, telah dicapai keberhasilan yang menonjol dalam permainan strategi dan kontrol robotik kompleks.
- f. **Implementasi Perceptron dalam Perangkat Keras:** Perkembangan teknologi perangkat keras khusus untuk JST, seperti unit pemrosesan grafis (GPU) dan *Tensor Processing Units* (TPU), telah memungkinkan pelatihan dan implementasi JST yang lebih cepat dan efisien. Implementasi perangkat keras yang dioptimalkan secara khusus untuk JST membantu mempercepat pemrosesan dan memungkinkan aplikasi JST *real-time* yang lebih luas.

Pencapaian terkini dalam JST terus berkembang seiring dengan kemajuan teknologi dan penelitian yang terus dilakukan dalam bidang kecerdasan buatan. JST telah menjadi bagian integral dari banyak aplikasi AI dan terus digunakan untuk memecahkan masalah kompleks dalam berbagai bidang, mulai dari visi komputer hingga pemrosesan bahasa alami dan robotika.

11.3 Kelebihan Jaringan Syaraf Tiruan

Algoritma jaringan saraf tiruan (JST) memiliki beberapa kelebihan yang menjadikannya populer dalam aplikasi pemrosesan data dan pembelajaran mesin. Berikut adalah beberapa kelebihan utama JST:

1. Kemampuan pemrosesan data non-linear: JST dapat memodelkan hubungan dan pola yang kompleks antara input dan output. Mereka mampu menangani masalah pemrosesan data non-linear yang sulit atau tidak mungkin diselesaikan dengan pendekatan linier tradisional. Hal ini membuat JST sangat efektif dalam pemrosesan data seperti pengenalan pola, pengenalan wajah, penerjemahan bahasa, dan sebagainya.
2. Pembelajaran adaptif: JST mampu belajar dari data dan mengadaptasi diri terhadap perubahan dalam lingkungan atau tugas yang diberikan. Mereka dapat mengenali dan menyesuaikan pola-pola baru tanpa perlu melakukan pemrograman ulang secara eksplisit. Ini membuat JST fleksibel dan dapat diaplikasikan dalam berbagai domain dan aplikasi yang dinamis.
3. Kemampuan generalisasi: JST memiliki kemampuan untuk menggeneralisasi pola yang dipelajari dari data pelatihan ke data yang belum pernah dilihat sebelumnya. Mereka dapat mengenali pola umum dan menciptakan model prediksi atau klasifikasi yang dapat diterapkan pada situasi baru.

Kemampuan ini sangat berguna dalam aplikasi di mana data yang tersedia terbatas atau tidak lengkap.

4. Tangkapan fitur otomatis: JST dapat secara otomatis mengekstraksi fitur-fitur yang relevan dari data input tanpa memerlukan pemodelan manual yang rumit. Ini memungkinkan JST untuk mengenali pola-pola penting dalam data dengan cara yang lebih efisien dan efektif.
5. Kecepatan pemrosesan paralel: JST dapat diimplementasikan dalam arsitektur paralel yang memungkinkan pemrosesan yang cepat dan efisien. Hal ini terutama penting dalam aplikasi yang membutuhkan pemrosesan real-time atau pemrosesan data yang besar.
6. Toleransi terhadap kesalahan dan kerusakan: JST memiliki tingkat toleransi terhadap kesalahan dan kerusakan yang cukup tinggi. Mereka dapat terus beroperasi dengan baik meskipun beberapa neuron atau koneksi mengalami kerusakan. Ini membuat JST lebih tangguh dalam menghadapi gangguan atau kegagalan perangkat keras.
7. Adaptabilitas pada masalah besar dan kompleks: JST dapat digunakan untuk menangani masalah yang melibatkan jumlah variabel yang besar dan kompleksitas tinggi. Mereka dapat digunakan dalam pemrosesan data multidimensional dan tugas-tugas yang melibatkan berbagai jenis input.
8. Pembelajaran tanpa pengawasan: JST juga dapat digunakan untuk pembelajaran tanpa pengawasan, di mana tidak ada label atau target yang tersedia. Dalam hal ini, JST dapat menemukan pola tersembunyi dalam data tanpa panduan eksternal. Ini berguna dalam eksplorasi data, pengelompokan (*clustering*), dan analisis asosiasi.
9. Kemampuan memproses data multimedia: JST dapat digunakan untuk memproses data multimedia seperti gambar, suara, dan video. Mereka dapat dilatih untuk

mengenali objek, wajah, karakteristik audio, dan berbagai fitur visual lainnya. Ini menjadikan JST sebagai alat yang kuat dalam bidang pengolahan citra, pengenalan suara, dan analisis video.

10. **Fleksibilitas arsitektur:** JST memiliki arsitektur yang fleksibel dan dapat disesuaikan dengan kebutuhan spesifik tugas. Mereka dapat dibangun dengan berbagai lapisan, termasuk lapisan tersembunyi (*hidden layers*) yang memungkinkan pemodelan yang lebih kompleks dan penanganan masalah yang lebih sulit. Selain itu, ada berbagai jenis JST seperti jaringan saraf konvolusional (CNN) untuk pengolahan citra dan jaringan saraf rekurens (RNN) untuk pemrosesan data berurutan.
11. **Interpretabilitas:** Meskipun JST sering dikenal sebagai "kotak hitam" karena kompleksitasnya, ada upaya yang dilakukan untuk meningkatkan interpretabilitas JST. Misalnya, teknik seperti visualisasi saluran gradien (*gradient channels*) dan atribusi fitur (*feature attribution*) digunakan untuk memahami kontribusi masing-masing fitur dalam pengambilan keputusan JST. Ini memungkinkan pemahaman yang lebih baik tentang alasan di balik prediksi atau keputusan yang dihasilkan oleh JST.
12. **Implementasi perangkat keras:** JST dapat diimplementasikan dalam perangkat keras khusus seperti prosesor neural atau rangkaian FPGA (*Field-Programmable Gate Array*). Implementasi perangkat keras dapat memberikan kecepatan dan efisiensi yang lebih tinggi dalam pemrosesan JST, serta memungkinkan aplikasi real-time yang lebih cepat.

Dalam kesimpulannya, algoritma jaringan saraf tiruan memiliki sejumlah kelebihan yang menjadikannya alat yang kuat dalam pemrosesan data dan pembelajaran mesin. Mereka

dapat mengatasi masalah non-linear, belajar adaptif, menggeneralisasi pola, menangkap fitur otomatis, dan bekerja secara paralel. Namun, pemilihan yang tepat dari arsitektur, pengaturan parameter, dan interpretasi hasil tetap merupakan tantangan yang perlu diperhatikan dalam penggunaan JST. Selain itu, penting untuk dicatat bahwa JST juga memiliki beberapa keterbatasan, seperti kecenderungan terhadap overfitting jika tidak dikendalikan dengan baik, kebutuhan akan jumlah data pelatihan yang cukup besar, dan kesulitan dalam interpretasi hasil yang kompleks.

Meskipun demikian, dengan kemajuan teknologi dan penelitian yang terus berlanjut, algoritma jaringan saraf tiruan terus berkembang dan diterapkan dalam berbagai bidang. Mereka telah menghasilkan kemajuan signifikan dalam pengenalan pola, pengolahan bahasa alami, penglihatan komputer, kendaraan otonom, dan banyak lagi. Kelebihan-kelebihan yang telah disebutkan membuat JST menjadi alat yang sangat berharga dalam analisis data dan pengambilan keputusan yang kompleks.

Namun, penting juga untuk mempertimbangkan faktor-faktor seperti waktu komputasi, kompleksitas model, dan ketersediaan data yang memadai saat memilih dan menerapkan JST. Setiap aplikasi memiliki kebutuhan dan tantangan tersendiri, dan penting untuk mempertimbangkan faktor-faktor tersebut dengan hati-hati.

Dalam kesimpulannya, algoritma jaringan saraf tiruan memberikan kemampuan yang kuat untuk pemrosesan data non-linear, pembelajaran adaptif, generalisasi pola, dan pengolahan data multimedia. Dengan kemampuan ini, JST telah membantu memecahkan masalah yang kompleks dan menghasilkan inovasi di berbagai bidang. Namun, penggunaan JST harus didasarkan pada pemahaman yang baik tentang

kelebihan dan keterbatasannya, serta pemilihan yang tepat dalam merancang dan melatih model JST.

11.4 Kekurangan Jaringan Syaraf Tiruan

Selain kelebihan-kelebihan yang telah disebutkan sebelumnya, algoritma jaringan saraf tiruan (JST) juga memiliki beberapa kekurangan yang perlu diperhatikan. Berikut adalah beberapa kekurangan umum dari JST:

1. Kompleksitas pemodelan: JST dapat menjadi sangat kompleks dalam hal arsitektur dan parameter. Terkadang sulit untuk menentukan jumlah lapisan dan neuron yang tepat, serta parameter-parameter lainnya seperti learning rate atau momentum. Kekurangan ini dapat menyebabkan kesulitan dalam perancangan dan tuning model JST yang optimal.
2. Ketergantungan pada jumlah data pelatihan: JST membutuhkan jumlah data pelatihan yang cukup besar untuk mencapai performa yang baik. Jika data pelatihan terbatas atau tidak representatif, JST dapat mengalami overfitting, di mana model terlalu mempelajari pola-pola pada data pelatihan dan gagal melakukan generalisasi pada data baru. Kekurangan data pelatihan yang cukup dapat mengurangi performa dan kualitas model JST.
3. Waktu pelatihan yang lama: Pelatihan JST dapat memakan waktu yang signifikan, terutama jika dataset besar atau model JST kompleks. Proses backpropagation yang digunakan dalam pelatihan JST memerlukan perhitungan gradien yang intensif dan mengharuskan iterasi berulang-ulang melalui dataset. Ini dapat menjadi kendala dalam aplikasi yang membutuhkan respons real-time atau pemrosesan cepat.
4. Tidak ada interpretabilitas yang jelas: JST seringkali dianggap sebagai "kotak hitam" karena kurangnya

- interpretabilitas yang jelas. Artinya, sulit untuk memahami dengan jelas mengapa JST membuat keputusan atau prediksi tertentu. Ini menjadi tantangan dalam situasi di mana transparansi dan penjelasan yang baik diperlukan, seperti dalam pengambilan keputusan yang kritis atau di bidang medis.
5. Sensitivitas terhadap inisialisasi bobot: Hasil pelatihan JST dapat dipengaruhi oleh inisialisasi bobot awal yang dilakukan secara acak. Jika inisialisasi bobot tidak tepat, JST dapat terjebak dalam minimum lokal dan tidak mencapai performa yang optimal. Oleh karena itu, pemilihan inisialisasi bobot yang baik menjadi penting dalam pelatihan JST.
 6. Kerentanan terhadap gangguan dan kebisingan: JST rentan terhadap gangguan dan kebisingan pada data input. Bahkan gangguan kecil dapat mempengaruhi kinerja JST dan menghasilkan prediksi yang tidak akurat. JST juga dapat mengambil keputusan yang tidak stabil jika diberikan input yang sedikit berbeda. Hal ini menjadi tantangan dalam aplikasi di mana keandalan dan ketahanan terhadap gangguan sangat penting.
 7. Ketergantungan pada pemilihan arsitektur yang tepat: Performa JST sangat tergantung pada pemilihan arsitektur yang tepat untuk tugas yang diberikan. Arsitektur yang tidak cocok dapat menghasilkan hasil yang buruk atau tidak memadai.
 8. Keterbatasan data yang tidak seimbang: Jika data pelatihan tidak seimbang, yaitu jika beberapa kelas atau kategori memiliki jumlah sampel yang sangat sedikit dibandingkan dengan yang lain, JST cenderung memberikan prediksi yang bias terhadap kelas mayoritas. Ini dapat mengurangi performa dan kualitas prediksi JST pada kelas yang kurang representatif.

9. Kesulitan dalam interpretasi fitur: Meskipun JST dapat mengekstraksi fitur secara otomatis, mereka tidak memberikan interpretasi yang jelas tentang fitur-fitur tersebut. JST mungkin mengidentifikasi fitur yang relevan, tetapi tidak memberikan pemahaman yang rinci tentang alasan di balik pentingnya fitur-fitur tersebut. Ini dapat menjadi hambatan dalam pemahaman domain spesifik dan interpretasi hasil JST.
10. Ketergantungan pada data yang berkualitas tinggi: JST cenderung memberikan hasil yang baik hanya jika data input berkualitas tinggi dan bebas dari kesalahan atau kebisingan yang signifikan. Ketika data input tidak berkualitas tinggi atau terdapat gangguan, kinerja JST dapat terpengaruh secara negatif. Oleh karena itu, prapemrosesan data yang baik dan penghilangan outlier menjadi penting dalam penggunaan JST.
11. Kelemahan dalam menghadapi masalah skala besar: Meskipun JST dapat mengatasi masalah yang kompleks, mereka mungkin menghadapi tantangan dalam skala besar. Ketika jumlah input atau dimensi ruang fitur sangat besar, JST dapat mengalami kesulitan dalam pemodelan yang efisien dan waktu komputasi yang cepat.
12. Kekurangan teori matematis yang jelas: Meskipun JST telah terbukti efektif dalam banyak tugas, terdapat kekurangan teori matematis yang lengkap untuk memahami dan menjelaskan perilaku JST. Beberapa aspek JST, seperti pemilihan arsitektur yang optimal atau kecenderungan untuk mengalami overfitting, masih menjadi area penelitian aktif.

Meskipun JST memiliki beberapa kekurangan, banyak dari kelemahan ini dapat diatasi atau dikurangi dengan pemilihan yang tepat dari arsitektur JST, pengaturan parameter

yang baik, dan perhatian terhadap kualitas data pelatihan. JST tetap menjadi algoritma yang kuat dan berguna dalam pemrosesan data dan pembelajaran mesin, dengan catatan bahwa pemahaman yang baik tentang kelemahan ini perlu diperhatikan dalam penggunaan dan interpretasi hasilnya.

11.5 Penggunaan Algoritma JST

Algoritma jaringan syaraf tiruan (JST) dapat digunakan dalam berbagai situasi dan masalah. Berikut adalah beberapa situasi di mana penggunaan JST mungkin tepat: (Ahamed and Akthar, 2016)

1. **Pengenalan Pola:** JST sering digunakan dalam pengenalan pola untuk mengenali dan mengklasifikasikan objek atau data berdasarkan pola-pola yang terdapat dalam input. Contoh penggunaan JST dalam pengenalan pola meliputi pengenalan wajah, pengenalan tulisan tangan, atau pengenalan suara.
2. **Prediksi dan Ramalan:** JST dapat digunakan untuk memprediksi atau meramalkan suatu kejadian atau perilaku berdasarkan data historis. Contoh penggunaan JST dalam prediksi meliputi prediksi harga saham, prediksi cuaca, atau prediksi permintaan produk.
3. **Pengenalan dan Pemrosesan Bahasa:** JST digunakan dalam aplikasi pemrosesan bahasa alami (*natural language processing*) untuk tugas seperti pemodelan bahasa, analisis sentimen, penerjemahan mesin, atau pengenalan entitas dalam teks.
4. **Pengenalan Suara:** JST digunakan dalam sistem pengenalan suara untuk mengenali kata-kata, perintah suara, atau untuk aplikasi pengenalan pembicara (*speaker identification*).
5. **Pengenalan Gambar:** JST digunakan dalam pengenalan gambar untuk mendeteksi objek, mengklasifikasikan

gambar, atau melaksanakan tugas-tugas seperti segmentasi gambar atau rekonstruksi gambar.

6. Kontrol dan Optimisasi: JST digunakan dalam sistem kontrol dan optimisasi untuk mengendalikan proses-proses kompleks dan membuat keputusan berdasarkan input sensorik. Contoh penggunaan JST dalam kontrol dan optimisasi meliputi kendali robot, kendali lalu lintas, atau optimisasi jaringan listrik.
7. Analisis Data dan Prediksi Pola: JST digunakan dalam analisis data untuk menemukan pola-pola tersembunyi dalam data yang kompleks dan membuat prediksi berdasarkan pola-pola tersebut. Contoh penggunaan JST dalam analisis data meliputi analisis pemrosesan sinyal, deteksi anomali, atau segmentasi data.
8. Rekomendasi dan Personalisasi: JST digunakan dalam sistem rekomendasi untuk memberikan rekomendasi yang disesuaikan dengan preferensi dan perilaku pengguna. Contoh penggunaan JST dalam rekomendasi meliputi sistem rekomendasi film, musik, atau produk e-commerce.
9. Pengenalan Tulisan Tangan: JST digunakan dalam pengenalan tulisan tangan untuk mengubah tulisan tangan menjadi teks yang dapat dibaca komputer, seperti dalam sistem pengenalan karakter optik (*Optical Character Recognition/OCR*).
10. Analisis Data Medis: JST dapat digunakan dalam analisis data medis untuk mendeteksi penyakit, mendiagnosis kondisi medis, atau memprediksi risiko penyakit berdasarkan data medis seperti gambar medis, catatan medis, atau data biometrik.
11. Pengenalan Penyimpangan: JST dapat digunakan untuk mendeteksi penyimpangan atau anomali dalam data, baik dalam konteks keamanan jaringan untuk mendeteksi

- serangan cyber, maupun dalam bidang manufaktur untuk mendeteksi cacat dalam produk.
12. Kendali Otomatis: JST dapat digunakan dalam sistem kendali otomatis untuk mengambil keputusan berdasarkan input sensorik dan mengontrol proses secara otomatis. Contohnya termasuk kendali otomatis dalam industri, kendaraan otonom, atau robotika.
 13. Pemodelan Fisika: JST dapat digunakan dalam pemodelan fisika untuk memprediksi perilaku sistem fisik yang kompleks atau melakukan simulasi berdasarkan prinsip-prinsip fisika. Contoh penggunaan JST dalam pemodelan fisika meliputi prediksi cuaca, simulasi dinamika fluida, atau prediksi pergerakan benda di alam.
 14. Pengenalan Ciri: JST dapat digunakan dalam pengenalan ciri atau fitur dalam data untuk tujuan pengklasifikasian atau pengelompokan. Contoh penggunaan JST dalam pengenalan ciri meliputi identifikasi sidik jari, pengenalan tulisan tangan, atau deteksi emosi dari ekspresi wajah.
 15. Prosesor Bicara: JST dapat digunakan dalam sistem prosesor bicara untuk mengubah suara menjadi teks atau menerjemahkan bahasa yang diucapkan menjadi bahasa lain.
 16. Pengenalan Gerakan: JST dapat digunakan dalam pengenalan gerakan untuk mengenali pola gerakan manusia, seperti dalam aplikasi pengenalan gerakan tangan untuk interaksi dengan perangkat.
 17. Prediksi Kegagalan Sistem: JST dapat digunakan untuk memprediksi kegagalan atau masalah pada sistem teknis atau mesin berdasarkan data pemantauan dan sensorik, sehingga memungkinkan perawatan preventif atau perbaikan yang tepat waktu.
 18. Pengenalan Pola Geospasial: JST dapat digunakan dalam pengenalan pola geospasial untuk analisis citra satelit,

pemetaan, penginderaan jauh, atau pengenalan pola dalam data spasial.

19. Perdagangan Saham dan Keuangan: JST dapat digunakan dalam analisis keuangan dan prediksi pergerakan pasar saham untuk mengidentifikasi tren dan membuat keputusan perdagangan yang lebih baik.
20. Pemodelan Biologi dan Kesehatan: JST dapat digunakan dalam pemodelan biologi, seperti prediksi struktur protein, simulasi dinamika molekuler, atau analisis data genomik untuk memahami penyakit dan mencari solusi pengobatan.

Penting untuk mempertimbangkan kompleksitas masalah yang dihadapi, ketersediaan data yang cukup, serta kemampuan komputasi yang tersedia sebelum memutuskan penggunaan JST dalam suatu aplikasi tertentu. Selain itu, faktor seperti interpretabilitas hasil, kecepatan komputasi, dan sumber daya yang tersedia juga harus dipertimbangkan.

Dalam beberapa kasus, penggunaan JST mungkin tidak selalu menjadi pilihan terbaik. Beberapa kekurangan dari algoritma JST yang perlu dipertimbangkan adalah:

1. Ketergantungan pada Data yang Tersedia: JST membutuhkan jumlah data yang cukup besar dan representatif untuk melatih model dengan baik. Jika data yang tersedia terbatas atau tidak mewakili variasi yang cukup, JST dapat menghasilkan model yang kurang akurat.
2. Kecepatan Pelatihan yang Lambat: JST dapat membutuhkan waktu yang lama untuk melatih model, terutama jika datasetnya besar atau JST memiliki kompleksitas yang tinggi. Proses iteratif propagasi maju dan propagasi balik untuk memperbarui bobot dan bias dalam JST dapat memakan waktu.
3. Overfitting: JST rentan terhadap overfitting, yaitu ketika model terlalu rumit dan mampu "menghafal" data

- pelatihan, tetapi tidak mampu melakukan generalisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Hal ini dapat mengurangi kemampuan JST untuk memberikan prediksi yang akurat pada data baru.
4. Kompleksitas Model: JST dengan jumlah lapisan dan neuron yang besar dapat menjadi sangat kompleks, sulit untuk dipahami, dan membutuhkan pengaturan parameter yang hati-hati. Memilih arsitektur JST yang tepat dan mengoptimalkan hyperparameter dapat menjadi tugas yang rumit.
 5. Penyesuaian Hyperparameter: JST memiliki beberapa hyperparameter yang perlu ditentukan oleh pengguna, seperti learning rate, jumlah lapisan, jumlah neuron, dan fungsi aktivasi. Memilih nilai yang tepat untuk hyperparameter ini dapat mempengaruhi kinerja dan kestabilan JST.
 6. Kesulitan dalam Interpretasi: Hasil dari JST sering kali sulit untuk diinterpretasikan. JST bekerja dengan cara yang tidak sepenuhnya dapat dipahami oleh manusia, sehingga sulit untuk menjelaskan mengapa suatu keputusan atau prediksi dibuat oleh model.
 7. Kebutuhan Komputasi yang Tinggi: JST dengan arsitektur yang kompleks dan dataset yang besar dapat memerlukan daya komputasi yang signifikan. Pelatihan dan evaluasi model JST dapat membutuhkan waktu yang lama dan sumber daya komputasi yang cukup, terutama jika menggunakan GPU untuk mempercepat perhitungan.
 8. Sensitivitas terhadap Data yang Tidak Relevan atau Noise: JST dapat menjadi sensitif terhadap data yang tidak relevan atau adanya noise dalam dataset. Jika dataset mengandung informasi yang tidak penting atau gangguan, hal ini dapat mempengaruhi performa JST dan mengurangi akurasi prediksinya.

9. Keterbatasan pada Data Langka atau Tidak Seimbang: JST dapat menghadapi kesulitan dalam mengatasi dataset yang langka atau tidak seimbang. Jika terdapat kelas atau label yang sangat sedikit dalam dataset, JST mungkin tidak dapat melatih model yang memadai untuk mengenali kelas tersebut dengan baik.
10. Tidak Menjamin Solusi Optimal: Meskipun JST dapat memberikan hasil yang baik dalam banyak kasus, mereka tidak menjamin solusi optimal untuk setiap masalah. Hasil yang dihasilkan oleh JST dapat dipengaruhi oleh inisialisasi bobot yang acak, arsitektur model yang dipilih, dan metode pelatihan yang digunakan.
11. Ketergantungan pada Ketersediaan Data Masa Depan: JST mempelajari pola-pola dari data pelatihan yang tersedia pada saat pelatihan. Namun, jika pola-pola tersebut tidak mencerminkan dengan akurat kondisi masa depan atau data yang belum pernah dilihat sebelumnya, performa JST dapat menurun. Perlu diperhatikan bahwa JST tidak memiliki kemampuan untuk memprediksi atau mengenali situasi yang belum pernah dihadapi sebelumnya.
12. Keterbatasan dalam Menangani Data Kontinu: JST lebih umum digunakan untuk data yang memiliki struktur dan representasi diskrit. Jika data yang akan diproses bersifat kontinu, seperti data sensorik yang berkelanjutan, JST dapat menghadapi kesulitan dalam memodelkan dan menginterpretasikan pola yang kompleks dalam data tersebut.
13. Kesulitan dalam Pemilihan Arsitektur yang Optimal: Memilih arsitektur JST yang optimal untuk masalah yang dihadapi bisa menjadi tantangan tersendiri. Terdapat berbagai jenis arsitektur JST, seperti Jaringan Saraf Multilayer (MLP), Jaringan Saraf Konvolusi (CNN), atau Jaringan Saraf Rekurensi (RNN), masing-masing dengan

- keunggulan dan kelemahan mereka sendiri. Menentukan arsitektur yang tepat serta konfigurasi parameter yang optimal dapat memerlukan pengetahuan dan eksperimen yang mendalam.
14. Kerentanan terhadap Data Outlier: JST cenderung sensitif terhadap data outlier, yaitu data yang berbeda secara signifikan dari pola umum dalam dataset. Jika ada data outlier dalam dataset, hal ini dapat mempengaruhi proses pelatihan dan menghasilkan model yang tidak akurat atau tidak stabil.
 15. Interpretabilitas yang Terbatas: Meskipun JST dapat memberikan prediksi yang akurat, interpretabilitas model sering kali menjadi tantangan. Model JST cenderung dianggap sebagai "kotak hitam" karena kompleksitasnya, sehingga sulit untuk memahami alasan di balik keputusan yang dibuat oleh model. Hal ini dapat menjadi kendala dalam situasi di mana interpretasi dan penjelasan yang jelas diperlukan, seperti dalam pengambilan keputusan yang kritis atau dalam bidang yang diatur oleh hukum atau etika.
 16. Ketergantungan pada Data Label yang Benar: JST membutuhkan dataset yang dilabeli dengan benar untuk melatih model dengan baik. Proses pemberian label yang akurat dan konsisten pada data dapat memakan waktu dan sumber daya manusia yang signifikan. Selain itu, jika dataset memiliki kesalahan labeling atau ketidakseimbangan label, dapat mempengaruhi performa dan generalisasi JST.
 17. Dibutuhkan Pengalaman dan Pengetahuan Khusus: Implementasi dan penggunaan JST memerlukan pemahaman yang mendalam tentang prinsip-prinsip dan konsep dasar dalam jaringan saraf. Selain itu, diperlukan pengetahuan tentang pemilihan arsitektur yang tepat,

tuning parameter, dan teknik pengoptimalan. Memiliki pengalaman dan pemahaman yang memadai dalam bidang ini akan membantu dalam memaksimalkan potensi JST dan mengatasi kendala yang mungkin muncul.

Pemahaman yang mendalam tentang kelebihan dan kekurangan algoritma Jaringan Saraf Tiruan memungkinkan programmer untuk memilih dengan bijak dan mempersiapkan langkah-langkah yang diperlukan dalam mengimplementasikan dan menggunakan JST untuk memecahkan masalah yang relevan. Meskipun JST memiliki beberapa kekurangan, mereka tetap menjadi algoritma yang kuat dan efektif dalam banyak aplikasi. Kelebihan dan kekurangan JST perlu dipertimbangkan dengan cermat sesuai dengan kebutuhan dan karakteristik masalah yang dihadapi. Selain itu, penggunaan teknik lain seperti pemrosesan data yang tepat, pemilihan fitur yang baik, dan penggunaan ensemble model juga dapat membantu mengatasi beberapa kekurangan tersebut dan meningkatkan performa JST.

11.6 Contoh Kasus Penggunaan Algoritma JST

Algoritma Jaringan Saraf Tiruan (JST) dapat digunakan dalam berbagai kasus untuk memecahkan masalah yang kompleks. Berikut ini adalah beberapa contoh kasus yang dapat diselesaikan menggunakan JST: (Jain, Mao and Mohiuddin, 1996)

1. **Klasifikasi Gambar:** JST dapat digunakan untuk mengklasifikasikan gambar berdasarkan kategori tertentu. Contohnya, mengenali jenis objek dalam gambar seperti mobil, pesawat, atau hewan.
2. **Deteksi Objek:** JST dapat digunakan untuk mendeteksi objek dalam gambar atau video. Contohnya, mendeteksi

- wajah manusia dalam gambar atau mendeteksi mobil dalam rekaman video lalu lintas.
3. Pengenalan Suara: JST dapat digunakan untuk pengenalan suara, seperti mengubah suara menjadi teks atau menerjemahkan bahasa yang diucapkan menjadi bahasa tertentu.
 4. Penerjemahan Bahasa: JST dapat digunakan untuk menerjemahkan teks dari satu bahasa ke bahasa lain. Contohnya, menerjemahkan teks dari bahasa Inggris ke bahasa Spanyol.
 5. Sistem Rekomendasi: JST dapat digunakan untuk membangun sistem rekomendasi yang dapat merekomendasikan produk, musik, film, atau konten lainnya berdasarkan preferensi pengguna.
 6. Prediksi Pasar Saham: JST dapat digunakan untuk menganalisis data pasar saham dan memprediksi pergerakan harga saham di masa depan.
 7. Deteksi Penipuan: JST dapat digunakan untuk mendeteksi aktivitas penipuan dalam transaksi keuangan, seperti kartu kredit atau asuransi.
 8. Analisis Sentimen: JST dapat digunakan untuk menganalisis sentimen dalam teks, seperti menentukan apakah suatu ulasan produk bersifat positif atau negatif.
 9. Pengenalan Tulisan Tangan: JST dapat digunakan untuk mengenali tulisan tangan dan mengonversinya menjadi teks yang dapat dibaca.
 10. Diagnosis Medis: JST dapat digunakan dalam analisis data medis untuk membantu mendiagnosis penyakit, mengenali pola dalam data medis, atau memprediksi risiko penyakit.
 11. Pemrosesan Bahasa Alami: JST dapat digunakan untuk memahami dan memproses bahasa manusia secara alami, seperti menjawab pertanyaan berdasarkan konteks

tertentu atau melakukan penerjemahan bahasa alami ke perintah yang dapat dipahami oleh mesin.

12. Pengenalan Tulisan Tangan: JST dapat digunakan untuk mengenali tulisan tangan dan mengonversinya menjadi teks yang dapat dibaca.
13. Otomasi Industri: JST dapat digunakan dalam sistem otomasi industri untuk mengendalikan dan mengoptimalkan proses produksi dan operasional.
14. Pemetaan dan Penginderaan Jauh: JST dapat digunakan dalam analisis data pemetaan dan penginderaan jauh untuk mengklasifikasikan dan menganalisis citra satelit atau data geospasial lainnya.
15. Pengenalan Pola dan Pengelompokan: JST dapat digunakan untuk pengenalan pola dalam data, seperti mengelompokkan data berdasarkan kesamaan.
16. Pengenalan Tulisan Tangan: JST dapat digunakan untuk mengenali tulisan tangan dan mengonversinya menjadi teks yang dapat dibaca.
17. Prediksi Cuaca: JST dapat digunakan untuk menganalisis data cuaca historis dan saat ini untuk memprediksi kondisi cuaca di masa depan, termasuk suhu, kelembaban, dan kemungkinan hujan.
18. Kendali Proses Industri: JST dapat digunakan dalam sistem kendali proses industri untuk mengoptimalkan operasi dan memprediksi kegagalan peralatan dengan memantau data sensorik dan mengambil tindakan yang sesuai.
19. Analisis Data Geospasial: JST dapat digunakan dalam analisis data geospasial untuk memahami pola dan tren dalam data seperti pemodelan permukaan, perubahan penggunaan lahan, atau prediksi pergerakan populasi.
20. Pengenalan Emosi: JST dapat digunakan untuk mengenali ekspresi wajah atau sentimen dalam teks untuk mengidentifikasi emosi manusia, yang dapat digunakan

dalam aplikasi seperti analisis feedback pelanggan atau penelitian psikologi.

21. Pengendalian Robot: JST dapat digunakan dalam pengendalian robot untuk membantu robot mempelajari dan menyesuaikan perilakunya berdasarkan interaksi dengan lingkungannya.
22. Pengenalan Pakaian dan Mode: JST dapat digunakan untuk mengenali jenis pakaian, gaya, dan tren dalam industri mode, yang dapat digunakan untuk sistem rekomendasi fashion atau analisis tren pasar.
23. Deteksi Penyakit pada Citra Medis: JST dapat digunakan dalam analisis citra medis, seperti pemindaian MRI atau CT scan, untuk mendeteksi dan mendiagnosis penyakit seperti kanker atau penyakit jantung.
24. Pengenalan Teks pada Gambar: JST dapat digunakan untuk mengenali dan mengekstrak teks dari gambar, yang dapat digunakan dalam aplikasi seperti pengenalan plat nomor kendaraan atau pembacaan barcode.
25. Peramalan Permintaan: JST dapat digunakan untuk memprediksi permintaan produk atau layanan di masa depan berdasarkan data historis, membantu perusahaan dalam perencanaan persediaan dan manajemen rantai pasok.

Penting untuk dicatat bahwa daftar ini hanya memberikan beberapa contoh dari berbagai kasus yang dapat diselesaikan menggunakan JST. Dalam praktiknya, JST dapat diterapkan dalam banyak bidang lainnya sesuai dengan kebutuhan dan ketersediaan data yang relevan.

11.7 Cara Kerja Algoritma JST

Algoritma Jaringan Saraf Tiruan (JST) bekerja dengan mensimulasikan cara kerja neuron dalam otak manusia. JST terdiri dari jaringan neuron buatan yang terhubung satu sama lain dan mampu melakukan pemrosesan informasi dan pembelajaran berdasarkan data yang diberikan. Berikut adalah langkah-langkah umum dalam cara kerja algoritma JST:

1. **Inisialisasi Bobot dan Bias:** Pada awalnya, bobot dan bias jaringan saraf diinisialisasi secara acak dengan nilai yang kecil. Bobot menggambarkan kekuatan hubungan antara neuron dalam jaringan, sedangkan bias adalah nilai tambahan yang ditambahkan ke keluaran neuron.
2. **Propagasi Maju (*Forward Propagation*):** Proses propagasi maju adalah langkah di mana data masukan diteruskan melalui jaringan dari lapisan input ke lapisan output. Pada setiap neuron, data masukan dikalikan dengan bobot yang sesuai dan dijumlahkan dengan bias. Kemudian, fungsi aktivasi diterapkan pada hasil penjumlahan untuk menghasilkan keluaran neuron. Proses ini berlanjut melalui setiap lapisan dalam jaringan hingga mencapai lapisan output.
3. **Perhitungan Error:** Setelah propagasi maju, perhitungan error dilakukan untuk mengevaluasi sejauh mana output jaringan sesuai dengan output yang diharapkan. Error diperoleh dengan membandingkan keluaran aktual jaringan dengan target yang diinginkan.
4. **Propagasi Balik (*Backpropagation*):** Setelah perhitungan error, langkah berikutnya adalah melakukan propagasi balik. Propagasi balik melibatkan penyesuaian bobot dan bias di setiap neuron berdasarkan error yang ditemukan. Algoritma backpropagation memperhitungkan kontribusi setiap bobot terhadap error dan menggunakan metode

- gradien turunan untuk memperbarui bobot dan bias agar jaringan menghasilkan output yang lebih akurat.
5. Iterasi Pelatihan: Proses propagasi maju, perhitungan error, dan propagasi balik diulang dalam beberapa iterasi atau epoch untuk melatih jaringan. Pada setiap iterasi, bobot dan bias diperbarui untuk meminimalkan error dan meningkatkan kinerja jaringan.
 6. Uji dan Evaluasi: Setelah melatih jaringan, langkah terakhir adalah menguji jaringan menggunakan data uji yang belum pernah dilihat sebelumnya. Jaringan memberikan keluaran berdasarkan data uji, dan performanya dievaluasi dengan membandingkan hasil prediksi dengan nilai target yang diketahui. Hasil evaluasi ini digunakan untuk mengukur kinerja jaringan dan menentukan apakah jaringan telah belajar dengan baik atau perlu disesuaikan lebih lanjut.

Proses ini terus diulang hingga jaringan mencapai tingkat akurasi yang memadai dalam tugas yang ditugaskan, seperti klasifikasi, regresi, atau pengenalan pola. Algoritma JST memiliki fleksibilitas yang besar dan dapat diadaptasi untuk berbagai jenis masalah dan aplikasi.

DAFTAR PUSTAKA

- Ahamed, K.I. and Akthar, S. 2016. 'A Study on Neural Network Architectures', *Computer Engineering and Intelligent Systems*, 7(9), pp. 1–7. Available at: <https://iiste.org/Journals/index.php/CEIS/article/view/32857>.
- Jain, A.K., Mao, J. and Mohiuddin, K.M. 1996. 'Artificial neural networks: A tutorial', *Computer*, 29(3), pp. 31–44. Available at: <https://doi.org/10.1109/2.485891>.

BAB 12

PLANNING

Oleh Ratna Dewi

12.1 Pengertian Planning

Perencanaan berbeda dari pemecahan masalah berbasis pencarian dalam representasi tujuan, ruang dan aktivitas, serta representasi dan struktur urutan tindakan. Desain ini bertujuan untuk mengatasi kesulitan yang terkait dengan pemecahan masalah berbasis pencarian. Desain adalah metode pemecahan masalah dimana suatu masalah dipecah menjadi submasalah yang lebih kecil, submasalah diselesaikan satu persatu, kemudian solusi dari submasalah tersebut digabungkan menjadi solusi lengkap, mengenali dan mengatasi interaksi yang ada pada masalah.

Pengujian fungsional dari metode desain dapat dilakukan dalam domain yang disebut Blocks-World. dinilai cukup bermanfaat sebagai ajang uji coba karena kompleksitas permasalahan yang bisa muncul di sana. Dalam dunia blok dikenal istilah start state dan gate state yang masing-masing diwakili oleh komposisi beberapa blok. Lalu ada sekumpulan operator yang dapat digunakan dengan robot untuk memanipulasi. Masalah di dunia blok adalah Urutan operator apa yang dapat mengubah keadaan awal menjadi keadaan target, Serangkaian aktor biasanya disebut rencana akhir Dua metode perencanaan yang cukup populer dan teruji di Block World adalah *Goal-Stack-Planning* (GSP) dan *Constraint-Posting* (CP). GSP dan CP memiliki kelebihan dan kekurangannya masing-masing. Dalam hal kemudahan implementasi dan biaya komputasi, GSP lebih baik daripada CP.

Sedangkan CP umumnya lebih baik dari GSP dalam hal efisiensi rencana solusi yang dihasilkan. Namun, untuk beberapa masalah potensial yang mungkin muncul di World of Blocks, meskipun GSP dan CP mampu membuat rencana solusi, rencana solusi yang dihasilkan biasanya tidak efektif [RIC91]. Namun, rencana resolusi yang efektif adalah salah satu hal terpenting, terutama untuk sistem raksasa. Dalam bidang-bidang tertentu, masalah harus diselesaikan dengan menggunakan komputer dengan cara memperlakukan bagian-bagian kecil dari masalah tersebut secara terpisah dan menggabungkan solusi dari bagian kecil tersebut untuk membentuk solusi yang lengkap dari masalah tersebut. Karena jika hal-hal tersebut tidak dilakukan, maka jumlah kombinasi state dari komponen yang bermasalah menjadi terlalu besar untuk waktu yang tersedia. Ada dua kondisi untuk melakukan dekomposisi di atas, Hindari menghitung ulang semua status masalah ketika ada perubahan dari satu status ke status lainnya. Membagi masalah menjadi beberapa submasalah yang relatif lebih mudah dipecahkan. Desain adalah penggunaan metode yang berfokus pada memecah masalah menjadi submasalah yang sesuai dan cara untuk mengingat dan menangani interaksi antara submasalah dalam proses pemecahan masalah. Memecahkan masalah dengan desain umum. *Goal-Directed state*, dimana solusi dicari dari state tujuan ke state awal yang dapat dicapai *ependency-Directed-Backtracking*.

Kecerdasan buatan atau *artificial intelligence* (AI) adalah teknologi di bidang ilmu komputer yang mensimulasikan kecerdasan manusia di dalam mesin (komputer) untuk menyelesaikan berbagai masalah bekerja layaknya seperti manusia, bahkan lebih baik dari manusia. Kecerdasan buatan adalah kecerdasan yang ditambahkan, diatur, dan dikembangkan orang ke dalam sistem teknologi dalam konteks

ilmiah, dan yang dibentuk dari kecerdasan unit-unit ilmiah yang ada. Pada dasarnya kecerdasan buatan atau *artificial intelligence* (AI) adalah pengetahuan yang memungkinkan komputer meniru kecerdasan manusia sehingga komputer dapat melakukan hal-hal yang dilakukan manusia yang membutuhkan kecerdasan, seperti analisis inferensial atau penerjemahan AI untuk membuat inferensi atau keputusan. dimulai pada musim panas 1956. Saat itu, sekelompok pakar dan peneliti komputer dari berbagai disiplin ilmu dan akademisi, industri dan berbagai kalangan berkumpul di Dartmouth College. membahas kemampuan komputer dalam konteks meniru atau mensimulasikan kecerdasan manusia (Rakhmad dan Sumarno, 2022).

Adanya kecerdasan buatan atau lebih dikenal *artificial intelligence* (AI) di era globalisasi ini memungkinkan komputer, robot, dan perangkat lunak dapat berpikir seperti manusia. Konsep kecerdasan buatan didasarkan pada penelitian tentang cara kerja otak manusia, seperti belajar, memecahkan masalah, dan membuat keputusan. Kecerdasan buatan sekarang dapat diterapkan dengan berbagai cara. Salah satunya dalam pengolahan data yang dapat membantu pemasaran produk di media sosial. Dalam hal ini, AI berkembang cukup pesat di dunia digital marketing karena kompleksitasnya. Media sosial bertenaga AI untuk pemasaran digital sangat membantu penjual produk. Pemasar dapat mengelompokkan pasar sasaran, memprediksi tren yang muncul, memahami reaksi pembeli terhadap suatu produk, dan membuat keputusan pengguna. Ini dapat memastikan bahwa kombinasi kecerdasan buatan dan media sosial dapat memberikan berbagai peluang kepada penjual produk, salah satunya adalah data yang mengangkat strategi pemasaran ke tingkat yang lebih tinggi. Namun, lebih baik tidak menyerah dan menyerahkan semuanya pada produk kecerdasan buatan. Karena marketing

terbaik adalah marketing dalam komunikasi dengan orang lain yang bisa memberikan sentuhan emosional. (Untung Rahardja,2022).



Gambar 12.1. Kecerdasan Buatan digital planning
Sumber:<https://jurnalpost.com/belajar-mengenai-perkembangan-artificial-intelligence-melalui-program-mbkm/35890/>

Kecerdasan buatan adalah salah satu teknologi yang banyak digunakan oleh semua jenis orang Perusahaan kecil dan besar termasuk Amazon, Netflix, Flipkart, dan YouTube sistem dan proses, menghasilkan lebih banyak pendapatan dan membuat keputusan yang lebih baik sesuai Sistem otomatis dan mobil self-driving adalah contoh kecerdasan yang bagus palsu Namun, selama proses pengembangan, beberapa pertanyaan muncul yang diperlukan berhasil mencapai keseimbangan yang baik antara kecerdasan buatan, etika bisnis, dan manusia. Bahwa tujuan utama pengembangan kecerdasan buatan adalah kemajuan masyarakat dan tidak

menentanginya. Studi ini membahas beberapa masalah kritis yang muncul sebagai hasil implementasi kecerdasan buatan dan mengusulkan cara untuk mengatasi masalah tersebut. Kecerdasan buatan berkembang sangat cepat dan sebagian besar perusahaan mencoba untuk mengintegrasikannya ke dalam solusi dari masalah mereka. ini karena arsitektur plugin yang sederhana. ada berbagai aplikasi Kecerdasan buatan memengaruhi semua sektor utama. kehadiran berbasis wajah, chatbots, perawatan kesehatan berbasis AI, menjadi bagian itu tidak dapat dipisahkan dari kehidupan sehari-hari. Kecerdasan buatan membutuhkan data membuat keputusan yang baik atau bijak. Studi ini merangkum Topik terkait kecerdasan buatan dan etika. Perusahaan besar seperti Alphabet, Google, Facebook, IBM, Amazon, dan Microsoft dengan pahlawan visioner seperti Stephen Hawking, peneliti kecerdasan buatan yang terkenal dan Elon Musk berpikir sekaranglah waktunya untuk angkat bicara cakupan AI yang bisa disempurnakan hampir tidak ada habisnya. dalam banyak hal, itu mewakili kemajuan baru dalam etika dan penilaian risiko dan teknologi yang tumbuh (Soni Agung Wahyudiyanta,2023).

12.2 Kecerdasari buatan dapat dilihat dari sudut pandang, adalah sebagai berikut:

1. Kecerdasan buatan adalah bagaimana membuat mesin yang cerdas dan dapat melakukan hal-hal yang dapat dilakukan manusia sebelumnya.
2. Perspektif penelitian. Studi tentang bagaimana membuat komputer melakukan hal-hal seperti manusia.
3. Perspektif bisnis. Seperangkat alat yang sangat canggih untuk memecahkan masalah bisnis.

4. Perspektif pemrograman Pemrograman . Kecerdasan buatan mencakup studi tentang pemrograman simbolik, pemecahan masalah, dan proses pencarian.

Fungsi dan Tujuan Kecerdasan Buatan AI memiliki banyak tugas dan berbagai hal diharapkan dapat memfasilitasi pekerjaan seseorang, seperti pemrosesan bahasa alami, penalaran, persepsi, memindahkan dan memanipulasi objek, pengetahuan atau juga melakukan pelajaran. Kesimpulannya adalah kecerdasan buatan atau kecerdasan buatan (AI mencoba membuat robot dengan kecerdasan yang mirip atau bahkan lebih cerdas dari kecerdasan manusia. Tujuan menciptakan kecerdasan buatan adalah: (Hanson,2023).

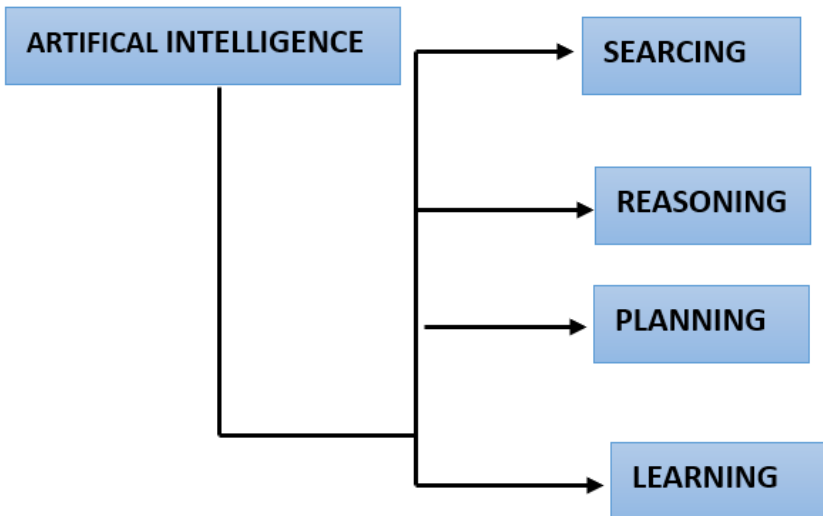
1. Membuat mesin lebih pintar.
2. Memahami akal.
3. Jadikan mesin lebih bermanfaat.
4. Mempersingkat jam kerja untuk membuat operasi lebih efisien
5. Kecerdasan buatan memudahkan orang untuk menyelesaikan tugas dan membuat keputusan.
6. Pekerjaan waktu pemrosesan lebih cepat.

12.3 Faktor-faktor yang mendorong pengembangan kecerdasan

Perkembangan pesat teknologi perangkat keras. Hampir semua aplikasi membutuhkan perangkat keras, memiliki kekuatan dan kecepatan lebih, meskipun hanya menggunakan perangkat lunak AI yang paling sederhana. Pengembangan perangkat lunak kecerdasan buatan. Bahasa dan platform pemrograman tingkat tinggi baru telah dikembangkan dan dikomersialkan secara luas, termasuk bahasa cerdas banyak orang yang telah menggunakan komputer yang telah

menyatakan keinginannya untuk menyediakan perangkat lunak canggih untuk pekerjaan mereka.⁴ Investor cukup terlibat dalam penelitian dan pengembangan teknologi kecerdasan buatan, sehingga menimbulkan semacam tekanan untuk tampil di depan umum.

12.4 Teknik Dasar dalam *Artificial Intelligence*



Gambar 12.2. *Teknik Dasar dalam Artificial Intelligence*

a. *Searching*

Teknik pencarian, yaitu teknik pemecahan masalah yang merepresentasikan masalah dalam ruang keadaan dan secara sistematis membangkitkan dan menguji keadaan dari keadaan awal sampai keadaan tujuan ditemukan. Digunakan untuk memandu seseorang dalam perjalanannya untuk menemukan rute yang optimal, misalnya di Swedia setiap taksi dilengkapi dengan GPS (*Global Positioning System*)

b. Reasoning

Teknik penalaran, yaitu teknik pemecahan masalah yang merepresentasikan masalah seperti logika (alat matematika yang digunakan untuk merepresentasikan dan memanipulasi fakta dan aturan). Perangkat lunak catur HITECH adalah sistem kecerdasan buatan pertama yang mengalahkan grandmaster dunia Arnold Danker

c. Planning

Suatu metode penyelesaian masalah dengan cara memecah masalah dalam sub-sub masalah yang lebih kecil, menyelesaikan sub-sub masalah satu demi satu, kemudian menggabungkan solusi-solusi dari sub-sub masalah tersebut menjadi sebuah solusi lengkap dengan tetap mengingat dan menangani interaksi yang terdapat pada sub-sub masalah tersebut. Dalam dunia manufaktur dan robotik. Software Optimum – AIV adalah suatu planner yang digunakan oleh *European Space Agency* untuk perakitan pesawat terbang.

d. Learning

Metode pemecahan masalah, dimana masalah dibagi menjadi submasalah yang lebih kecil, submasalah diselesaikan satu per satu, kemudian solusi dari submasalah tersebut digabungkan menjadi solusi lengkap, dengan mempertimbangkan dan menangani interaksi antar submasalah. - masalah. Di dunia manufaktur dan robotika. Perangkat Lunak Optimal - AIV adalah perancang yang digunakan oleh Badan Antariksa Eropa untuk merakit pesawat. secara otomatis menemukan aturan yang biasanya berlaku untuk data yang tidak kami ketahui keberadaannya. digunakan dalam bidang transportasi. Perangkat lunak ALVINN digunakan di dalam mobil tanpa pengemudi manusia menggunakan JST yang dilatih dengan gambar

kondisi jalan dari kamera mobil yang berbeda. (Kevin *et al*, 2022).

12.5 Konsep Dasar Kecerdasan Buatan (*Artificial Intelligence*)

Konsep Dasar Kecerdasan Buatan Kemajuan yang dibuat oleh kecerdasan buatan, atau lebih dikenal dengan kecerdasan buatan, sangat mengejutkan. Untuk memajukan konsep kecerdasan buatan telah menghasilkan beberapa inovasi yang benar-benar luar biasa. Hasil inovasi yang mungkin sudah banyak diketahui orang, seperti big data, penelitian medis, Teknologi ini hanyalah beberapa dari sekian banyak aplikasi luarbiasa yang muncul dari pengembangan kecerdasan buatan. Untuk memahami beberapa konsep yang lebih dalam seperti manajemen data, pemrosesan bahasa alami (NLP) dan manajemen perangkat lunak, Anda perlu mengetahui tiga konsep dasar kecerdasan buatan yaitu pembelajaran mesin, pembelajaran mendalam. dan jaringan syaraf tiruan (Neural network). Kecerdasan buatan umumnya dianggap sebagai istilah yang lebih luas, sementara pembelajaran mesin dan dua kecerdasan buatan lainnya adalah bagian darinya. (Novi *et al*, 2022).

Pembelajaran mesin Saat ini, kita secara tidak sadar dapat berinteraksi dengan beberapa bentuk kecerdasan buatan dalam aktivitas kita sehari-hari. Misalnya, jika Anda menggunakan Gmail, Anda dapat menikmati pemfilteran email otomatis, atau jika Anda memiliki ponsel cerdas, Meskipun perangkat lunak ini sangat berguna, ia tidak dapat belajar dengan sendirinya. Ini karena perangkat ini tidak dapat berpikir di luar kode yang diberikan kepada mereka. Pembelajaran mesin adalah cabang kecerdasan buatan yang bertujuan untuk memungkinkan mesin mempelajari tugas tanpa kode yang ada. Sederhananya, mesin menerima sampel

uji dalam jumlah besar untuk tugas tertentu. Saat mesin diuji, ia mempelajari dan menyesuaikan strategi yang tepat untuk mencapai tujuan yang diinginkan. Misalnya, mesin pengenalan gambar dapat menerima jutaan gambar untuk dianalisis. Setelah mutasi yang lama, mesin dapat mengenali pola, bentuk, wajah, dan lainnya. Pembelajaran mendalam Pembelajaran mendalam adalah teknik pembelajaran mesin yang mengajarkan komputer untuk melakukan apa yang alami bagi manusia. belajar dengan contoh. Pembelajaran mendalam adalah teknologi inti untuk mobil tanpa pengemudi. Hal ini memungkinkan mereka mengenali tanda berhenti atau membedakan pejalan kaki dari tiang lampu. Ini adalah kunci kontrol suara di perangkat konsumen seperti ponsel, tablet, TV, dan speaker bebas genggam.

Pembelajaran mendalam mendapatkan banyak perhatian akhir-akhir ini karena dapat mencapai hasil yang sebelumnya tidak mungkin dilakukan. Dalam pembelajaran mendalam, model komputer belajar melakukan tugas klasifikasi langsung dari gambar, teks, atau audio. Model pembelajaran mendalam dapat mencapai akurasi canggih yang terkadang melebihi kinerja manusia. Model ini dilatih menggunakan sejumlah besar data berlabel dan arsitektur jaringan saraf multilayer. Jaringan saraf tiruan (jaringan saraf) Neural network, atau juga jaringan saraf tiruan, adalah berbagai teknik pembelajaran mendalam yang juga dikategorikan dalam kecerdasan buatan atau kecerdasan buatan. Jaringan saraf adalah paradigma pemrosesan informasi yang terinspirasi oleh cara sistem saraf biologis, seperti otak, memproses informasi. Elemen kunci dari paradigma ini adalah struktur baru dari sistem pemrosesan informasi. Jaringan saraf dikonfigurasi untuk aplikasi spesifik seperti pengenalan pola atau klasifikasi data melalui proses pembelajaran. Jaringan ini terdiri dari sejumlah besar elemen pemrosesan (neuron)

yang sangat saling berhubungan yang bekerja sama untuk memecahkan masalah tertentu. Aplikasi komersial dari teknologi ini biasanya berfokus pada pemecahan masalah pemrosesan sinyal atau pengenalan pola yang kompleks. Contoh aplikasi komersial yang signifikan sejak tahun 2000 meliputi pengenalan tulisan tangan untuk pemrosesan cek, transkripsi ucapan ke teks, analisis data, dan prakiraan cuaca serta pengenalan wajah. Jaringan saraf tiruan ini terinspirasi oleh hal-hal yang kita temukan sebagai manusia dalam biologi kita. Cara neural network mempelajari sesuatu sama dengan manusia yaitu learning by example.

Model jaringan saraf menggunakan prinsip matematika dan ilmu komputer untuk meniru proses otak manusia. Jaringan saraf tiruan mencoba mensimulasikan proses sel-sel otak yang terkait erat, tetapi alih-alih dibangun oleh biologi, mereka dibangun oleh kode, atau simpul sebagaimana mereka disebut. Ketiga konsep AI ini memungkinkan robot perangkat keras dan perangkat lunak untuk berpikir dan bertindak secara dinamis di luar kode yang telah ditentukan sebelumnya. Memahami konsep dasar di atas akan membawa AI ke masa depan yang lebih cerdas dari yang dibayangkan.

12.6 Teknologi *Artificial Intelligence* dan Manusia

teknologi AI Keunggulan teknologi kecerdasan buatan dibandingkan dengan manusia, teknologi kecerdasan buatan dapat bekerja tanpa mengenal lelah, mengutamakan efisiensi waktu, dan hasilnya berbeda. Kelemahannya adalah kurangnya kreativitas dan dikendalikan oleh manusia dan hanya melakukan tugas seperti yang diarahkan oleh pengembang. Manusia memiliki kemampuan kognitif dimana hasil yang diperoleh berbeda-beda sesuai dengan kreativitasnya. Namun manusia memiliki kelemahan yaitu dapat merasa lelah dan rentan terhadap human error (tugas ini memerlukan SOP yang

panjang). Kecerdasan buatan dalam meniru manusia Penglihatan mesin Ini berkaitan dengan bagaimana komputer meniru mata manusia dan memberikan wawasan sehingga komputer dapat memeriksa gambar dan menafsirkan objek di sekitarnya seperti yang dilakukan manusia. Kecerdasan buatan tiruan manusia memperkirakan apakah itu objek atau bukan berdasarkan deteksi objek. Deteksi AI diklasifikasikan berdasarkan apakah termasuk dalam kategori tertentu atau tidak. Jadi pendeteksian objek yang ada ada 2 model yaitu untuk menentukan bahwa sesuatu adalah objek dan untuk menentukan kelas citra yang ada.

Pengenalan ucapan Sintesis ucapan Teknik di mana kecerdasan buatan meniru suara manusia. Komputer dapat mengubah audio menjadi teks dan teks menjadi audio sehingga menjadi objek di speaker pintar, misalnya. Contoh penggunaannya antara lain *chat box*, *speech to text* dan Widya Wicara *text to speech*. Ini memudahkan pembuat konten yang ingin menjadikan teks sebagai suara alami untuk berita dan konten lainnya. Keputusan Kecerdasan buatan bertindak sebagai pemecah masalah atau pembuat keputusan. Faktanya, AI dapat memproses dan menganalisis kumpulan data jauh lebih cepat daripada otak manusia. Dengan demikian, AI dapat digunakan untuk mempercepat pengambilan keputusan berbasis data. Fitur ini menjadikan AI sebagai teknologi yang sangat berharga di banyak sektor bisnis. Terjemahan ucapan Ini merujuk pada bagaimana AI dapat meniru manusia dalam hal terjemahan bahasa. Misalnya, di Google Terjemahan. Google Terjemahan menjadi lebih baik setiap hari karena memiliki kecerdasan buatan bawaan.

12.7 Pendekatan Planning

Rencana bisnis adalah alat penting dalam semua jenis bisnis. Sama seperti rumah yang dibangun tanpa rencana atau film yang difilmkan tanpa naskah, tidak adanya rencana bisnis formal dalam dokumen tertulis dapat mempersulit dan membawa lebih banyak ketidakpastian bagi perusahaan dalam pelaksanaan rencana tersebut. Tidak peduli seberapa hati-hati, sistematis dan menarik rencana bisnis disiapkan dan disajikan, perbedaan sebenarnya terletak pada bagaimana rencana tersebut dapat mempengaruhi hasil dan kinerja bisnis. tidak hanya tertulis dalam dokumen, tetapi yang terpenting adalah kemampuan rencana untuk membantu pengusaha mencapai harapan bisnis. Sayangnya, pengusaha bukanlah peramal. Dalam beberapa tahun terakhir, ketidak mampuan untuk mengantisipasi perubahan yang cepat dan meminimalkan kegagalan yang sering terjadi pada rencana bisnis konvensional telah mendorong pengembangan pendekatan baru dalam perencanaan bisnis. Pendekatan *Lean Design* menghancurkan pendekatan perencanaan pemeliharaan tradisional, yang dianggap memakan waktu, boros, berisiko, tidak efisien, dan tidak efektif. *Lean Planning [Leon Planning]* adalah perencanaan bisnis modern, yang mengatakan bahwa menurut sudut pandang ini, titik awal bagi perusahaan adalah mengembangkan model bisnis, rencana awal, dan prototipe produk yang cocok untuk konsumen dan kelompok minat lainnya daripada membuat Bisnis Rencana. dokumen . Berbeda dengan pemahaman tradisional yang memulai perencanaan dengan dokumen formal yang kaku dan detail, startup tidak perlu menyiapkan dokumen business plan di awal, melainkan hanya menyiapkan dokumen seperlunya untuk meminimalisir kegagalan, dan rencana yang dibuat lebih fleksibel. untuk bereaksi komite dan pihak eksternal lainnya. (Aniesa,2018).

Pendekatan Planning adalah mempercayai perusahaan inovatif ini {start-up1 bukanlah versi kecil (misalnya dalam hal jumlah karyawan, besar modal, besar pendapatan) dari sebuah perusahaan besar. Namun, startup adalah desain organisasi sementara (Blank, 2013) di mana model bisnis, prototipe produk, dan rencana yang cocok untuk perusahaan besar atau kecil serta perusahaan baru atau perusahaan yang berbeda ditemukan dan diuji.

Setelah menyelesaikan tugas, startup dapat mengembangkan fase ekspansi bisnisnya dengan membuat perusahaan yang berfungsi dan menerapkan node, prototipe produk, dan rencana miliknya yang diuji oleh tim. Proses aplikasi ini dimulai dengan mengembangkan asumsi tentang kebutuhan konsumen atau pemangku kepentingan lainnya dan ide bisnis mereka, yang kemudian dituangkan ke dalam fondasi bisnis dengan membuat prototyping/betaing produk, menguji desain rencana untuk konsumen atau orang lain. Lingkaran dekat menguji produk dan rencana awal dan model bisnis langsung di pasar dan di sekitarnya, mengumpulkan umpan balik, jika terjadi kegagalan pasar dan lingkungan bisnis, pengusaha dapat melakukan iterasi (setiap langkah proses pengembangan rencana dapat dilakukan diulang secara siklis untuk meminimalkan kesalahan), diikuti oleh titik balik (perubahan di beberapa bagian rencana bisnis, tetapi masih berdasarkan rencana dan tujuan bisnis raksasa) model bisnis mereka berdasarkan kualifikasi konsumen atau pihak yang berkepentingan, (Aniesa,2018).

Model desain Xeon mengasumsikan bahwa perusahaan membawa rencana dan pihak terkait ke keadaan tetap J sebelum mencapai fase desain dan mencoba menghindari dokumentasi formal, sistematis, dan terperinci pada fase awal atau mundur hingga fase akhir. perusahaan menghadapi kegagalan dengan cepat dan kondisi terus belajar dan

mengembangkan produk dengan konsumen. Setelah menerima IDC dan rencana serta model bisnis yang terbukti Selama fase pengembangan, di mana dengan cepat memenuhi permintaan tambang, sewa meningkatkan penjualan dan pemasaran penjualan. Pada tahap ini, startup memerlukan pendanaan disebut juga rencana (pinjaman bank), hibah kompetitif dari pemerintah, investor individu investor sosial, investor sudut, dan pemodal ventura atau investor grup . (Aniesa,2018).

DAFTAR PUSTAKA

- Agustini, K. 2021. *Komunikasi Data dan Jaringan Komputer serta Analoginya dalam Konsep Subak - Rajawali Pers*. PT Rajagrafindo Persada.
- Rakhmad Fahmi Putra, Sumarno. 2022. Aplikasi Sistem Pakar Perencanaan Investasi Pasar Modal Menggunakan Metode Forward Chaining Berbasis Website Vol. 3 Desember
- Untung Rahardja. 2022. Masalah Etis dalam Penerapan Sistem Kecerdasan Buatan Fakultas Sains dan Teknologi Technomedia Journal (TMJ) p-ISSN: 2620-3383 Vol. 7 No. 2 Oktober 2022
- Soni Agung Wahyudiyanta. 2023. Pengendalian Persediaan Produk Kecerdasan Buatan Product Inventory control Artificial intelligence ISSN: 2962-3545 Prosiding SAINTTEK: Sains dan Teknologi Vol.2 No.1 Tahun 2023
- Kevin Angelino, Jenny, Lina, Hendri Wijaya, Tondy. 2023. Analisis *Supply Chain Management* dan Enterprise Resource Planning pada Perusahaan H&M Fendy Cuandra6 VOL 1 NO 6 E-ISSN : 2809-1612, P-ISSN : 2809-1620
- Novi Manunggal, Imam Teguh Santoso, Sigit Wicaksana. 2022. Pengaruh Sistem Informasi Sumber Daya Manusia (HRIS) dan Kecerdasan Buatan Terhadap Kinerja Industri Pertahanan Journal of Industrial Engineering & Management Research Vol.3 No.6 <http://www.jiemar.org> DOI: <https://doi.org/10.7777/jiemar> e-ISSN : 2722-8878.2021
- <https://idmetafora.com/id/blog/read/318/Mengenal-3-Konsep-Dasar-Kecerdasan-Buatan-Artificial-Intelligence.html>

<https://widyaanalytic.com/implementasi-artificial-intelligence-dalam-meniru-manusia/>
Kecerdasan Buatan. Hanson Prihantoro Putro, S.T., M.T. Terbit
Februari. 2023. ISBN : 978-623-09-2088-2(PDF).
Perencanaan Bisnis pariwisata (Pendekatan Lean Planning).
Aniesa Samira Bafadhal. ISBN 978-602-432-4. 978-602-
432-596-1 (elektronik) Terbit November 2018.

BAB 13

LEARNING

Oleh Risanto Darmawan

13.1 Pendahuluan

Mirip dengan manusia, program komputer juga belajar dengan cara yang berbeda. Berbicara tentang AI (*Artifisial Intelligence*) Kecerdasan Buatan, *learning* (pembelajaran) dengan platform ini selanjutnya dipisahkan ke dalam berbagai bentuk. Salah satu komponen penting dari AI, pembelajaran untuk AI mencakup metode trial-and-error. Solusinya terus memecahkan masalah sampai menemukan hasil yang tepat. Dengan cara ini, program mencatat semua gerakan yang memberikan hasil positif dan menyimpannya dalam basis datanya untuk digunakan saat berikutnya komputer diberikan masalah yang sama.

Komponen pembelajaran AI termasuk menghafal item individu seperti solusi yang berbeda untuk masalah, kosa kata, bahasa asing, dll., Juga dikenal sebagai pembelajaran hafalan. Metode pembelajaran ini kemudian diimplementasikan dengan menggunakan metode generalisasi.

Dalam Konteks Kecerdasan Buatan *AI Digital*, pembelajaran adalah salah satu konsep yang sangat penting. Dalam pembelajaran AI Digital, sistem komputer belajar dari data atau pengalaman yang ada dan menggunakan informasi ini untuk meningkatkan kapasitas sistem dalam memproses informasi, membuat keputusan, dan menyelesaikan tugas (Shalev-Shwartz, 2014).

Learning pada AI Digital dapat dianggap sebagai proses adaptasi, di mana sistem secara bertahap meningkatkan

kemampuan dan pengetahuannya dalam mengenali pola-pola, memproses informasi, dan mengambil keputusan berdasarkan data. Dalam *Learning AI Digital*, teknologi seperti Machine Learning, Deep Learning, Natural Language Processing, dan Computer Vision sering digunakan untuk menciptakan sistem yang dapat memproses informasi dan belajar secara independen.

Ada beberapa teknik pembelajaran (*Learning*) yang digunakan dalam AI Digital, seperti Supervised Learning, Unsupervised Learning, Semi Supervised Learning dan Reinforcement Learning. Masing-masing teknik ini memiliki keunggulan dan kelemahan tergantung pada jenis data atau tugas yang dihadapi.

Dalam pengembangan AI Digital, tujuan akhirnya adalah untuk menciptakan sistem yang mampu belajar secara independen dari data atau pengalaman yang ada, sehingga dapat mengambil keputusan dan bertindak secara cerdas dan adaptif. Dengan demikian, pembelajaran merupakan aspek kunci dari pengembangan AI Digital yang dapat menghasilkan solusi cerdas dan efektif untuk berbagai masalah yang kompleks.

13.2 Konsep Dasar

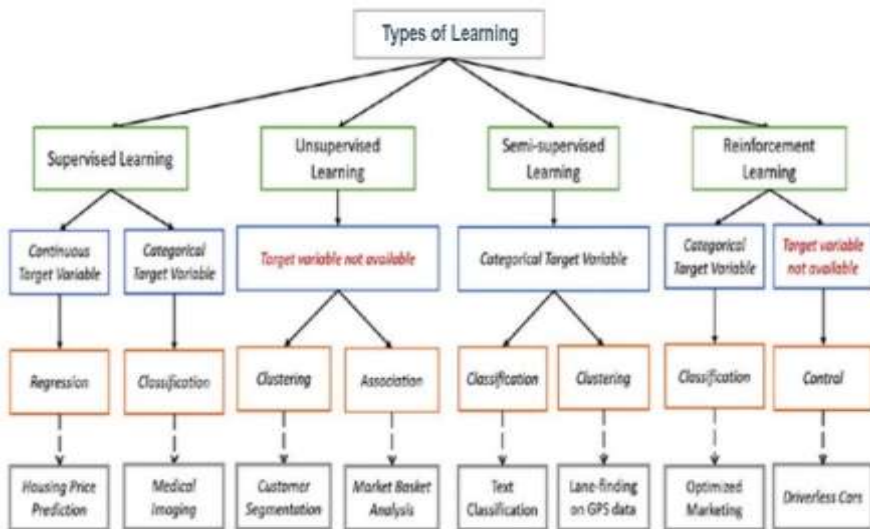
Pembelajaran pada Kecerdasan Buatan Digital (AI Digital) dapat didefinisikan sebagai suatu proses di mana sistem komputer belajar dari data atau pengalaman yang ada dan menggunakan informasi tersebut untuk meningkatkan kemampuan sistem dalam memproses informasi, membuat keputusan, dan menyelesaikan tugas. Tujuan dari pembelajaran pada AI Digital adalah untuk menciptakan sistem yang dapat memproses informasi dan belajar secara mandiri, sehingga dapat mengambil keputusan dan bertindak secara cerdas dan adaptif (Shalev-Shwartz, 2014).

Ada beberapa alasan mengapa pembelajaran merupakan aspek kunci dalam pengembangan AI Digital. Pertama, dengan belajar dari data atau pengalaman, sistem komputer dapat meningkatkan kemampuan dan pengetahuannya dalam mengenali pola-pola, memproses informasi, dan mengambil keputusan berdasarkan data. Kedua, pembelajaran memungkinkan sistem untuk menyesuaikan diri dengan lingkungan dan tugas yang berbeda-beda, sehingga dapat bekerja dengan efektif dalam berbagai situasi. Ketiga, pembelajaran juga dapat membantu sistem untuk mengoptimalkan performa dan meningkatkan akurasi dalam menyelesaikan tugas-tugas yang kompleks.

Dalam pembelajaran pada AI, terdapat empat jenis utama teknik pembelajaran, yaitu *Supervised Learning*, *Unsupervised Learning*, dan *Reinforcement Learning*, *Reinforcement Learning*. Setiap teknik memiliki pendekatan dan karakteristik yang berbeda, tergantung pada jenis data atau tugas yang dihadapi.

13.3 Teknik Pembelajaran

Ada beberapa teknik pembelajaran pada AI, seperti *Supervised Learning*, *Unsupervised Learning*, *Semi Supervised Learning* dan *Reinforcement Learning* (Alpaydın, 2010).



Gambar 13.1. Teknik pembelajaran

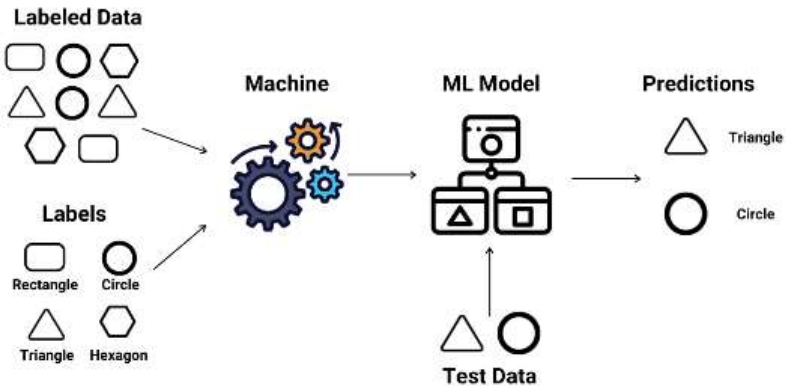
Sumber (towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861)

Ini konsep, kelebihan, dan kekurangan dari masing-masing teknik tersebut :

13.3.1 *Supervised Learning*

Teknik pembelajaran *Supervised Learning* adalah teknik di mana sistem komputer belajar dari data yang sudah diketahui label atau kategori yang tepat. Proses pembelajaran dimulai dengan memberikan data training yang sudah dilabeli kepada sistem, kemudian sistem akan mencari pola-pola dan hubungan antar variabel dalam data tersebut (Theobald, 2017).

Supervised Learning



Gambar 13.2. Supervised Learning

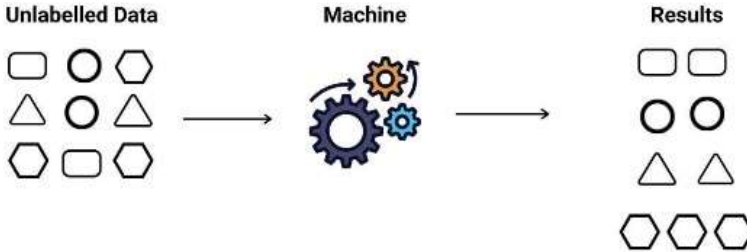
Sumber (www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning)

Setelah sistem berhasil menemukan pola-pola dan hubungan tersebut, maka sistem dapat digunakan untuk memprediksi kategori atau label dari data yang belum diketahui labelnya. Contoh aplikasi dari teknik *Supervised Learning* adalah klasifikasi gambar, pengenalan suara, atau prediksi harga saham.

13.3.2 Unsupervised Learning

Teknik pembelajaran *Unsupervised Learning* adalah teknik di mana sistem komputer belajar dari data yang tidak memiliki label atau kategori yang jelas. Proses pembelajaran dimulai dengan memberikan data training yang tidak memiliki label atau kategori kepada sistem, kemudian sistem akan mencari pola-pola dan struktur yang tersembunyi dalam data tersebut (Alpaydın, 2010).

Unsupervised Learning



Gambar 13.3. Unsupervised Learning

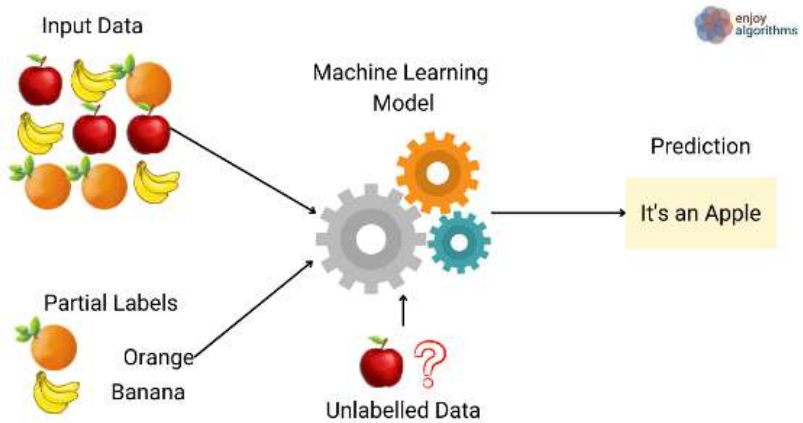
Sumber (www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning)

Setelah sistem berhasil menemukan pola-pola dan struktur tersebut, maka sistem dapat digunakan untuk melakukan klasifikasi atau pengelompokan data yang memiliki karakteristik yang sama.

Contoh aplikasi dari teknik *Unsupervised Learning* adalah klasifikasi dokumen, segmentasi gambar, atau penemuan pola dalam data sensor.

13.3.3 Semi-supervised learning

Semi-supervised learning adalah kombinasi dari *supervised learning* dan *unsupervised learning*. Di dalam *semi-supervised learning*, model diberikan sebagian data yang diberi label (*supervised*) dan sebagian data tidak diberi label (*unsupervised*) (Alpaydm, 2010). Tujuannya adalah untuk meningkatkan kinerja model pada data yang tidak diberi label dengan memanfaatkan informasi yang diperoleh dari data yang diberi label.

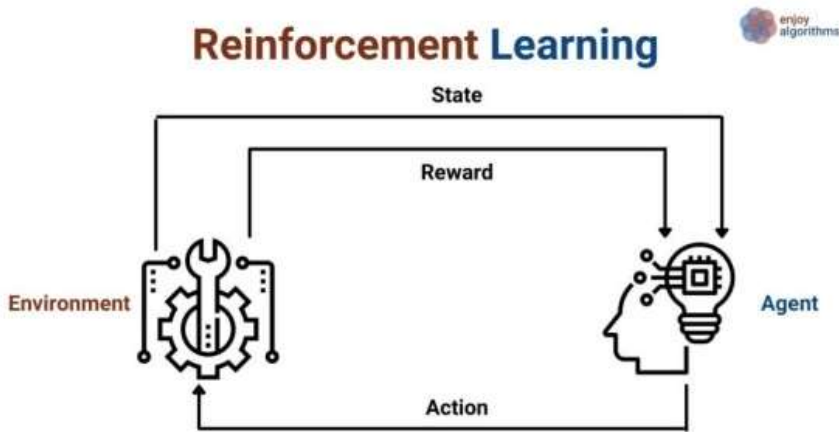


Gambar 13.4. Semi Supervised Learning

Sumber (www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning)

13.3.4 Reinforcement Learning

Reinforcement Learning adalah teknik di mana sistem komputer belajar melalui interaksi dengan lingkungannya. Proses pembelajaran dimulai dengan memberikan sistem komputer suatu tugas atau tujuan yang harus dicapai, kemudian sistem akan melakukan tindakan-tindakan yang menghasilkan suatu reward atau hukuman dari lingkungan. Setelah sistem berhasil mencapai tujuan atau tugas tersebut, maka sistem akan mendapatkan reward yang sesuai (Mitchell, 1997). Contoh aplikasi dari teknik Reinforcement Learning adalah game AI, kontrol robot, atau pengoptimalan strategi bisnis.



Gambar 13.5. Reinforcement Learning

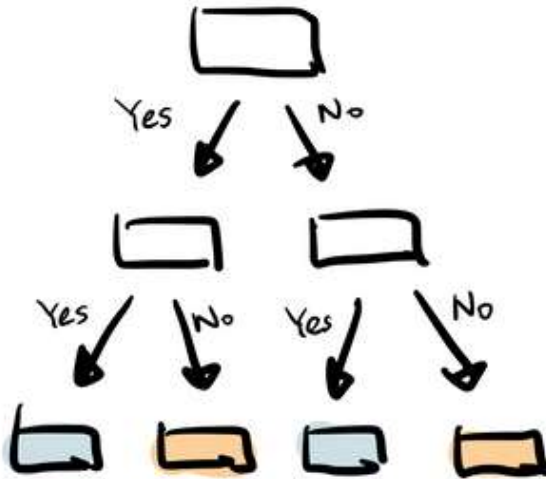
Sumber (www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning)

13.4 Algoritma Pembelajaran

Algoritma pembelajaran pada merupakan kumpulan instruksi atau aturan yang digunakan oleh sistem komputer untuk memproses data dan belajar dari data tersebut. Algoritma pembelajaran dapat dibagi menjadi beberapa jenis, tergantung pada pendekatan dan karakteristiknya. Berikut ini adalah beberapa jenis algoritma pembelajaran :

13.4.1 Decision Tree

Algoritma Decision Tree merupakan algoritma yang menggunakan struktur pohon keputusan untuk memprediksi nilai dari suatu variabel target (Theobald, 2017).



Gambar 13.6. Decision Tree

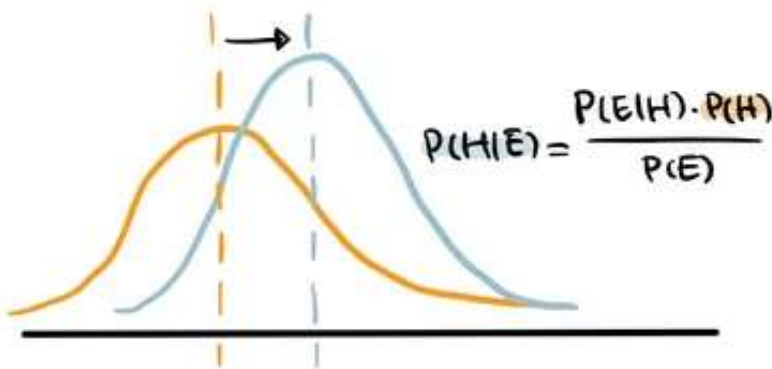
Sumber (towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501)

Algoritma ini cocok untuk data dengan variabel target kategori atau diskrit. Proses pembelajaran pada algoritma *Decision Tree* dimulai dengan membagi data ke dalam subset-subset yang lebih kecil, kemudian memilih variabel terbaik yang dapat membagi data dengan cara yang paling efisien.

13.4.2 Naive Bayes

Algoritma Naive Bayes merupakan algoritma yang menggunakan teorema Bayes untuk memprediksi nilai dari suatu variabel target (Mitchell, 1997). Algoritma ini cocok untuk data dengan variabel target kategori atau diskrit. Proses pembelajaran pada algoritma Naive Bayes melibatkan perhitungan probabilitas dari masing-masing variabel dalam

data dan kemudian menghitung probabilitas kondisional dari setiap kategori atau label.



Gambar 13.7. Naïve Bayes

Sumber (towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501)

Teori Bayes bekerja untuk mencapai hipotesis (H) dari kumpulan bukti (E) yang diberikan. Teori ini berkaitan dengan dua aspek: probabilitas hipotesis sebelum bukti $P(H)$ dan probabilitas hipotesis setelah bukti $P(H|E)$. Persamaan berikut menjelaskan Teori Bayes:

$$P(H|E) = (P(E|H) * P(H)) / P(E)$$

Dalam persamaan tersebut:

$P(H|E)$ menunjukkan probabilitas peristiwa H terjadi ketika peristiwa E terjadi.

$P(E|H)$ menyatakan seberapa sering peristiwa E terjadi ketika peristiwa H terjadi terlebih dahulu.

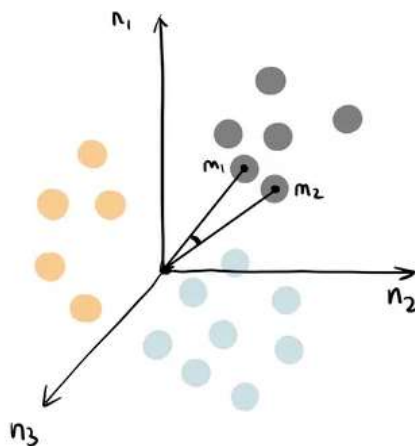
$P(H)$ merepresentasikan probabilitas kejadian X terjadi secara mandiri.

$P(E)$ merepresentasikan probabilitas kejadian Y terjadi secara mandiri.

Aturan Bayes adalah cara untuk menentukan $P(H|E)$ dari $P(E|H)$. Dengan kata lain, ini memberikan cara untuk menghitung probabilitas hipotesis dengan bukti yang diberikan.

13.4.3 K-Nearest Neighbors

Algoritma *K-Nearest Neighbors* (KNN) merupakan algoritma yang memprediksi nilai dari suatu variabel target dengan mempertimbangkan nilai dari variabel-variabel terdekat dalam data (Theobald, 2017). Algoritma ini cocok untuk data dengan variabel target kontinu atau diskrit. Proses pembelajaran pada algoritma KNN melibatkan perhitungan jarak antara setiap data dan kemudian mencari data dengan jarak terdekat.

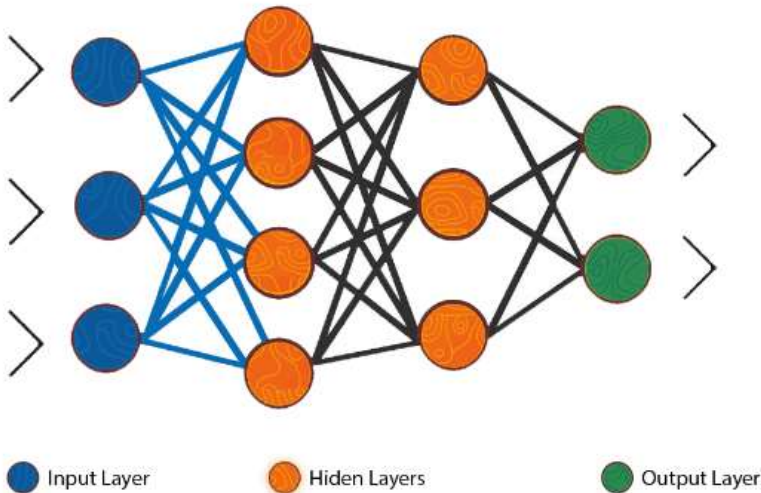


Gambar 13.8. KNN

Sumber (towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501)

13.4.4 Artificial Neural Networks

Algoritma Artificial Neural Networks (ANN) merupakan algoritma yang terinspirasi dari struktur dan fungsi otak manusia. Algoritma ini cocok untuk data dengan variabel target kontinu atau diskrit. Proses pembelajaran pada algoritma ANN melibatkan simulasi dari jaringan neuron dalam otak manusia dan pembentukan bobot atau koneksi antara setiap neuron (Theobald, 2017).



Gambar 13.9. ANN

Sumber

(www.analyticsvidhya.com/blog/2021/07/understanding-the-basics-of-artificial-neural-network-ann)

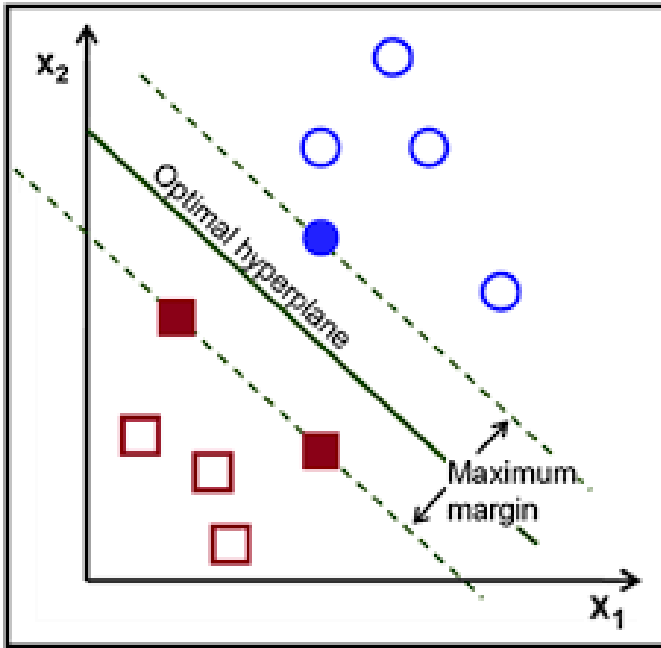
Jaringan saraf yang paling murni terdiri dari tiga lapisan, yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Lapisan masukan berfungsi untuk menerima sinyal masukan dan mentransfernya ke lapisan berikutnya,

sedangkan lapisan keluaran memberikan hasil akhir atau prediksi. Sebelum mampu menyelesaikan masalah tertentu, jaringan saraf harus dilatih dengan menggunakan beberapa data pelatihan dan algoritma pembelajaran mesin.

13.4.5 *Support Vector Machines*

Algoritma *Support Vector Machines* (SVM) merupakan algoritma yang memprediksi nilai dari suatu variabel target dengan mencari hyperplane terbaik yang dapat memisahkan data ke dalam kelas-kelas yang berbeda (Shalev-Shwartz, 2014). Algoritma ini cocok untuk data dengan variabel target kontinu atau diskrit. Proses pembelajaran pada algoritma SVM digunakan untuk mencari hyperplane terbaik dengan memaksimalkan jarak antar kelas. Hyperplane adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai line whereas, fungsi yang digunakan untuk klasifikasi antar kelas dalam 3-D disebut plane similarly, sedangkan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut hyperplane.

Ketika memilih algoritma pembelajaran pada AI Digital, faktor-faktor seperti jenis data, ukuran data, dan tujuan dari pembelajaran harus dipertimbangkan. Algoritma yang tepat dapat membantu meningkatkan akurasi dan performa sistem komputer dalam memproses informasi dan membuat keputusan. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan linear maupun non linear.



Gambar 13.10. SVN

Sumber (Awad Mariette, Kanna Rahul, 2015)

13.5 Aplikasi Pembelajaran

Aplikasi pembelajaran pada AI Digital merujuk pada penggunaan algoritma pembelajaran mesin dalam berbagai bidang, seperti pengenalan wajah, deteksi objek, penerjemahan bahasa, dan banyak lagi (Hurwitz, 2018). Beberapa contoh aplikasi pembelajaran:

1. **Pengenalan Suara:** Aplikasi ini dapat digunakan untuk mengenali suara manusia dan membedakan antara suara berbeda atau bahkan bahasa yang berbeda. Contohnya adalah asisten virtual seperti Siri, Alexa, atau Google Assistant yang dapat memahami dan merespons perintah suara pengguna.

2. **Deteksi Objek:** Aplikasi ini digunakan untuk mendeteksi objek dalam gambar atau video, seperti kendaraan, orang, atau binatang. Contohnya adalah sistem keamanan di bandara atau toko yang menggunakan kamera untuk mendeteksi objek yang mencurigakan atau penggunaan dalam industri otomotif untuk mendeteksi bahaya dan memperingatkan pengemudi.
3. **Penerjemahan Bahasa:** Aplikasi ini dapat digunakan untuk menerjemahkan bahasa dari satu bahasa ke bahasa lain. Contohnya adalah Google Translate yang dapat menerjemahkan teks atau ucapan dari bahasa Inggris ke bahasa lain, dan sebaliknya.
4. **Diagnosis Medis:** Aplikasi ini dapat membantu dokter dalam mendiagnosis penyakit dengan mengidentifikasi gejala dan menghubungkannya dengan data medis. Contohnya adalah sistem deteksi kanker payudara atau sistem pendeteksi retinopati diabetik yang dapat membantu dokter dalam mendiagnosis penyakit dan memberikan perawatan yang tepat.
5. **Pemrosesan Bahasa Alami:** Aplikasi ini digunakan untuk memahami dan memproses bahasa manusia, termasuk teks dan ucapan. Contohnya adalah chatbot yang dapat memahami pertanyaan dan memberikan jawaban yang sesuai atau sistem analisis sentimen yang dapat menganalisis perasaan dari teks yang ditulis.

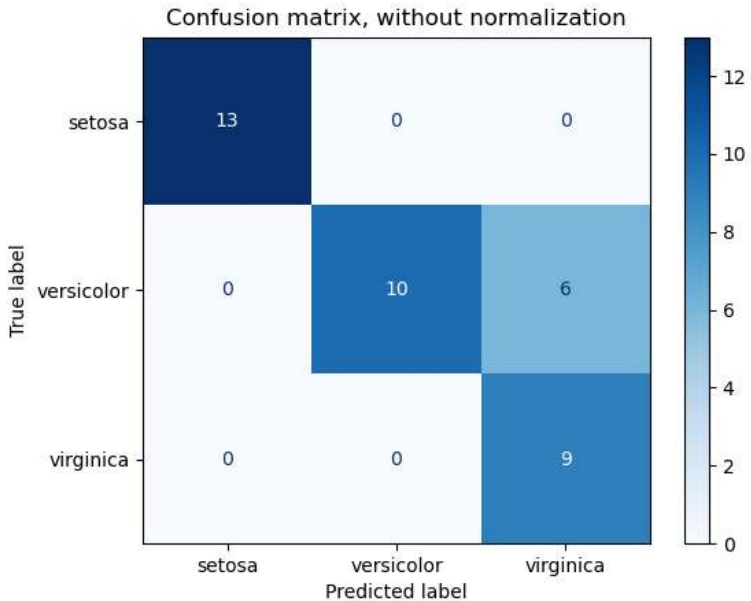
Dalam pengembangan aplikasi pembelajaran pada AI Digital, penting untuk memperhatikan keamanan dan privasi data. Selain itu, perlu juga diingat bahwa aplikasi pembelajaran pada AI Digital bukanlah solusi yang sempurna dan masih memiliki kelemahan, seperti kesalahan identifikasi atau ketidakakuratan dalam prediksi. Oleh karena itu, perlu dilakukan pengujian yang baik dan pemantauan terus-menerus

untuk meningkatkan performa aplikasi dan mengatasi masalah yang muncul

13.6 Evaluasi Pembelajaran

Evaluasi pembelajaran pada AI Digital merupakan tahap penting dalam pengembangan sistem kecerdasan buatan. Evaluasi ini dilakukan untuk mengetahui sejauh mana performa sistem yang telah dibuat dan apakah memenuhi tujuan yang diharapkan. Evaluasi pembelajaran pada AI Digital dapat dilakukan dengan beberapa cara, antara lain:

1. **Confusion Matrix:** merupakan metode evaluasi yang dapat digunakan untuk menghitung kinerja atau tingkat kebenaran dari proses klasifikasi. Confusion matrix merupakan tabel yang menunjukkan jumlah prediksi yang benar atau salah untuk setiap kelas atau label. Confusion matrix dapat digunakan untuk mengukur akurasi, presisi, recall dan sebagainya.



Gambar 13.11. Confusion Matrix

Sumber (scikit-

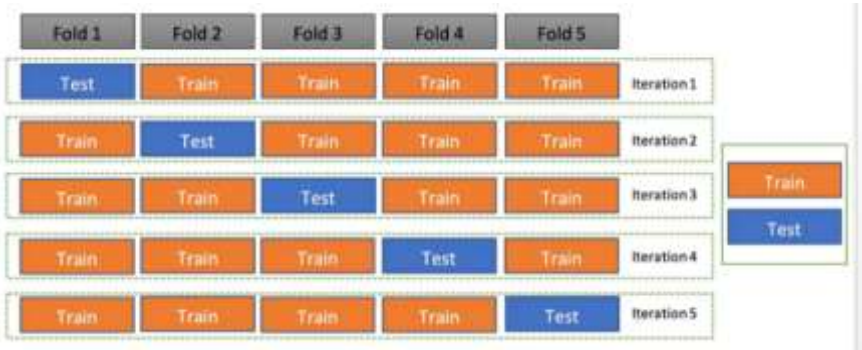
learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html)

Berdasarkan tabel Confusion Matrix, dapat dilakukan perhitungan untuk mendapatkan nilai akurasi, presisi, dan recall.

Akurasi adalah cara pengujian yang didasarkan pada seberapa dekat nilai prediksi dengan nilai aktual. Presisi adalah cara pengujian yang membandingkan jumlah informasi relevan yang diberikan oleh sistem dengan jumlah informasi yang diambil oleh sistem, baik relevan maupun tidak relevan. Recall adalah cara pengujian yang membandingkan jumlah informasi relevan yang diberikan oleh sistem dengan jumlah informasi relevan yang ada

dalam koleksi informasi, baik yang diambil atau tidak diambil oleh sistem.

- 2. **Cross Validation:** Cross validation merupakan teknik statistik yang dipakai untuk mengestimasi kinerja (atau ketepatan) model pembelajaran mesin. Metode ini dimanfaatkan untuk mencegah overfitting pada model prediktif, terutama pada situasi di mana jumlah data terbatas. Dalam validasi silang, kita membagi data ke dalam beberapa fold (*subsets*) yang sama besar, melakukan analisis pada setiap fold, dan menghitung rata-rata dari kesalahan prediksi secara keseluruhan (Mohri, 2018).

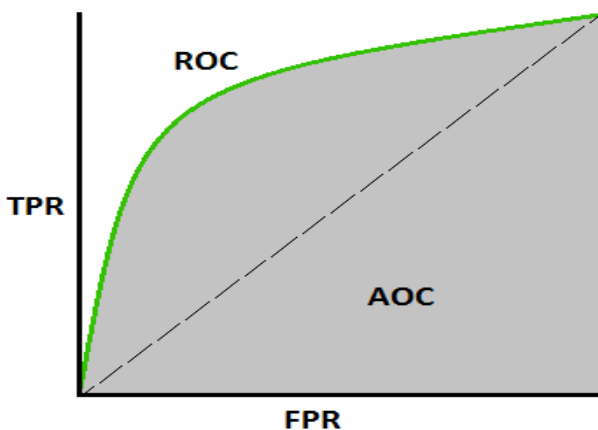


Gambar 13.12. Cross Validation

Sumber (sqlrelease.com/introduction-to-k-fold-cross-validation-in-python)

Cross validation digunakan untuk mengevaluasi performa model pada data yang tidak digunakan saat pelatihan. Metode *cross validation* ini melatih model dengan menggunakannya secara bergantian pada setiap bagian. Kemudian, model diuji pada bagian yang tidak digunakan.

3. **AUC-ROC Curve:** Kurva AUC-ROC digunakan untuk mengevaluasi performa model klasifikasi pada dataset yang tidak seimbang atau dataset dengan jumlah kelas yang tidak merata. AUC-ROC curve menunjukkan seberapa baik model dapat membedakan antara kelas positif dan negatif.



Gambar 13.13. AUC - ROC Curve

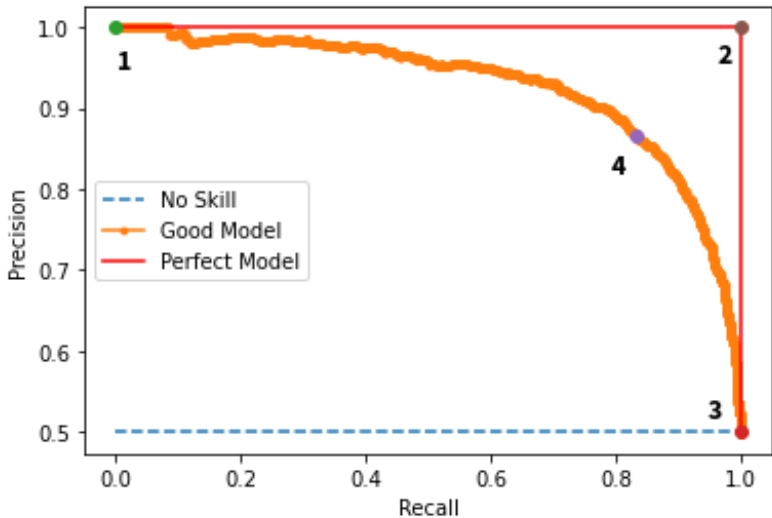
Sumber (towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)

Kurva AUC-ROC merupakan evaluasi kinerja untuk klasifikasi pada berbagai nilai ambang. ROC adalah grafik probabilitas dan AUC mengindikasikan tingkat atau ukuran diskriminasi. Ini menunjukkan kemampuan model untuk membedakan antara kelas. Semakin tinggi nilai AUC, semakin akurat model dalam memprediksi kelas 0 sebagai 0 dan kelas 1 sebagai 1.

Dalam analogi, semakin tinggi nilai AUC, semakin baik model dalam membedakan pasien dengan penyakit dan tanpa penyakit. Kurva AUC-ROC digambarkan dengan TPR terhadap FPR di mana TPR terletak pada sumbu y dan FPR pada sumbu x.

4. ***Precision-Recall Curve***: Precision-Recall curve digunakan untuk mengevaluasi performa model klasifikasi pada dataset yang tidak seimbang atau dataset dengan jumlah kelas yang tidak merata. Precision-Recall curve menunjukkan seberapa baik model dapat membedakan antara kelas positif dan negatif berdasarkan precision dan recall.

Precision dan *Recall* merupakan dua ukuran yang sangat penting dalam mengevaluasi model klasifikasi yang tidak seimbang. Hal ini membantu kita untuk mengetahui seberapa banyak dari hasil positif yang benar-benar terklasifikasi dengan tepat, dan di antara yang terklasifikasi positif, seberapa banyak yang benar-benar positif.



Gambar 13.14. AUC - ROC Curve

Sumber (towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)

5. **Mean Squared Error (MSE):** MSE digunakan untuk mengevaluasi performa model pada masalah regresi. Metode ini menghitung selisih antara nilai prediksi dengan nilai sebenarnya, kemudian menghitung rata-rata dari selisih kuadrat (Shalev-Shwartz, 2014).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Mean Error Squared

Gambar 13.15. Mean Squared Error

Sumber (suboptimal.wiki/explanation/mse/)

Pengukuran kesalahan kuadrat memperhitungkan perbedaan antara nilai yang diharapkan dan nilai aktual untuk mengevaluasi kesalahan pada setiap baris. Total kesalahan model, yang disebut mean squared error (MSE), dapat memberikan gambaran tentang kinerja model secara keseluruhan.

Keuntungan utama dari MSE adalah kemampuannya untuk menyoroti atau menghukum kesalahan yang signifikan dengan cara mengkuadratkan kesalahan tersebut. Oleh karena itu, MSE sangat berguna dalam mengurangi potensi kesalahan besar yang mungkin terjadi selama pengembangan model.

6. ***Mean Absolute Error (MAE)***: adalah rata-rata selisih mutlak nilai sebenarnya (aktual) dengan nilai prediksi (peramalan). MAE digunakan untuk mengukur keakuratan suatu model statistik dalam melakukan prediksi atau peramalan MAE adalah metode evaluasi yang umum digunakan dalam data science. MAE menghitung rata-rata dari selisih absolut antara nilai prediksi dan nilai aktual (Theobald, 2017).

Dengan kata lain, MAE menghitung berapa rata-rata kesalahan absolut dalam prediksi. Semakin kecil nilai MAE, semakin baik kualitas model tersebut

The diagram illustrates the Mean Absolute Error (MAE) formula: $MAE = \frac{1}{n} \sum |y - \hat{y}|$. Annotations include:

- A blue box around $\frac{1}{n}$ with the text "Divide by the total number of data points".
- A green box around y with the text "Actual output value".
- A yellow box around $\hat{y}$ with the text "= predicted output value".
- A bracket under the absolute value term $|y - \hat{y}|$ with the text "The absolute value of the residual".
- A blue arrow points from the summation symbol $\sum$ to the word "Sum of".

Gambar 13.16. Mean Absolute Error

Sumber (stackoverflow.com/questions/56401346/mean-absolute-error-in-tensorflow-without-built-in-functions)

7. **Root Mean Squared Error (RMSE):** RMSE adalah turunan dari MSE. Seperti namanya, RMSE adalah akar kuadrat dari MSE.

RMSE menghitung rata-rata dari selisih kuadrat antara nilai prediksi dan nilai aktual kemudian diambil akar kuadratnya. Semakin kecil nilai RMSE, semakin baik kualitas model tersebut (Mitchell, 1997).

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}$$

Gambar 13.17. Root Mean Square Error

Sumber (towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e)

Standar deviasi residual (kesalahan prediksi). Residu adalah ukuran seberapa jauh dari titik data garis regresi; RMSE adalah ukuran seberapa tersebar sisa-sisa

ini. Dengan kata lain, ini memberi tahu Anda seberapa terkonsentrasi data di sekitar garis yang paling sesuai. Root mean square error umumnya digunakan dalam klimatologi, peramalan, dan analisis regresi untuk memverifikasi hasil eksperimen.

8. **Mean Absolute Percentage Error (MAPE):** MAPE adalah metode evaluasi lain yang digunakan dalam data science. MAPE menghitung rata-rata dari selisih persentase antara nilai prediksi dan nilai aktual (Trivusi, 2023). Dengan kata lain, MAPE menghitung berapa rata-rata kesalahan dalam prediksi sebagai persentase dari nilai aktual. Semakin kecil nilai MAPE, semakin baik kualitas model tersebut.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|A_i - F_i|}{A_i}$$

Gambar 13.18. Mean Absolute Percentage Error
Sumber (www.geeksforgeeks.org/how-to-calculate-mean-absolute-percentage-error-in-excel/)

MAPE merupakan rata-rata kesalahan persentase absolut yang mengukur kesalahan relatif secara proporsional terhadap perpanjangan sederhana dari varians absolut dalam satuan persentase, dan bukan dalam satuan variabel. Penggunaan nilai absolut pada MAPE mencegah kesalahan positif dan negatif saling menghapus satu sama lain, sementara penggunaan kesalahan relatif memungkinkan perbandingan akurasi perkiraan antara model deret waktu.

DAFTAR PUSTAKA

- Alpaydm Ethem, 2010. Introduction to Machine Learning Second Edition, The MIT Press Cambridge, Massachusetts
- Awad Mariette, Khanna Rahul, 2015. Efficient Learning Machines, ApressOpen
- Hurwitz Judith, Kirsch Daniel, 2018, Machine Learning for dummies, IBM Limited Edition, John Wiley & Sons, Inc.
- Mitchell Tom M. 1997, Machine Learning, McGraw-Hill Science/Engineering/Math
- Mohri Mechryar, Rostamizadeh Afshin, Talwalkar Ameet, 2018, Foundation of Machine Learning second edition, Massachusetts Institute of Technology
- Raharjo Budi, 2021, Pembelajaran Mesin (Machine Learning), Yayasan Prima Agus Teknik
- Robandi Imam. 2019. Artifisial Intelligence, mengupas Rekayasa Kecerdasan Tiruan, Yogyakarta, Penerbit Andi.
- Shalev-Shwartz Shai, Ben-David Shai , 2014, Understanding Machine Learning , From Theory to Algorithms, Cambridge University Press.
- Suyanto. 2014. Artifisial Intelligence, Searching, Reasoning, Planning, Learning. Bandung, Penerbit Informatika.
- Suyanto, dkk. 2022. Explainable Artificial Intelligence, menggunakan metode-metode berbasis Nearest Neighbors, Bandung, Penerbit Informatika.
- Suyanto, 2018. Machine Learning, Tingkat dasar dan lanjut, Bandung. Penerbit Informatika.
- Theobald Oliver, 2017, Machine Learning For Absolute Beginners

Websites :

Sumeet Bansal, 2021, How Do Artificial Intelligence & Different Components of AI Work?

<https://www.analytixlabs.co.in/blog/components-of-artificial-intelligence/> diakses pada April 2023

Trivusi , 2023, Perbedaan MAE, MSE, RMSE, dan MAPE pada Data Science

<https://www.trivusi.web.id/2023/03/perbedaan-mae-mse-rmse-dan-mape.html> diakses pada Mei 2023

SUBOPTIMaL (Seriously Unnecessary Baffling Obscure Perplexing Terms In Machine Learning)

<https://suboptimal.wiki/category/machine-learning-terms> diakses pada Mei 2023

Statistics How To, Regression Analysis

<https://www.statisticshowto.com/probability-and-statistics/regression-analysis/>

diakses pada Mei 2023

BAB 14

ALGORITMA GENETIKA

Oleh Suwito Pomalingo

14.1 Pengantar Algoritma Genetika

Algoritma Genetika atau *Genetic Algorithm* (GA) adalah sebuah teknik yang memadukan prinsip-prinsip evolusi biologis dengan kecerdasan buatan untuk menciptakan solusi inovatif bagi berbagai masalah yang sulit dipecahkan (Sivanandam and Deepa, 2008). Bab ini akan membahas tentang penjelasan dan penerapan algoritma genetika sebagai salah satu pendekatan terkemuka dalam optimasi dan pencarian. Selain itu akan diberikan contoh praktis yang diharapkan dapat membantu dalam memahami dasar-dasar algoritma genetika, serta potensi penerapan teknik ini dalam berbagai bidang.

14.1.1 Sejarah dan Evolusi Algoritma Genetika

Penelitian tentang Algoritma Genetika (GA) dimulai pada tahun 1960-an oleh seorang ilmuwan dan profesor di University of Michigan bernama John Holland. Holland adalah orang pertama yang secara formal mengembangkan konsep algoritma genetika melalui bukunya yang berjudul "*Adaptation in Natural and Artificial Systems*" yang terbit tahun 1975. Buku ini kemudian menjadi referensi awal dan utama dalam pengembangan algoritma genetika sebagai metode baru untuk optimasi dan pencarian.

Sejak diperkenalkannya konsep algoritma genetika oleh Holland, banyak peneliti lain yang juga mengembangkan dan menyempurnakan teknik ini. Tahun 1989, David E. Goldberg mempublikasikan buku "*Genetic Algorithms in Search,*

Optimization, and Machine Learning" yang menjadi salah satu referensi penting dalam bidang ini. Buku ini membahas berbagai aspek GA, termasuk representasi, operasi genetik, dan implementasi dalam berbagai aplikasi (Goldberg, 1989).

Kemudian tahun 1992, Koza mengenalkan konsep baru yang dikenal sebagai "*Genetic Programming*" (GP). Dalam bukunya yang berjudul "*Genetic Programming: On the Programming of Computers by Means of Natural Selection*," Koza menunjukkan bagaimana algoritma genetika dapat digunakan untuk menghasilkan program komputer yang mampu menyelesaikan masalah secara otomatis (Koza, 1992).

Selama beberapa dekade terakhir, GA telah mengalami evolusi yang signifikan dan menjadi salah satu teknik optimasi heuristik yang paling populer dan efektif. Teknik ini telah diaplikasikan dalam berbagai bidang, seperti kecerdasan buatan, perencanaan jadwal, optimasi industri, dan teknologi medis.

14.1.2 Prinsip Dasar Algoritma Genetika

Algoritma Genetika (GA) pada dasarnya adalah teknik optimasi dan pencarian heuristik yang terinspirasi oleh proses evolusi biologis. Prinsip dasar GA meliputi representasi, seleksi, *crossover* (reproduksi), dan mutasi. Berikut uraian singkat tentang prinsip-prinsip dasar GA:

- 1) Representasi: Dalam GA, solusi potensial dari masalah diwujudkan sebagai "kromosom" yang terdiri dari rangkaian "gen." Representasi ini bisa berupa string biner, karakter, atau bilangan, tergantung pada masalah yang dihadapi (Goldberg, 1989).
- 2) Seleksi: Seleksi merupakan proses memilih individu terbaik dari populasi berdasarkan tingkat kebugaran mereka. Individu yang lebih baik memiliki peluang lebih tinggi untuk dipilih dan menghasilkan keturunan. Seleksi

bertujuan untuk menggantikan individu yang kurang baik dengan individu yang lebih baik (Holland, 1992).

- 3) *Crossover* (Reproduksi): *Crossover* adalah proses menggabungkan informasi genetik dari dua orang tua untuk menghasilkan satu atau lebih individu keturunan baru. Teknik *crossover* yang umum digunakan meliputi *crossover* satu titik, *crossover* dua titik, dan *crossover uniform* (Goldberg, 1989).
- 4) Mutasi: Mutasi adalah proses mengubah informasi genetik pada kromosom secara acak. Tujuan mutasi adalah untuk mempertahankan keanekaragaman genetik dalam populasi dan mencegah terjebak pada solusi suboptimal. Mutasi biasanya dilakukan dengan probabilitas rendah (Holland, 1992).

GA secara berulang kali menerapkan proses seleksi, *crossover*, dan mutasi untuk menciptakan generasi baru individu yang lebih baik. GA memanfaatkan prinsip evolusi dan seleksi alam untuk mencapai solusi yang optimal atau mendekati optimal dalam masalah yang kompleks.

14.2 Representasi dan Kromosom

Dalam GA, representasi dan kromosom memainkan peran penting dalam menentukan cara solusi potensial dari suatu masalah diwujudkan dan diproses. Representasi adalah cara solusi potensial dari suatu masalah diwujudkan dalam GA. Representasi ini ditentukan berdasarkan karakteristik masalah yang akan diselesaikan dan bergantung pada bagaimana kita ingin mengkodekan informasi. Sedangkan kromosom adalah struktur yang menyimpan informasi genetik dalam bentuk rangkaian gen. Dalam konteks GA, kromosom adalah suatu representasi konkret dari solusi potensial untuk suatu masalah. Gen dalam kromosom adalah elemen dasar yang menyimpan

informasi spesifik mengenai solusi tersebut, seperti nilai variabel atau parameter. Kromosom digunakan dalam GA untuk melakukan operasi genetik seperti seleksi, crossover, dan mutasi, yang bertujuan untuk mencari solusi yang lebih baik dari generasi ke generasi.

Beberapa jenis representasi yang umum digunakan dalam GA meliputi:

14.2.1 Representasi Biner

Representasi biner dalam algoritma genetika (GA) adalah salah satu metode untuk mengkodekan solusi potensial dalam bentuk string biner yang terdiri dari 0 dan 1. Representasi ini cukup populer dan banyak digunakan dalam GA karena sifatnya yang sederhana dan mudah dioperasikan. Dalam representasi biner, kromosom berbentuk string yang terdiri dari sejumlah gen biner (0 atau 1). Panjang string ini bergantung pada kompleksitas masalah dan jumlah variabel yang diperlukan. Setiap gen biner menyimpan informasi spesifik mengenai solusi tersebut.

Representasi biner memiliki beberapa kelebihan, seperti kemudahan dalam melakukan operasi genetik (seleksi, *crossover*, dan mutasi), serta kemampuannya untuk merepresentasikan berbagai tipe data, seperti bilangan bulat, real, atau kategori, dengan mengkonversi nilai-nilai tersebut ke dalam bentuk biner.

14.2.2 Representasi Real

Berbeda dengan representasi biner, representasi real akan mengkodekan solusi dari suatu masalah dengan menggunakan bilangan real. Dalam metode ini, setiap individu dalam populasi direpresentasikan sebagai vektor bilangan real. Representasi ini biasanya digunakan untuk masalah yang melibatkan variabel kontinu, seperti optimasi fungsi dan masalah-masalah pada bidang teknik dan ilmu pengetahuan.

Contohnya, misalkan ingin mencari nilai minimum dari suatu fungsi matematika yang melibatkan dua variabel kontinu, x dan y . Pertama, kita akan menciptakan populasi awal yang terdiri dari beberapa individu dengan koordinat (x, y) sebagai bilangan real. Setiap individu akan dihitung nilai *fitness* berdasarkan fungsi yang ingin kita minimalkan. Kemudian, akan dilakukan proses seleksi untuk memilih individu yang lebih unggul berdasarkan nilai *fitness*-nya. Setelah itu, akan diterapkan operator crossover dan mutasi yang sesuai untuk representasi real. Salah satu metode crossover yang populer dalam representasi real adalah BLX-alpha (*Bounded Linear Crossover*), yang menciptakan keturunan dengan kombinasi linear dari kedua orang tua. Mutasi pada representasi real dapat dilakukan dengan menambahkan atau mengurangi nilai pada setiap gen dengan nilai acak yang dihasilkan dari distribusi tertentu, seperti distribusi normal atau distribusi seragam. Proses ini akan diulangi beberapa kali hingga mendapatkan individu dengan *fitness* yang lebih baik atau mencapai kriteria berhenti yang telah ditentukan. Dalam hal ini, representasi real memungkinkan kita untuk menerapkan konsep Algoritma Genetika dalam mencari solusi optimal pada masalah yang melibatkan variabel kontinu. Dengan demikian, kita dapat menggabungkan kekuatan komputasi dan prinsip evolusi alam untuk menghasilkan solusi yang efisien dan efektif.

14.2.3 Representasi Permutasi

Representasi Permutasi dalam GA adalah metode pengkodean solusi yang menggunakan urutan atau permutasi elemen tertentu. Metode ini biasa digunakan pada masalah yang solusinya terdiri dari rangkaian elemen dalam urutan tertentu, seperti masalah penjadwalan, pengiriman barang, atau penjual keliling (*travelling salesman problem*).

Dalam representasi permutasi, setiap individu dalam populasi merupakan urutan elemen tertentu yang mencerminkan solusi masalah yang dihadapi. Sebagai contoh, kita ambil masalah penjual keliling dengan 5 kota yang harus dikunjungi. Setiap individu dalam populasi akan terdiri dari permutasi dari 5 kota tersebut, seperti (1, 2, 3, 4, 5) atau (3, 5, 1, 4, 2). Untuk menghitung nilai *fitness* setiap individu, dapat menggunakan fungsi objektif yang relevan dengan masalah yang sedang dipecahkan. Dalam kasus penjual keliling, fungsi objektif dapat berupa total jarak yang ditempuh oleh penjual saat mengunjungi semua kota dalam urutan yang diberikan oleh individu.

Proses seleksi, crossover, dan mutasi dalam representasi permutasi perlu disesuaikan agar tetap menghasilkan individu yang valid. Misalnya, operator crossover yang umum digunakan dalam representasi permutasi adalah *Order Crossover* (OX) atau *Partially Mapped Crossover* (PMX), yang mempertahankan urutan relatif elemen dalam permutasi. Mutasi pada representasi permutasi dapat dilakukan dengan cara menukar posisi dua elemen secara acak dalam permutasi.

Dengan menerapkan representasi permutasi dalam Algoritma Genetika, kita bisa mencari solusi optimal untuk masalah yang melibatkan urutan elemen. Proses evolusi akan membantu kita menemukan kombinasi terbaik dari elemen tersebut untuk mencapai tujuan optimasi yang ditetapkan.

14.3 Operator Algoritma Genetika

GA menggunakan operator-operator untuk menjalankan proses evolusi dan menciptakan generasi baru dari individu. Dengan berjalannya waktu dan evolusi, populasi akan cenderung ke arah solusi yang lebih optimal untuk masalah yang dihadapi.

Terdapat beberapa operator utama yang digunakan untuk menjalankan proses evolusi dan menciptakan generasi baru dari individu yang ada. Berikut adalah operator-operator utama dalam Algoritma Genetika:

14.3.1 Seleksi

Operator seleksi bertujuan untuk memilih individu yang lebih baik dari populasi berdasarkan nilai *fitness* mereka. Proses seleksi ini akan memberikan individu dengan nilai *fitness* yang lebih baik peluang lebih besar untuk dipilih dan berkontribusi dalam menciptakan generasi baru.

Operator ini sangat penting dalam menentukan individu mana yang akan dipilih untuk berkontribusi dalam proses reproduksi, seperti crossover dan mutasi, sehingga dapat dikatakan bahwa operator ini akan mempengaruhi kinerja Algoritma Genetika, karena menentukan bagaimana individu yang berbeda berkontribusi pada proses evolusi. Oleh karena itu, penting untuk memilih metode seleksi yang sesuai dengan masalah yang dihadapi dan karakteristik populasi yang digunakan.

Berikut ini beberapa metode seleksi yang umum digunakan :

1. *Roulette Wheel Selection* (Seleksi Roda Rolet): Metode ini menggunakan konsep probabilitas yang sebanding dengan nilai *fitness* individu. Dalam *Roulette Wheel Selection*, setiap individu akan memiliki peluang untuk dipilih sebanding dengan nilai *fitness* mereka dibandingkan dengan total kebugaran seluruh populasi. Individu dengan *fitness* yang lebih tinggi akan memiliki peluang lebih besar untuk dipilih, tetapi individu dengan *fitness* yang lebih rendah masih memiliki kesempatan untuk terpilih.
2. *Rank Selection* (Seleksi Peringkat): Metode ini mengurangi tekanan selektif pada individu dengan *fitness* tinggi. Individu akan diurutkan berdasarkan nilai *fitness* mereka, dan kemudian probabilitas seleksi akan ditentukan

berdasarkan peringkat individu tersebut. Dengan cara ini, individu dengan *fitness* tertinggi akan tetap memiliki peluang lebih besar untuk dipilih, tetapi perbedaan antara peluang individu akan lebih merata.

3. *Tournament Selection* (Seleksi Turnamen): Metode ini melibatkan pemilihan acak sejumlah individu dari populasi (peserta turnamen) dan kemudian memilih individu dengan *fitness* tertinggi di antara mereka. Proses ini diulangi beberapa kali hingga jumlah individu yang diinginkan terpilih. Ukuran turnamen dapat disesuaikan untuk mengontrol tekanan selektif; turnamen yang lebih besar akan menghasilkan tekanan selektif yang lebih tinggi, sedangkan turnamen yang lebih kecil akan menghasilkan tekanan selektif yang lebih rendah.
4. *Stochastic Universal Sampling* (SUS): Metode ini merupakan variasi dari *Roulette Wheel Selection* yang mengurangi bias terhadap individu dengan *fitness* sangat tinggi. SUS memilih individu dengan cara yang sama dengan *Roulette Wheel Selection*, tetapi menggunakan satu set poin sepanjang roda rolet yang memiliki jarak yang sama satu sama lain, sehingga memastikan individu yang berbeda akan terpilih.

14.3.2 Crossover (Rekombinasi)

Operator *Crossover* atau Rekombinasi digunakan untuk menggabungkan informasi dari dua individu orang tua untuk menciptakan satu atau lebih individu keturunan. Proses ini meniru konsep reproduksi seksual dalam alam, di mana informasi genetik dari kedua orang tua dikombinasikan untuk menghasilkan keturunan. Beberapa metode crossover yang umum digunakan adalah :

1. *Single-Point Crossover*, metode ini memilih satu titik pemotongan secara acak pada kedua individu orang tua. Kemudian, bagian di belakang titik pemotongan dari kedua

orang tua ditukar untuk membentuk dua individu keturunan baru. Metode ini biasa digunakan pada representasi biner.

2. *Multi-Point Crossover*, mirip dengan *Single-Point Crossover*, tetapi dalam kasus ini, lebih dari satu titik pemotongan dipilih secara acak. Bagian antara titik-titik pemotongan kemudian ditukar secara bergantian antara kedua individu orang tua untuk menghasilkan keturunan baru. Metode ini juga biasa digunakan pada representasi biner.
3. *Uniform Crossover*, metode ini menghasilkan keturunan dengan menggabungkan gen orang tua secara acak dan independen. Untuk setiap gen dalam individu keturunan, gen tersebut diambil dari salah satu orang tua dengan probabilitas yang sama. Metode ini sering digunakan pada representasi biner, tetapi juga dapat diterapkan pada representasi real atau permutasi.
4. *Order Crossover (OX)*, metode ini dirancang khusus untuk representasi permutasi. Pada OX, rentang elemen yang bersebelahan dipilih secara acak dari salah satu orang tua dan disalin ke keturunan. Kemudian, elemen yang tersisa dari orang tua yang lain disalin ke keturunan dalam urutan yang ada tanpa mengubah posisi relatif mereka.
5. *Partially Mapped Crossover (PMX)*, merupakan metode crossover yang juga dirancang untuk representasi permutasi. Dalam PMX, rentang elemen yang bersebelahan dipilih secara acak dari kedua orang tua dan kemudian ditukar antara mereka. Setelah itu, elemen yang konflik dipecahkan dengan memetakan elemen yang ditukar ke posisi asalnya pada individu yang lain.

Pemilihan metode tergantung pada representasi yang digunakan dalam Algoritma Genetika (biner, real, atau permutasi) dan karakteristik masalah yang dihadapi. Crossover

memungkinkan pertukaran informasi genetik antara individu dalam populasi, yang pada akhirnya dapat menyebabkan penemuan solusi yang lebih baik dan lebih optimal.

14.3.3 Mutasi

Operator mutasi bertujuan untuk menghasilkan variasi dalam populasi dengan mengubah informasi genetik pada individu secara acak. Proses ini meniru konsep mutasi dalam alam, yang merupakan perubahan acak dalam DNA. Mutasi membantu menjaga keragaman dalam populasi dan mencegah konvergensi prematur ke solusi yang kurang optimal. Beberapa metode mutasi yang umum digunakan adalah *Bit Flip Mutation* (untuk representasi biner), *Gaussian Mutation* (untuk representasi real), dan *Swap Mutation*, *Insertion Mutation*, dan *Scramble Mutation* (untuk representasi permutasi).

Sama halnya dengan operator lainnya, operator mutasi yang dipilih tergantung pada representasi yang digunakan dalam Algoritma Genetika (biner, real, atau permutasi) dan karakteristik masalah yang dihadapi. Mutasi memungkinkan penjelajahan ruang pencarian yang lebih luas dan membantu populasi untuk mencapai solusi yang lebih optimal sepanjang proses evolusi.

14.4 Strategi dan Parameter Algoritma Genetika

Strategi dan parameter dalam GA sangat penting untuk menentukan kinerja dan hasil akhir algoritma. Strategi mencakup pemilihan metode untuk operator genetika seperti seleksi, crossover, dan mutasi, serta cara mereka digunakan dalam proses evolusi. Parameter mencakup nilai-nilai yang menentukan tingkat variasi dan kecepatan evolusi, seperti probabilitas crossover dan mutasi.

14.4.1 Populasi dan Inisialisasi

Populasi adalah kumpulan individu yang mewakili solusi potensial dari suatu masalah. Setiap individu dalam populasi memiliki informasi genetik yang dinyatakan dalam bentuk representasi tertentu, seperti biner, real, atau permutasi. Populasi menjadi basis untuk proses evolusi, di mana individu akan dipilih, dikombinasikan, dan dimodifikasi melalui operator genetika seperti seleksi, crossover, dan mutasi. Ukuran populasi sangat penting, karena mempengaruhi seberapa luas ruang pencarian yang dijelajahi oleh algoritma dan kecepatan konvergensi ke solusi yang optimal.

Sedangkan inisialisasi merupakan tahap awal dalam GA di mana populasi pertama kali dibentuk. Proses inisialisasi melibatkan pembentukan individu dengan cara yang acak atau menggunakan informasi apriori tentang masalah yang dihadapi. Tujuan inisialisasi adalah untuk menciptakan populasi awal yang memiliki keragaman genetik yang cukup untuk memungkinkan algoritma menemukan solusi yang baik sepanjang proses evolusi. Inisialisasi yang baik sangat penting, karena populasi awal yang buruk atau tidak bervariasi dapat mengakibatkan algoritma tidak dapat mencapai solusi yang optimal.

Sebagai contoh, dalam masalah optimasi, individu dalam populasi akan mewakili titik-titik di ruang pencarian yang diharapkan mengandung solusi optimal. Selama proses inisialisasi, kita ingin memastikan bahwa populasi awal mencakup sejumlah titik yang tersebar secara merata di seluruh ruang pencarian, sehingga algoritma memiliki kesempatan yang lebih baik untuk menemukan solusi yang optimal. Metode yang umum digunakan untuk menginisialisasi populasi meliputi penggunaan distribusi seragam acak untuk menghasilkan individu atau penggunaan teknik heuristik untuk

memandu pembentukan individu berdasarkan pengetahuan tentang masalah yang dihadapi.

14.4.2 Kriteria Terminasi

Kriteria terminasi dalam GA adalah kondisi yang menandakan kapan algoritma harus berhenti melakukan evolusi. Kriteria terminasi penting untuk menentukan seberapa lama algoritma beroperasi dan berapa banyak komputasi yang diperlukan.

Beberapa kriteria terminasi yang umum digunakan meliputi:

1. Jumlah generasi: Algoritma akan berhenti setelah mencapai jumlah generasi tertentu yang telah ditentukan sebelumnya.
2. Konvergensi: Algoritma akan berhenti ketika populasi mencapai tingkat konvergensi tertentu, misalnya ketika sebagian besar individu dalam populasi memiliki *fitness* yang hampir sama.
3. Capaian solusi yang memuaskan: Algoritma akan berhenti ketika solusi dengan tingkat kebugaran yang memuaskan telah ditemukan.
4. Waktu komputasi: Algoritma akan berhenti setelah waktu komputasi yang ditentukan telah berlalu.

Pemilihan kriteria terminasi yang tepat sangat penting, karena kriteria yang terlalu ketat dapat mengakibatkan algoritma berhenti sebelum mencapai solusi yang optimal, sedangkan kriteria yang terlalu longgar dapat menyebabkan algoritma berjalan terlalu lama tanpa peningkatan yang signifikan pada solusi.

14.5 Evaluasi dan Fitness Function

Evaluasi dan *Fitness Function* merupakan komponen yang digunakan untuk mengukur kualitas solusi yang diwakili oleh individu dalam populasi. Evaluasi adalah proses

menghitung nilai *fitness* setiap individu dalam populasi. Evaluasi dilakukan dengan menerapkan *Fitness Function* pada individu tersebut.

14.5.1 Konsep Fitness Function

Konsep dasar *Fitness Function* berkaitan dengan bagaimana mengukur dan mengevaluasi kualitas solusi yang diwakili oleh individu dalam populasi. *Fitness Function* merupakan fungsi yang dirancang khusus untuk menghitung seberapa baik solusi yang diwakili oleh individu dalam konteks masalah yang dihadapi. Nilai *fitness* yang dihasilkan akan digunakan dalam proses seleksi, di mana individu dengan *fitness* yang lebih tinggi memiliki peluang yang lebih besar untuk dipilih sebagai orang tua untuk generasi berikutnya.

Fitness Function harus dirancang sesuai dengan karakteristik masalah yang dihadapi. Misalnya, dalam masalah optimasi, *Fitness Function* biasanya berhubungan langsung dengan fungsi objektif yang ingin dioptimalkan. Dalam kasus ini, individu dengan nilai *fitness* yang lebih tinggi akan dianggap sebagai solusi yang lebih baik, dan tujuan dari Algoritma Genetika adalah untuk menemukan individu dengan nilai *fitness* tertinggi. Dalam masalah klasifikasi, *Fitness Function* dapat mengukur akurasi atau presisi dari klasifikasi yang dihasilkan oleh individu.

Pemilihan *Fitness Function* yang tepat sangat penting, karena fungsi ini menentukan bagaimana algoritma mengeksplorasi ruang pencarian dan mencapai solusi yang optimal. *Fitness Function* yang baik harus mampu menggambarkan seberapa baik solusi yang diwakili oleh individu dalam konteks masalah yang dihadapi dan mengarahkan algoritma menuju solusi yang lebih baik sepanjang proses evolusi. Oleh karena itu, perancangan *Fitness Function* yang efektif dan efisien menjadi salah satu faktor

kunci dalam implementasi sukses Algoritma Genetika untuk menyelesaikan berbagai masalah.

14.5.2 Desain Fitness Function yang Efektif

Desain Fitness Function yang efektif sangat penting karena memiliki peran dalam mengarahkan algoritma untuk mencapai solusi optimal.

Berikut beberapa langkah untuk mendesain Fitness Function yang efektif:

1. **Pemahaman masalah:** Pertama, pemahaman yang mendalam tentang masalah yang ingin diselesaikan sangat penting. Ini termasuk mengidentifikasi variabel, batasan, dan sasaran yang ingin dicapai. Dengan memahami masalah secara menyeluruh, kita dapat merancang Fitness Function yang mencerminkan tujuan dan karakteristik masalah dengan baik.
2. **Tepat sasaran:** Fitness Function harus dirancang sedemikian rupa sehingga mencerminkan tujuan yang ingin dicapai. Dalam masalah optimasi, fungsi objektif yang ingin dioptimalkan sebaiknya menjadi bagian dari Fitness Function. Dalam masalah klasifikasi, Fitness Function dapat mengukur akurasi atau presisi dari klasifikasi yang dihasilkan oleh individu.
3. **Sensitif terhadap perubahan:** Fitness Function yang baik harus mampu menggambarkan perbedaan kecil dalam solusi. Dengan demikian, algoritma dapat membedakan individu dengan solusi yang hampir sama dan mengarahkan evolusi menuju solusi yang lebih baik.
4. **Skalabilitas:** Pastikan Fitness Function yang dirancang memiliki kinerja yang baik dalam berbagai ukuran masalah. Hal ini akan memungkinkan algoritma untuk menyelesaikan masalah yang lebih kompleks dengan jumlah variabel yang lebih besar.

5. Batasan: Jika ada batasan dalam masalah yang dihadapi, pastikan Fitness Function mampu menangani batasan tersebut. Salah satu cara untuk melakukannya adalah dengan memasukkan hukuman atau penalti ke dalam Fitness Function ketika batasan dilanggar oleh solusi.
6. Efisiensi komputasi: Fitness Function yang dirancang sebaiknya tidak memerlukan waktu komputasi yang lama, karena akan mempengaruhi efisiensi keseluruhan Algoritma Genetika. Cobalah untuk menyederhanakan Fitness Function sebisa mungkin tanpa mengorbankan kemampuannya untuk menggambarkan kualitas solusi.

Dengan mengikuti langkah-langkah tersebut, kita dapat merancang Fitness Function yang efektif, yang akan membantu Algoritma Genetika mencapai solusi yang optimal dalam waktu yang lebih singkat dan dengan efisiensi yang lebih baik.

14.6 Aplikasi Algoritma Genetika

Dalam beberapa dekade terakhir, Algoritma Genetika telah menjadi salah satu teknik yang populer dan efektif untuk menyelesaikan berbagai masalah yang melibatkan optimasi dan pencarian solusi dalam ruang yang besar dan kompleks. Aplikasi Algoritma Genetika meluas ke berbagai bidang, seperti optimasi, pembelajaran mesin, penjadwalan, ruting, desain dan evolusi, bioinformatika, dan optimasi portofolio, menunjukkan fleksibilitas dan kemampuannya dalam mengatasi berbagai masalah yang dihadapi.

14.6.1 Optimasi Fungsi Matematika

Aplikasi Algoritma Genetika dalam Optimasi Fungsi Matematika melibatkan pencarian solusi optimal untuk masalah yang didefinisikan oleh suatu fungsi objektif matematika. Tujuan dari optimasi ini adalah untuk mencari nilai minimum atau maksimum dari fungsi objektif tersebut,

dengan mempertimbangkan variabel dan batasan yang ada. Algoritma Genetika menawarkan pendekatan heuristik yang mampu mengeksplorasi ruang solusi secara efisien dan efektif dalam mencari solusi optimal.

Sebagai contoh, misalkan kita memiliki fungsi matematika $f(x, y)$ yang ingin dioptimalkan dengan mempertimbangkan batasan-batasan tertentu pada x dan y . Algoritma Genetika akan diinisialisasi dengan populasi solusi acak, di mana setiap individu dalam populasi merepresentasikan pasangan nilai x dan y . Selanjutnya, Fitness Function akan dirancang berdasarkan fungsi objektif $f(x, y)$ dan batasan yang ada.

Algoritma akan mengulangi proses evolusi melalui operator seleksi, crossover, dan mutasi, dengan tujuan meningkatkan kebugaran individu dalam populasi. Proses evolusi ini akan berlangsung hingga kriteria terminasi terpenuhi, seperti mencapai jumlah generasi maksimum atau mencapai tingkat konvergensi tertentu. Pada akhir proses, algoritma akan menghasilkan individu dengan kebugaran terbaik, yang merepresentasikan solusi optimal untuk masalah optimasi fungsi matematika tersebut.

Algoritma Genetika telah berhasil diterapkan dalam berbagai masalah optimasi fungsi matematika, termasuk fungsi objektif non-linear, multimodal, dan diskontinu. Kelebihan Algoritma Genetika dalam hal ini adalah kemampuannya untuk mengeksplorasi ruang solusi secara global dan menghindari jebakan solusi lokal, yang sering menjadi tantangan dalam metode optimasi tradisional.

14.6.2 Optimasi Rute dan Penjadwalan

Aplikasi Algoritma Genetika dalam Optimasi Rute dan Penjadwalan melibatkan penggunaan metode ini untuk menemukan solusi terbaik dalam menentukan urutan tempat

yang harus dikunjungi atau urutan tugas yang harus diselesaikan, dengan meminimalkan biaya, waktu tempuh, atau sumber daya yang digunakan. Beberapa contoh dari masalah rute dan penjadwalan yang dapat diselesaikan dengan Algoritma Genetika meliputi *Traveling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP), dan penjadwalan produksi atau sumber daya manusia.

1. *Traveling Salesman Problem* (TSP): Dalam masalah ini, seorang salesman harus mengunjungi sejumlah kota dengan jumlah minimum jarak tempuh. Setiap kota harus dikunjungi tepat satu kali, dan salesman harus kembali ke kota asal setelah mengunjungi semua kota lainnya. Algoritma Genetika digunakan untuk mencari urutan optimal kota yang harus dikunjungi. Dalam kasus ini, representasi individu dalam populasi bisa berupa urutan kota, dan Fitness Function didasarkan pada jarak tempuh total dari urutan tersebut.
2. *Vehicle Routing Problem* (VRP): VRP merupakan perluasan dari TSP, di mana beberapa kendaraan harus mengirimkan barang ke sejumlah pelanggan dengan meminimalkan jarak tempuh total atau waktu operasional. Algoritma Genetika digunakan untuk mencari rute optimal bagi setiap kendaraan dan pembagian pelanggan antar kendaraan. Representasi individu dalam populasi bisa berupa urutan pelanggan dan pembagian antara kendaraan, sementara Fitness Function didasarkan pada biaya operasional total atau jarak tempuh.
3. *Penjadwalan Produksi atau Sumber Daya Manusia*: Dalam masalah penjadwalan, kita harus menentukan urutan tugas yang harus diselesaikan atau jadwal kerja karyawan dengan meminimalkan biaya produksi, waktu penyelesaian, atau jumlah karyawan yang diperlukan. Algoritma Genetika digunakan untuk mencari urutan tugas atau jadwal yang

optimal. Representasi individu dalam populasi bisa berupa urutan tugas atau jadwal kerja, sementara *Fitness Function* didasarkan pada biaya atau waktu penyelesaian total.

14.7 Algoritma Genetika Paralel dan Hibrid

Algoritma Genetika Paralel memanfaatkan arsitektur komputer paralel untuk menjalankan beberapa proses evolusi secara bersamaan, sehingga mempercepat pencarian solusi optimal dan mengurangi waktu komputasi yang diperlukan. Sementara itu, Algoritma Genetika Hibrid menggabungkan Algoritma Genetika dengan metode optimasi atau pencarian heuristik lainnya, seperti *Hill Climbing*, *Tabu Search*, atau *Particle Swarm Optimization*, dengan tujuan memanfaatkan kelebihan dari masing-masing metode dan menghasilkan solusi yang lebih baik dan lebih cepat.

14.7.1 Algoritma Genetika Paralel

Algoritma Genetika Paralel merupakan pendekatan yang menggabungkan prinsip-prinsip dasar Algoritma Genetika dengan teknologi komputasi paralel untuk meningkatkan kinerja dan efisiensi pencarian solusi optimal. Dalam Algoritma Genetika Paralel, beberapa proses evolusi dijalankan secara bersamaan pada beberapa unit pemrosesan atau komputer yang terhubung, memungkinkan pencarian solusi yang lebih cepat dan efisien daripada algoritma genetika standar (Cantú-Paz, 2001).

Berikut adalah beberapa model Algoritma Genetika Paralel yang umum digunakan (Tomassini, 1999):

1. Model Global. Dalam model ini, seluruh populasi individu dipecah menjadi beberapa sub-populasi yang dijalankan secara paralel pada unit pemrosesan yang berbeda. Setiap sub-populasi menjalani proses evolusi secara independen, namun secara berkala individu terbaik dari setiap sub-populasi akan ditukar dengan individu terbaik dari sub-

populasi lain melalui mekanisme migrasi. Hal ini memungkinkan pertukaran informasi genetik antar sub-populasi dan mempercepat konvergensi ke solusi optimal.

2. Model Fine-Grained. Model ini juga dikenal sebagai *Synchronous Parallel Genetic Algorithm* (SPGA) atau *Cellular Genetic Algorithm* (CGA). Dalam model ini, individu dalam populasi ditempatkan pada jaringan dua dimensi, di mana setiap individu hanya berinteraksi dengan individu tetangganya dalam proses evolusi. Proses seleksi, crossover, dan mutasi dijalankan secara paralel pada setiap individu, memungkinkan komputasi yang lebih cepat dan efisien. Model ini juga dapat meningkatkan variasi genetik dan mengurangi risiko terjebak di solusi lokal.
3. Model Hybrid: Model ini merupakan kombinasi dari model Global dan Fine-Grained, di mana beberapa sub-populasi dijalankan secara paralel, dan setiap sub-populasi diorganisir dalam jaringan dua dimensi. Model ini mencoba memanfaatkan kelebihan dari kedua model sebelumnya, dengan mempercepat konvergensi dan menjaga variasi genetik dalam populasi.

Algoritma Genetika Paralel menawarkan beberapa keuntungan, seperti peningkatan efisiensi komputasi, penjelajahan lebih luas dari ruang solusi, dan kemampuan untuk menemukan solusi optimal lebih cepat. Namun, implementasi Algoritma Genetika Paralel juga memerlukan perhatian khusus pada komunikasi antar unit pemrosesan dan sinkronisasi proses evolusi, serta penentuan parameter seperti ukuran populasi dan tingkat migrasi yang sesuai.

14.7.2 Teknik Integrasi dengan Algoritma Lain

Teknik integrasi Algoritma Genetika dengan algoritma lain melibatkan penggabungan Algoritma Genetika dengan metode optimasi atau pencarian heuristik lainnya untuk meningkatkan kinerja dan efisiensi dalam menemukan solusi optimal. Pendekatan ini dikenal sebagai Algoritma Genetika Hibrid. Algoritma Genetika Hibrid mencoba memanfaatkan kelebihan dari masing-masing metode, sehingga menghasilkan solusi yang lebih baik dan lebih cepat.

Beberapa contoh dari teknik integrasi Algoritma Genetika dengan algoritma lain meliputi:

1. Memadukan Algoritma Genetika dengan *Hill Climbing*: Algoritma ini dikenal sebagai *Memetic Algorithm* atau *Lamarckian Genetic Algorithm* (Krasnogor and Smith, 2005). Dalam pendekatan ini, setelah individu terpilih untuk reproduksi, *Hill Climbing* diaplikasikan pada individu tersebut untuk mencari solusi lokal yang lebih baik sebelum melakukan crossover dan mutasi. Hal ini dapat meningkatkan kecepatan konvergensi dan kemampuan untuk menemukan solusi optimal.
2. Menggabungkan Algoritma Genetika dengan Tabu Search: Dalam kombinasi ini, Tabu Search diintegrasikan ke dalam Algoritma Genetika untuk meningkatkan pencarian solusi di ruang solusi yang besar (Glover and Laguna, 1998). Tabu Search digunakan untuk menggali lebih dalam ke ruang solusi dan menghindari solusi yang telah dijelajahi sebelumnya.
3. Mengintegrasikan Algoritma Genetika dengan *Particle Swarm Optimization* (PSO): Dalam metode ini, PSO diaplikasikan pada populasi Algoritma Genetika dengan menggabungkan kecepatan dan posisi partikel dalam ruang pencarian. Hal ini memungkinkan Algoritma Genetika untuk memanfaatkan pencarian global yang efisien dari PSO,

sementara PSO memanfaatkan operator crossover dan mutasi Algoritma Genetika untuk meningkatkan variasi solusi (Moscato, 1989).

14.8 Evaluasi Kinerja dan Metrik

Evaluasi kinerja dan metrik merupakan aspek penting dalam algoritma genetika, karena memberikan gambaran mengenai seberapa baik solusi yang dihasilkan oleh algoritma tersebut dalam menyelesaikan masalah yang diberikan (Haupt and Haupt, 2003). Evaluasi kinerja melibatkan pengukuran beberapa metrik yang mencerminkan kualitas solusi dan efisiensi algoritma dalam mencapai solusi tersebut.

Berikut beberapa metrik umum yang digunakan untuk mengukur kinerja algoritma genetika:

1. Nilai *fitness* menggambarkan seberapa baik suatu individu dalam populasi mampu menyelesaikan masalah yang diberikan. Biasanya, nilai *fitness* dihitung berdasarkan fungsi objektif yang ingin dioptimalkan. Metrik ini penting untuk mengukur kualitas solusi yang dihasilkan oleh algoritma.
2. Konvergensi menggambarkan seberapa cepat algoritma genetika menemukan solusi optimal atau mendekati solusi optimal. Algoritma yang cepat konvergenya dianggap lebih efisien dan efektif dalam menemukan solusi terbaik.
3. Diversitas menggambarkan variasi solusi yang ada dalam populasi algoritma genetika. Diversitas yang tinggi dapat membantu algoritma menjelajahi ruang pencarian secara lebih luas dan mengurangi risiko terjebak di solusi lokal.
4. Efisiensi komputasi menggambarkan seberapa cepat algoritma genetika menemukan solusi optimal dengan sumber daya komputasi yang tersedia, seperti waktu CPU dan memori yang digunakan. Algoritma yang efisien dalam hal komputasi lebih diinginkan dalam aplikasi praktis.

14.9 Studi Kasus dan Implementasi

14.9.1 Studi Kasus: Optimasi Fungsi Ackley

Fungsi Ackley adalah salah satu fungsi benchmark yang digunakan dalam bidang optimasi komputasi untuk menguji kinerja algoritma optimasi. Fungsi ini memiliki banyak minimum lokal dan satu minimum global, sehingga menjadikannya tantangan yang baik untuk dipecahkan oleh algoritma optimasi, termasuk Algoritma Genetika. Fungsi Ackley didefinisikan sebagai berikut:

$$f(x) = -20 * \exp(-0.2 * \sqrt{1/n * \sum (x_i^2)}) - \exp(1/n * \sum (\cos(2\pi * x_i))) + 20 + e$$

di mana $x = (x_1, x_2, \dots, x_n)$ adalah vektor solusi dengan n dimensi, dan $\sum$ adalah simbol sigma yang melambangkan penjumlahan.

Untuk mengoptimalkan fungsi Ackley menggunakan Algoritma Genetika, kita perlu mengikuti langkah-langkah berikut:

1. Inisialisasi: Tentukan ukuran populasi, panjang kromosom, dan rentang nilai untuk setiap variabel x_i . Inisialisasi populasi awal dengan individu yang dihasilkan secara acak dalam rentang nilai yang ditentukan.
2. Evaluasi: Hitung nilai kebugaran untuk setiap individu dalam populasi dengan menggantikan vektor solusi x ke dalam fungsi Ackley. Ingatlah bahwa tujuan kita adalah meminimalkan fungsi Ackley, jadi individu dengan nilai *fitness* yang lebih rendah akan dianggap lebih baik.
3. Seleksi: Pilih individu-individu yang akan digunakan untuk proses reproduksi berdasarkan nilai *fitness* mereka, menggunakan metode seleksi seperti *Roulette Wheel Selection* atau *Tournament Selection*.
4. Crossover: Lakukan crossover (pertukaran materi genetik) antara individu yang dipilih untuk menghasilkan individu keturunan baru. Ada beberapa teknik crossover yang bisa

digunakan, seperti *Single-Point Crossover*, *Multi-Point Crossover*, atau *Uniform Crossover*.

5. Mutasi: Lakukan mutasi (perubahan acak pada materi genetik) pada individu keturunan dengan probabilitas mutasi yang telah ditentukan. Mutasi membantu menjaga variasi genetik dalam populasi dan mencegah algoritma terjebak di minimum lokal.
6. Penggantian: Gantikan individu-individu lama dalam populasi dengan individu keturunan yang baru dihasilkan.
7. Terminasi: Ulangi langkah 2-6 sampai kriteria terminasi terpenuhi, seperti jumlah generasi maksimum atau perubahan nilai *fitness* minimum yang diinginkan.

Setelah algoritma selesai, individu dengan nilai kebugaran terbaik dalam populasi akan dianggap sebagai solusi optimal untuk masalah optimasi Fungsi Ackley.

14.9.2 Studi Kasus: Penjadwalan Mesin Job Shop

Penjadwalan Mesin Job Shop adalah salah satu masalah kombinatorial optimasi yang kompleks dan sering muncul dalam industri manufaktur. Tujuan dari masalah ini adalah menemukan urutan pekerjaan yang optimal pada sejumlah mesin untuk meminimalkan waktu penyelesaian total (makespan). Masalah ini terkenal karena sifat NP-hard-nya, yang berarti bahwa solusi optimal sulit ditemukan dalam waktu yang wajar untuk masalah skala besar.

Algoritma Genetika dapat digunakan untuk menemukan solusi suboptimal untuk masalah Penjadwalan Mesin Job Shop dengan langkah-langkah berikut:

1. Inisialisasi: Tentukan ukuran populasi dan panjang kromosom. Inisialisasi populasi awal dengan individu yang dihasilkan secara acak, di mana setiap individu

- merepresentasikan urutan pekerjaan pada mesin. Representasi permutasi sering digunakan dalam kasus ini.
2. Evaluasi: Hitung nilai *fitness* untuk setiap individu dalam populasi dengan menghitung waktu penyelesaian total (makespan) berdasarkan urutan pekerjaan yang direpresentasikan oleh individu tersebut. Individu dengan nilai *fitness* yang lebih rendah (makespan yang lebih pendek) dianggap lebih baik.
 3. Seleksi: Pilih individu-individu yang akan digunakan untuk proses reproduksi berdasarkan nilai kebugaran mereka, menggunakan metode seleksi seperti *Roulette Wheel Selection* atau *Tournament Selection*.
 4. Crossover: Lakukan crossover (pertukaran materi genetik) antara individu yang dipilih untuk menghasilkan individu keturunan baru. Teknik crossover khusus untuk representasi permutasi, seperti *Order Crossover* (OX) atau *Cycle Crossover* (CX), dapat digunakan dalam kasus ini.
 5. Mutasi: Lakukan mutasi (perubahan acak pada materi genetik) pada individu keturunan dengan probabilitas mutasi yang telah ditentukan. Teknik mutasi khusus untuk representasi permutasi, seperti *Swap Mutation* atau *Inversion Mutation*, dapat digunakan dalam kasus ini.
 6. Penggantian: Gantikan individu-individu lama dalam populasi dengan individu keturunan yang baru dihasilkan.
 7. Terminasi: Ulangi langkah 2-6 sampai kriteria terminasi terpenuhi, seperti jumlah generasi maksimum atau perubahan nilai *fitness* minimum yang diinginkan.

Setelah algoritma selesai, individu dengan nilai *fitness* terbaik dalam populasi akan dianggap sebagai solusi suboptimal untuk masalah Penjadwalan Mesin Job Shop. Walaupun Algoritma Genetika tidak selalu mendapatkan solusi

optimal, namun kadang mendapatkan solusi yang cukup baik dibandingkan dengan metode optimasi eksak.

DAFTAR PUSTAKA

- Cantú-Paz, E. 2001. *Efficient and Accurate Parallel Genetic Algorithms*. Boston, MA: Springer US (Genetic Algorithms and Evolutionary Computation). doi: 10.1007/978-1-4615-4369-5.
- Glover, F. and Laguna, M. 1998. 'Tabu Search', in *Handbook of Combinatorial Optimization*. Boston, MA: Springer US, pp. 2093–2229. doi: 10.1007/978-1-4613-0303-9_33.
- Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st edn. USA: Addison-Wesley Longman Publishing Co., Inc.
- Haupt, R. L. and Haupt, S. E. 2003. *Practical Genetic Algorithms*. Wiley. doi: 10.1002/0471671746.
- Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems*. Reprint Ed. Bradford Books.
- Koza, J. R. 1992. *Genetic Programming On the Programming of Computers by Means of Natural Selection*. The MIT Press.
- Krasnogor, N. and Smith, J. 2005. 'A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues', *IEEE Transactions on Evolutionary Computation*, 9(5), pp. 474–488. doi: 10.1109/TEVC.2005.850260.
- Moscato, P. 1989. *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms*. Pasadena.
- Sivanandam, S. N. and Deepa, S. N. 2008. *Introduction to Genetic Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-73190-0.
- Tomassini, M. 1999. 'Parallel and Distributed Evolutionary Algorithms: A Review', *Advances in Soft Computing: Engineering Design and Manufacturing* Springer-Verlag.

BAB 15

COGNITIF SCIENCE

Oleh Moh Safii

15.1 Sejarah perkembangan *cognitif science*

Cognitif science merupakan disiplin ilmu interdisipliner yang melibatkan banyak bidang seperti psikologi, filsafat, ilmu saraf, antropologi, dan linguistik. *Cognitif science* mempelajari cara kerja pikiran manusia, mulai dari persepsi, memori, pemikiran, dan bahasa(Stella, 2022).

Sejarah perkembangan *cognitif science* dimulai pada tahun 1940-an ketika para psikolog mulai memperhatikan kognisi sebagai objek studi. Pada saat itu, pandangan behaviorisme mendominasi psikologi, yang memandang bahwa perilaku manusia dipengaruhi oleh pengalaman yang diperoleh melalui interaksi dengan lingkungan.(Mendoza-Collazos & Zlatev, 2022)

Namun, pada awal 1950-an, pandangan behaviorisme mulai dipertanyakan oleh para psikolog dan ilmuwan lainnya, seperti Noam Chomsky, George Miller, dan Herbert Simon. Mereka berpendapat bahwa perilaku manusia tidak dapat sepenuhnya dijelaskan oleh pengalaman belajar semata, melainkan juga melibatkan proses mental yang kompleks, seperti pemikiran, perhatian, memori, dan bahasa(Hipólito, 2022).

Pada tahun 1956, John McCarthy memperkenalkan istilah "*Artificial Intelligence*" atau kecerdasan buatan. Kecerdasan buatan menjadi salah satu topik utama dalam *cognitif science* dan menjadi semakin populer pada tahun 1960-an dan 1970-an. Pada tahun 1975, Marvin Minsky dan Seymour

Papert menerbitkan buku "*Perceptrons*" yang mengkritik model jaringan saraf tiruan (*artificial neural network*) yang saat itu sedang populer.

Pada tahun 1980-an, *cognitif science* semakin berkembang dan banyak disiplin ilmu lain yang terkait dengan kognisi seperti ilmu saraf dan bahasa mulai mendapatkan perhatian yang lebih besar. Pada tahun 1990-an, *cognitif science* semakin populer dengan adanya perkembangan teknologi yang semakin canggih, seperti fMRI (*functional magnetic resonance imaging*) yang memungkinkan pengamatan langsung terhadap otak saat beraktivitas.

Sejak awal abad ke-21, *cognitif science* semakin berkembang dan semakin banyak terdapat aplikasi teknologi yang berbasis kognisi seperti virtual reality, augmented reality, dan *machine learning*. *Cognitif science* juga menjadi subjek studi penting dalam pengembangan robotika dan kecerdasan buatan (Goldstone, 2019).

15.2 Tujuan dari *cognitif science*

Tujuan dari *cognitif science* adalah untuk memahami bagaimana pikiran manusia bekerja. Disiplin ilmu ini mencakup berbagai aspek kognisi, seperti persepsi, memori, bahasa, pemikiran, dan kecerdasan, dan bertujuan untuk menjelaskan proses mental yang mendasari perilaku manusia (Engbert et al., 2022).

Beberapa tujuan utama dari *cognitif science* adalah:

a) Menjelaskan proses mental

Cognitif science bertujuan untuk menjelaskan bagaimana otak manusia memproses informasi dan menghasilkan pemikiran, persepsi, dan perilaku.

- b) Mengembangkan teori
Cognitif science berusaha untuk mengembangkan teori yang dapat menjelaskan proses mental manusia dengan cara yang sistematis dan konsisten.
- c) Mengidentifikasi faktor-faktor yang memengaruhi kognisi
Cognitif science mencoba untuk mengidentifikasi faktor-faktor yang memengaruhi proses mental manusia, seperti genetik, lingkungan, dan pengalaman.
- d) Menerapkan pengetahuan *cognitif science* dalam kehidupan sehari-hari
Cognitif science memiliki banyak aplikasi praktis, seperti dalam pengembangan teknologi yang berbasis kognisi, seperti kecerdasan buatan, robotika, virtual reality, dan augmented reality.
- e) Meningkatkan kesehatan mental
Cognitif science juga bertujuan untuk meningkatkan kesehatan mental manusia dengan mengembangkan teknik terapi dan pengobatan yang berbasis pada pengetahuan tentang proses mental manusia.

15.3 *Cognitif science* dalam bidang teknologi

Perkembangan *cognitif science* telah memberikan banyak kontribusi dalam bidang teknologi. *Cognitif science* menyediakan dasar teoritis dan metodologis untuk pengembangan teknologi yang dapat memanfaatkan pengetahuan tentang proses mental manusia (Dennett, 2009). Beberapa bidang teknologi yang telah memanfaatkan pengetahuan *cognitif science* antara lain:

- a) Kecerdasan Buatan (*Artificial Intelligence*)

Cognitif science telah menjadi landasan untuk pengembangan kecerdasan buatan (AI). AI adalah cabang dari teknologi komputer yang menggunakan algoritma untuk memproses informasi dan mengambil keputusan

seperti manusia. AI berusaha untuk mereplikasi proses mental manusia, seperti pemikiran, belajar, dan pemecahan masalah. Pengetahuan tentang proses mental manusia sangat penting dalam pengembangan algoritma AI yang efektif (Courtial & Law, 1989).

Kecerdasan Buatan (*Artificial Intelligence* atau AI) merupakan bagian dari *cognitive science* yang melibatkan studi tentang bagaimana mesin dapat meniru dan meningkatkan kemampuan intelektual manusia, seperti pemrosesan bahasa alami, penglihatan komputer, pemecahan masalah, dan pengambilan keputusan. Studi dalam bidang AI juga mencakup penelitian tentang bagaimana mesin dapat belajar dari pengalaman dan melakukan tugas-tugas yang biasanya memerlukan kecerdasan manusia. Oleh karena itu, AI merupakan cabang penting dari *cognitive science* karena berkaitan dengan kemampuan kognitif manusia dan bagaimana kita dapat mereplikasinya pada mesin (Bender, 2022).

b) Robotika

Cognitive science juga memiliki kontribusi dalam pengembangan robotika. Robotika berusaha untuk mengembangkan robot yang dapat melakukan tugas-tugas seperti manusia. Pengetahuan tentang proses mental manusia sangat penting dalam pengembangan robotika yang dapat meniru perilaku manusia, seperti pengambilan keputusan, interaksi sosial, dan adaptasi pada situasi yang berubah-ubah.

Robotika adalah bidang yang berkaitan dengan pengembangan robot dan sistem robotik. *Cognitive science*, di sisi lain, adalah disiplin ilmu yang berkaitan dengan studi tentang kognisi, pemrosesan informasi, dan perilaku manusia (Ho & Griffiths, 2022). Kedua bidang ini saling terkait, karena robotika juga berfokus pada pemrosesan

informasi dan pengambilan keputusan, yang merupakan topik yang terkait dengan cognitive science. Berikut unsur *cognitive science* dalam robotika.

Pertama, salah satu unsur cognitive science dalam robotika adalah penggunaan sistem sensorik untuk memungkinkan robot untuk berinteraksi dengan lingkungannya. Robot dapat dilengkapi dengan sensor yang mirip dengan indera manusia seperti sensor suara, sensor cahaya, dan sensor sentuhan. Dalam cognitive science, ini dikenal sebagai pengolahan input sensorik, yang juga diperlukan dalam persepsi manusia (Ho & Griffiths, 2022). Kemampuan robot untuk memproses input sensorik ini akan memungkinkan robot untuk mengambil keputusan dan bertindak secara efektif di lingkungan yang kompleks.

Kedua, robotika juga melibatkan pengembangan sistem pengambilan keputusan otomatis yang mirip dengan cara manusia mengambil keputusan. Ini juga merupakan topik yang terkait dengan cognitive science. Robot dapat dilengkapi dengan algoritma pembelajaran mesin dan pengambilan keputusan, sehingga mereka dapat mempelajari dan memperbaiki kinerja mereka seiring waktu (Calzavarini & Paternoster, 2022). Kemampuan robot untuk belajar dan mengambil keputusan ini dikenal sebagai "kecerdasan buatan" dan mirip dengan kemampuan kognitif manusia.

Ketiga, robotika juga melibatkan pengembangan sistem yang dapat berinteraksi dengan manusia. Hal ini memerlukan kemampuan robot untuk memahami bahasa manusia dan meresponsnya. Dalam cognitive science, ini dikenal sebagai pengolahan bahasa alami dan merupakan topik yang penting dalam studi tentang kognisi manusia. Pengembangan robot yang dapat berinteraksi dengan manusia memungkinkan mereka untuk digunakan dalam

berbagai aplikasi seperti asisten pribadi dan layanan pelanggan.

Keempat, robotika juga memerlukan pengembangan sistem yang dapat mengingat dan mengakses informasi. Ini dikenal sebagai memori dan merupakan topik penting dalam cognitive science. Robot dapat dilengkapi dengan sistem penyimpanan data dan algoritma pengambilan keputusan yang memungkinkan mereka untuk mengingat informasi dan membuat keputusan berdasarkan informasi yang tersedia.

Kelima, robotika juga melibatkan pengembangan sistem kontrol motorik yang mirip dengan sistem motorik manusia. Ini melibatkan pengembangan sistem yang dapat mengontrol gerakan robot dan memberikan respon yang akurat terhadap lingkungan sekitarnya. Dalam cognitive science, sistem motorik manusia merupakan topik penting dalam studi tentang gerakan manusia dan koordinasi tindakan.

c) *Virtual Reality dan Augmented Reality*

Cognitive science juga memainkan peran penting dalam pengembangan teknologi *virtual reality* (VR) dan *augmented reality* (AR). Teknologi VR dan AR menggunakan informasi sensorik untuk menciptakan lingkungan yang mirip dengan dunia nyata. Pengetahuan tentang persepsi dan memori manusia sangat penting dalam pengembangan teknologi VR dan AR yang dapat menciptakan lingkungan yang realistis dan mempengaruhi pengalaman manusia.

Virtual Reality (VR) dan *Augmented Reality* (AR) adalah teknologi yang memungkinkan pengguna untuk terlibat dalam pengalaman yang meniru lingkungan nyata atau menambahkan elemen digital ke dunia nyata. Teknologi VR dan AR sangat terkait dengan Cognitive

Science karena keduanya melibatkan interaksi manusia-komputer dan memerlukan pemahaman yang dalam tentang kognisi manusia. Berikut unsur *Cognitive Science* pada *Virtual Reality dan Augmented Reality*.

Pertama, cognitive science sangat penting dalam pengembangan VR dan AR. Pengalaman VR dan AR melibatkan penggunaan indera manusia seperti penglihatan, pendengaran, dan perasaan. Pengembangan VR dan AR yang efektif memerlukan pemahaman yang mendalam tentang kognisi manusia. Studi tentang kognisi manusia dan pemrosesan informasi merupakan bagian penting dari cognitive science, sehingga penting untuk mempertimbangkan aspek kognitif dalam pengembangan VR dan AR.

Kedua, penggunaan VR dan AR memungkinkan pengalaman immersif bagi pengguna. Ini berarti pengguna merasakan bahwa mereka benar-benar berada dalam lingkungan digital atau lingkungan yang ditingkatkan. Pengalaman ini memerlukan penggunaan teknologi yang canggih dan realistis, yang memerlukan pemahaman yang mendalam tentang persepsi dan pengolahan sensorik manusia. Dalam cognitive science, ini dikenal sebagai pengolahan input sensorik dan merupakan topik yang penting dalam studi tentang kognisi manusia.

Ketiga, VR dan AR memungkinkan pengguna untuk berinteraksi dengan lingkungan dan objek digital. Ini memerlukan pengembangan sistem yang dapat memahami bahasa manusia dan gerakan tubuh manusia. Dalam cognitive science, ini dikenal sebagai pengolahan bahasa alami dan sistem motorik. Pengembangan sistem yang dapat memahami bahasa manusia dan gerakan tubuh manusia memungkinkan pengguna untuk berinteraksi

dengan lingkungan digital dengan cara yang lebih alami dan intuitif.

Keempat, VR dan AR memungkinkan pengguna untuk mengambil keputusan dan membuat keputusan berdasarkan informasi yang tersedia. Kemampuan ini dikenal sebagai "kecerdasan buatan" dan mirip dengan kemampuan kognitif manusia. Dalam *cognitive science*, ini dikenal sebagai pengambilan keputusan dan merupakan topik penting dalam studi tentang kognisi manusia. Pengembangan sistem kecerdasan buatan memungkinkan VR dan AR untuk digunakan dalam berbagai aplikasi seperti pelatihan, pendidikan, dan hiburan.

Kelima, VR dan AR juga memerlukan pengembangan sistem yang dapat mengingat dan mengakses informasi. Ini dikenal sebagai memori dan merupakan topik penting dalam *cognitive science*. VR dan AR dapat dilengkapi dengan sistem penyimpanan data dan algoritma pengambilan keputusan yang memungkinkan mereka untuk mengingat informasi dan membuat keputusan berdasarkan informasi yang tersedia.

d) Pengenalan Wajah dan Suara

Cognitif science juga telah digunakan dalam pengembangan teknologi pengenalan wajah dan suara. Teknologi ini dapat digunakan dalam berbagai aplikasi, seperti keamanan, identifikasi, dan kontrol akses. Pengetahuan tentang proses pengenalan wajah dan suara manusia sangat penting dalam pengembangan teknologi yang akurat dan efektif.

Dengan pengetahuan tentang proses mental manusia, *cognitif science* memberikan kontribusi yang signifikan dalam pengembangan teknologi yang dapat meningkatkan kualitas hidup manusia. Melalui pengembangan teknologi yang berbasis kognisi, kita dapat menciptakan alat dan

aplikasi yang dapat membantu manusia dalam berbagai bidang, seperti kesehatan, pendidikan, dan industri.

Pengenalan wajah dan suara adalah teknologi yang semakin populer dan digunakan dalam berbagai aplikasi seperti keamanan, pengenalan pengguna, dan pengenalan bicara. Berikut unsur *cognitive science* pada teknologi pengenalan wajah dan suara.

Pertama, teknologi pengenalan wajah dan suara melibatkan pemrosesan informasi sensorik manusia. Penggunaan teknologi pengenalan wajah memerlukan pengolahan informasi visual, sementara penggunaan teknologi pengenalan suara memerlukan pengolahan informasi suara. Dalam *cognitive science*, ini dikenal sebagai pengolahan input sensorik dan merupakan topik yang penting dalam studi tentang kognisi manusia.

Kedua, teknologi pengenalan wajah dan suara memerlukan pemahaman tentang bagaimana manusia mengenali dan memproses informasi. Penggunaan teknologi pengenalan wajah dan suara memerlukan pemahaman tentang bagaimana manusia mengenali dan memproses informasi. Pengenalan wajah melibatkan pengenalan pola wajah yang unik, sedangkan pengenalan suara melibatkan pengenalan pola suara yang unik. Dalam *cognitive science*, ini dikenal sebagai pengolahan pola dan merupakan topik yang penting dalam studi tentang kognisi manusia.

Ketiga, teknologi pengenalan wajah dan suara memerlukan pengembangan algoritma yang dapat memahami informasi yang diberikan. Algoritma ini memerlukan pemahaman yang mendalam tentang bagaimana manusia memproses informasi dan membuat keputusan. Dalam *cognitive science*, ini dikenal sebagai pengambilan keputusan dan merupakan topik yang

penting dalam studi tentang kognisi manusia. Algoritma pengenalan wajah dan suara harus dapat memahami informasi yang diberikan dan membuat keputusan berdasarkan informasi tersebut.

Keempat, teknologi pengenalan wajah dan suara memerlukan pengembangan sistem yang dapat belajar dari data. Sistem ini memerlukan pemahaman yang mendalam tentang bagaimana manusia belajar dari data dan membuat keputusan berdasarkan data tersebut. Dalam cognitive science, ini dikenal sebagai pembelajaran mesin dan merupakan topik yang penting dalam studi tentang kognisi manusia. Sistem pengenalan wajah dan suara yang efektif harus dapat belajar dari data dan membuat keputusan berdasarkan data tersebut.

Kelima, teknologi pengenalan wajah dan suara juga memerlukan pengembangan sistem yang dapat mengingat dan mengakses informasi. Ini dikenal sebagai memori dan merupakan topik penting dalam cognitive science. Sistem pengenalan wajah dan suara dapat dilengkapi dengan sistem penyimpanan data dan algoritma pengambilan keputusan yang memungkinkan mereka untuk mengingat informasi dan membuat keputusan berdasarkan informasi yang tersedia.

15.4 Pentingnya mempelajari Cognitif science

Mempelajari kognitif science sangat penting dalam bidang teknologi karena pengetahuan tentang proses mental manusia sangat diperlukan dalam pengembangan teknologi yang berfokus pada pengalaman pengguna yang lebih baik. Beberapa alasan mengapa mempelajari kognitif science penting dalam bidang teknologi adalah sebagai berikut:

- a) Memahami perilaku pengguna
Dalam pengembangan teknologi, penting untuk memahami perilaku pengguna dalam menggunakan teknologi tersebut. Dengan memahami bagaimana pengguna memproses informasi, memori, dan persepsi, pengembang dapat menciptakan produk dan aplikasi yang lebih efektif dan intuitif.
- b) Meningkatkan keamanan teknologi
Pengetahuan tentang proses mental manusia dapat membantu dalam pengembangan teknologi keamanan seperti identifikasi wajah atau suara. Dengan memahami bagaimana otak manusia memproses informasi sensorik, pengembang dapat menciptakan teknologi keamanan yang lebih akurat dan dapat diandalkan.
- c) Mengembangkan teknologi yang lebih adaptif
Dalam lingkungan teknologi yang terus berubah, penting untuk dapat mengembangkan teknologi yang dapat beradaptasi dengan cepat pada perubahan yang terjadi. Dengan memahami bagaimana manusia memproses informasi dan belajar, pengembang dapat menciptakan teknologi yang dapat belajar dan beradaptasi dengan cara yang lebih efektif.
- d) Meningkatkan pengalaman pengguna
Dalam pengembangan teknologi, pengalaman pengguna adalah hal yang sangat penting. Dengan memahami bagaimana manusia memproses informasi, pengembang dapat menciptakan teknologi yang lebih mudah digunakan dan lebih intuitif. Hal ini dapat meningkatkan kepuasan pengguna dan mengurangi kesalahan yang dilakukan oleh pengguna.
- e) Meningkatkan efisiensi dan produktivitas
Pengetahuan tentang proses mental manusia dapat membantu dalam pengembangan teknologi yang dapat

meningkatkan efisiensi dan produktivitas. Dengan memahami bagaimana manusia memproses informasi dan melakukan tugas-tugas tertentu, pengembang dapat menciptakan teknologi yang dapat membantu dalam meningkatkan kinerja dan produktivitas manusia.

15.5 Kajian Kognitif science

Cognitive science adalah bidang interdisipliner yang menggabungkan konsep dan metode dari beberapa bidang, termasuk psikologi, ilmu komputer, linguistik, neurosains, filosofi, dan antropologi. Bidang ini bertujuan untuk memahami dan menjelaskan bagaimana manusia memproses informasi, berpikir, dan belajar.

Kajian dalam *cognitive science* meliputi berbagai topik yang terkait dengan fungsi kognitif manusia, seperti persepsi, perhatian, ingatan, bahasa, pengambilan keputusan, dan pembelajaran. Beberapa konsep penting dalam *cognitive science* adalah pemrosesan informasi, representasi mental, dan struktur kognitif.

Salah satu fokus utama dalam *cognitive science* adalah memahami bagaimana manusia memproses informasi dan memecahkan masalah. Dalam hal ini, bidang ini memeriksa bagaimana manusia menggunakan atensi, memilih dan mengatur informasi, dan melakukan pemrosesan semantik dan sintaksis pada informasi yang diterima.

Selain itu, *cognitive science* juga memeriksa peran bahasa dalam pemrosesan informasi. Konsep-konsep seperti tata bahasa, semantik, pragmatik, dan pemahaman membantu memahami bagaimana manusia menggunakan bahasa untuk memproses dan mengomunikasikan informasi (Ho & Griffiths, 2022).

Pembelajaran juga merupakan topik yang penting dalam *cognitive science*. Bidang ini memeriksa bagaimana manusia

belajar dari pengalaman dan bagaimana mereka menggunakan pengetahuan dan keterampilan yang diperoleh untuk memecahkan masalah baru. Dalam hal ini, *cognitive science* memeriksa konsep seperti pembelajaran berdasarkan masalah, pembelajaran melalui penguatan, dan pembelajaran dalam konteks sosial.

Dalam kajian tentang *cognitive science*, juga memeriksa bagaimana fungsi kognitif manusia diproses dalam otak. Metode neuroimaging, seperti fMRI dan EEG, digunakan untuk mempelajari aktivitas otak yang terkait dengan pemrosesan informasi, bahasa, dan pembelajaran (Saharrea, 2022).

Terakhir, *cognitive science* juga memeriksa bagaimana faktor lain, seperti budaya dan lingkungan, mempengaruhi fungsi kognitif manusia. Kajian ini menyoroti pentingnya mempertimbangkan konteks sosial dan budaya dalam memahami bagaimana manusia berpikir, belajar, dan berperilaku.

15.6 Algoritma dalam *Cognitive Science*

Algoritma adalah serangkaian instruksi yang digunakan untuk menyelesaikan tugas tertentu atau mencapai tujuan tertentu. Dalam bidang *cognitive science* di bidang komputer, ada beberapa contoh algoritma yang digunakan, antara lain:

a) Algoritma machine learning:

Algoritma ini digunakan untuk mempelajari pola dari data dan menggunakan pengetahuan tersebut untuk membuat prediksi atau membuat keputusan. Contoh algoritma *machine learning* termasuk *Decision Tree*, *Random Forest*, *Neural Network*, dan *Support Vector Machines* (Mittman, 2018). Machine learning memiliki banyak aplikasi dalam berbagai bidang, seperti dalam pengenalan wajah, pengenalan suara, klasifikasi citra, klasifikasi teks, dan banyak lagi. Dengan kemampuan untuk mempelajari pola

dari data, machine learning dapat membantu memecahkan masalah kompleks yang sulit atau bahkan tidak mungkin dilakukan dengan metode konvensional (Khalifa *et al.*, 2022; Sands, 2019). Dalam machine learning, algoritma diprogram untuk menganalisis dan mempelajari pola dari data yang diberikan, kemudian menggunakan pengetahuan yang diperoleh dari data tersebut untuk membuat prediksi atau membuat keputusan.

b) Algoritma Deep Learning

Algoritma ini adalah subkategori dari machine learning dan digunakan untuk mempelajari representasi data yang kompleks. Contoh algoritma deep learning termasuk *Convolutional Neural Networks* (CNN), *Recurrent Neural Networks* (RNN), dan *Generative Adversarial Networks* (GAN). *Deep learning* memiliki kemampuan untuk mempelajari fitur-fitur kompleks dalam data yang sulit dipahami oleh manusia, seperti dalam pengenalan wajah, pengenalan suara, dan banyak lagi. Dalam beberapa kasus, deep learning telah terbukti lebih akurat dalam memprediksi hasil dari data yang kompleks dan abstrak. Namun, deep learning memerlukan jumlah data yang besar dan kompleksitas komputasi yang tinggi.

c) Algoritma *Natural Language Processing* (NLP)

Algoritma ini digunakan untuk memahami dan memproses bahasa manusia. Contoh algoritma NLP termasuk *Sentiment Analysis*, *Named Entity Recognition* (NER), dan *Language Translation* (Solanki & Jain, 2020). Tujuan utama dari NLP adalah untuk memungkinkan komunikasi yang lebih efektif antara manusia dan komputer. NLP melibatkan teknik-teknik seperti tokenisasi, pengindeksan, parsing, klasifikasi, dan analisis sentimen untuk memproses teks dalam bahasa alami. Beberapa aplikasi dari NLP, antara lain:

- 1) Penerjemahan bahasa
NLP dapat digunakan untuk menerjemahkan teks dari satu bahasa ke bahasa lain dengan mengidentifikasi pola-pola dalam bahasa asli dan mencocokkannya dengan bahasa target.
 - 2) Analisis sentiment
NLP dapat digunakan untuk menganalisis sentimen dalam teks, baik itu positif, negatif, atau netral. Hal ini dapat membantu perusahaan dalam memantau umpan balik pelanggan atau mengidentifikasi tren dalam opini publik.
 - 3) Pemrosesan bahasa alami untuk chatbot
NLP dapat digunakan untuk membangun chatbot yang dapat memproses pertanyaan dalam bahasa alami dan memberikan jawaban yang tepat.
 - 4) Ekstraksi informasi
NLP dapat digunakan untuk mengekstraksi informasi tertentu dari dokumen teks, seperti nama, tanggal, atau lokasi.
 - 5) Analisis teks
NLP dapat digunakan untuk mengidentifikasi topik, kategori, dan tema dalam teks, yang dapat membantu dalam penelitian atau analisis data.
- d) Algoritma *Computer Vision*
Teknologi *Computer Vision* adalah sebuah cabang dari kecerdasan buatan (*Artificial Intelligence*) yang mengajarkan komputer untuk memahami dan memproses informasi visual dari dunia nyata. Teknologi ini memungkinkan komputer untuk menganalisis, memahami, dan mengambil tindakan berdasarkan data visual yang diterima dari kamera atau sensor lainnya (Boring, 2002). Beberapa teknologi utama dalam *Computer Vision* meliputi:

- 1) *Image Recognition*: Teknologi ini memungkinkan komputer untuk mengenali objek, wajah, atau benda tertentu dalam gambar atau video.
- 2) *Object Detection*: Teknologi ini memungkinkan komputer untuk mendeteksi keberadaan dan lokasi objek dalam gambar atau video.
- 3) *Image Segmentation*: Teknologi ini memungkinkan komputer untuk memisahkan objek dari latar belakang dalam gambar atau video.
- 4) *Face Recognition*: Teknologi ini memungkinkan komputer untuk mengenali wajah dan membedakannya dari wajah orang lain.
- 5) *Video Analysis*: Teknologi ini memungkinkan komputer untuk menganalisis video untuk mendeteksi objek, gerakan, dan situasi yang mencurigakan atau berbahaya. Algoritma ini digunakan untuk memproses dan menganalisis gambar dan video. Contoh algoritma *computer vision* termasuk *Object Detection*, *Image Classification*, dan *Image Segmentation*.

e) Algoritma Bayesian

Algoritma ini digunakan untuk menghitung probabilitas suatu peristiwa berdasarkan informasi yang tersedia. Contoh algoritma bayesian termasuk Naive Bayes, Bayesian Networks, dan Markov Chain Monte Carlo (MCMC) (Thompson *et al.*, 1992).

f) Algoritma Reinforcement Learning

Algoritma ini digunakan untuk membuat agen cerdas yang dapat belajar dari interaksi dengan lingkungan. Contoh algoritma *reinforcement learning* termasuk *Q-Learning*, SARSA, dan *Actor-Critic*. *Reinforcement Learning* (RL) adalah sebuah cabang dari Machine Learning yang mengajarkan sebuah program atau agen untuk belajar dan mengambil keputusan dalam sebuah lingkungan yang

dinamis. RL memberikan sebuah metode bagi program untuk memperbaiki performanya melalui trial and error, yaitu dengan mencoba berbagai pilihan dan belajar dari hasilnya (Heyes, 2018; Miłkowski & Hohol, 2021).

Konsep dasar dari RL adalah adanya agen yang berada dalam sebuah lingkungan dan harus memilih tindakan yang tepat untuk mencapai tujuannya. Setiap tindakan yang diambil oleh agen akan memengaruhi lingkungan dan memberikan feedback berupa reward atau hukuman, yang nantinya akan digunakan untuk memperbaiki tindakan agen selanjutnya. Tujuan akhir dari RL adalah untuk memaksimalkan jumlah reward yang diterima oleh agen dari lingkungan.

g) *Algoritma Evolutionary*

Evolutionary Computing adalah salah satu bidang dalam kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada pengembangan algoritma yang terinspirasi dari prinsip-prinsip seleksi alam dalam evolusi biologis. Tujuannya adalah untuk menghasilkan solusi yang optimal atau paling sesuai untuk masalah yang kompleks dengan cara simulasi evolusi dalam populasi. Algoritma ini digunakan untuk memecahkan masalah dengan menggunakan strategi evolusi. Contoh algoritma *evolutionary* termasuk *Genetic Algorithm*, *Evolutionary Programming*, dan *Evolutionary Strategies*.

DAFTAR PUSTAKA

- Bender, A. 2022. Toward Greater Integration: Fellows Perspectives on Cognitive Science. In *Topics in Cognitive Science* (Vol. 14, Issue 1). <https://doi.org/10.1111/tops.12599>
- Boring, R. L. 2002. Human-Computer Interaction as Cognitive Science. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 46(21). <https://doi.org/10.1177/154193120204602103>
- Calzavarini, F., & Paternoster, A. 2022. The semantic view of computation and the argument from the cognitive science practice. *Synthese*, 200(2). <https://doi.org/10.1007/s11229-022-03542-z>
- Courtial, J. P., & Law, J. 1989. A Co-Word Study of Artificial Intelligence. *Social Studies of Science*, 19(2), 301–311. <https://doi.org/10.1177/030631289019002005>
- Dennett, D. C. 2009. The Part of Cognitive Science That Is Philosophy. *Topics in Cognitive Science*, 1(2). <https://doi.org/10.1111/j.1756-8765.2009.01015.x>
- Engbert, R., Rabe, M. M., Schwetlick, L., Seelig, S. A., Reich, S., & Vasishth, S. 2022. Data assimilation in dynamical cognitive science. In *Trends in Cognitive Sciences* (Vol. 26, Issue 2). <https://doi.org/10.1016/j.tics.2021.11.006>
- Goldstone, R. L. 2019. Becoming Cognitive Science. *Topics in Cognitive Science*, 11(4). <https://doi.org/10.1111/tops.12463>
- Heyes, C. 2018. Enquire within: Cultural evolution and cognitive science. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1743). <https://doi.org/10.1098/rstb.2017.0051>

- Hipólito, I. 2022. Cognition Without Neural Representation: Dynamics of a Complex System. *Frontiers in Psychology*, 12. <https://doi.org/10.3389/fpsyg.2021.643276>
- Ho, M. K., & Griffiths, T. L. 2022. Cognitive Science as a Source of Forward and Inverse Models of Human Decisions for Robotics and Control. In *Annual Review of Control, Robotics, and Autonomous Systems* (Vol. 5). <https://doi.org/10.1146/annurev-control-042920-015547>
- Khalifa, K., Islam, F., Gamboa, J. P., Wilkenfeld, D. A., & Kostić, D. 2022. Integrating Philosophy of Understanding With the Cognitive Sciences. *Frontiers in Systems Neuroscience*, 16. <https://doi.org/10.3389/fnsys.2022.764708>
- Mendoza-Collazos, J., & Zlatev, J. 2022. A Cognitive-Semiotic Approach to Agency: Assessing Ideas from Cognitive Science and Neuroscience. *Biosemiotics*, 15(1). <https://doi.org/10.1007/s12304-022-09473-z>
- Miłkowski, M., & Hohol, M. 2021. Explanations in cognitive science: unification versus pluralism. In *Synthese* (Vol. 199). <https://doi.org/10.1007/s11229-020-02777-y>
- Mittman, B. 2018. Workshop in Cognitive Sciences - Computer Chess. *ICGA Journal*, 5(2). <https://doi.org/10.3233/icg-1982-5211>
- Saharrea, J. M. 2022. Out-of-place learning as a pragmatist critique of the cognitive sciences. *Sophia(Ecuador)*, 32. <https://doi.org/10.17163/soph.n32.2022.08>
- Sands, P. 2019. Addressing cognitive load in the computer science classroom. *ACM Inroads*, 10(1). <https://doi.org/10.1145/3210577>

- Solanki, A., & Jain, D. K. 2020. Emerging Trends and Applications in Cognitive Computing. *Recent Advances in Computer Science and Communications*, 13(5). <https://doi.org/10.2174/266625581305201028104513>
- Stella, M. 2022. Cognitive Network Science for Understanding Online Social Cognitions: A Brief Review. *Topics in Cognitive Science*, 14(1). <https://doi.org/10.1111/tops.12551>
- Thompson, E., Palacios, A., & Varela, F. J. 1992. vision as a case for cognitive science. In *BEHAVIORAL AND BRAIN SCIENCES* (Vol. 15).

BIODATA PENULIS



Rizal Furqan Ramadhan., S.Kom., M.T

Dosen Universitas Islam Negeri Sayyid Ali Rahmatullah
Tulungagung

Penulis lahir dan berdomisili di Trenggalek Provinsi Jawa Timur. Penulis merupakan Dosen Tetap bukan PNS di Universitas Islam Negeri Sayyid Ali Rahmatullah Tulungagung. Penulis merupakan dosen dengan bidang keilmuan Sistem Informasi/Elektro Informatika/Teknik Informatika dan ditempatkan di Fakultas Ekonomi dan Bisnis Islam dengan mengampu matakuliah Digital Marketing serta matakuliah digital lainnya. Sebelumnya, penulis memiliki pengalaman mengajar di berbagai universitas antara lain PROBIS Universitas Negeri Malang, Universitas Kanjuruhan Malang serta Universitas Islam Kadir dengan mengampu matakuliah Informatika seperti Database, perancangan sistem informasi, *mobile computing* dan lain-lain.

Penulis menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang (2010-2014) kemudian melanjutkan studi S2 di Universitas Brawijaya Malang pada Program Studi Magister Teknik Elektro Minat Sistem Komunikasi dan Informatika

(2014-2016). Penulis aktif menulis jurnal ilmiah, buku serta riset bidang informatika/komputer/sistem informasi.

BIODATA PENULIS



Wahyuddin S., S.Kom., M.Kom
Dosen STMIK Amika Soppeng

Wahyuddin S. was born at Malaka-Bone-Sulawesi Selatan in 1992. In 2011 he attended STMIK Dipanegara Makassar and was completed in 2015. He was completed after attending 7 semesters and active on an XPcom (Extreme Programmer Computer) campus organization. He was also active as a lecturer assistant for three semesters and taught several courses on programming. He continued his Master of Information systems at UNIKOM Bandung in 2016 and was completed in April 2019. He worked as a lecturer at a campus (STMIK Amika Soppeng) 2019 to present and also a Freelance Web Programmer. Has competence in the field of software engineer, application developer, multimedia, web developer, network security, and data analyst.

BIODATA PENULIS



Fahmy Rinanda Saputri, S.T., M.Eng.

Dosen Program Studi Teknik Fisika
Fakultas Teknik dan Informatika Universitas Multimedia
Nusantara

Penulis lahir pada bulan Agustus 1993 di Cilacap. Penulis adalah dosen tetap pada Program Studi Teknik Fisika Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada Jurusan Teknik Fisika Universitas Gadjah Mada dan melanjutkan studi pada Program Magister Teknik Fisika Universitas Gadjah Mada. Penulis menekuni bidang instrumentasi, fisika bangunan, dan energi baru terbarukan.

BIODATA PENULIS



Jarwo, ST, MM

Dosen Program Studi Teknik Informatika
Sekolah Tinggi Teknologi Pomosda Nganjuk

Penulis lahir di Magetan pada tanggal 07 Mei 1981. Penulis adalah dosen tetap pada Program Studi Teknik Informatika Sekolah Tinggi Teknologi Pomosda Nganjuk. Menyelesaikan Pendidikan S1 Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Pomosda Nganjuk dan melanjutkan S2 pada Universitas Islam Kediri Kediri pada jurusan Manajemen dengan konsentrasi Sistem Informasi Manajemen. Pada perguruan tinggi homebase, penulis mengampu mata kuliah Mata Kuliah Bahasa Otomata, Teknik Kompilasi, Matematika Diskrit, Kecerdasan Buatan dan Mekatronika IOT. Penulis aktif dan tertarik pada bidang Pengelolaan Informasi, Decision Support System dan beberapa kajian khas pada Kecerdasan Buatan. Penulis dapat dihubungi melalui surat elektronik jarwosttpmd@gmail.com.

BIODATA PENULIS



Ir. Johni S. Pasaribu, MT.

Dosen Program Studi Sistem Informasi
Fakultas IT Politeknik Piksi Ganesha

Penulis lahir di Medan tanggal 1 Juni 1966 dari pasangan HA Pasaribu (alm) dan Taruli Tambunan (alm). Penulis adalah anak keempat dari enam bersaudara. Pendidikan yang ditempuh yaitu SD Methodist Medan lulus tahun 1979, SMPN 88 Jakarta lulus tahun 1982 dan SMAN 3 Jakarta lulus tahun 1985. Kemudian melanjutkan studi di ITB jurusan Teknik Fisika spesialisasi Instrumentasi dan Kontrol. Penulis melakukan penelitian untuk tugas akhir di PT IPTN dengan judul Perancangan Sistem Kendali Fly By Wire Gerak Longitudinal Pada Engineering Flight Simulator Pesawat N250-100 dan menyelesaikan gelar sarjana teknik Insinyur (Ir) tahun 1992.

Kemudian penulis bekerja di PT. IPTN (sekarang PT Dirgantara Indonesia) di Departemen Iron Bird kemudian berpindah di Departemen Engineering Flight Simulator (EFS). Penulis terlibat dalam Operation and Maintenance untuk EFS sehingga simulator N250-100 ini dapat digunakan saat dilakukan pengujian-pengujian pesawat melalui simulator

sebelum pengujian langsung pesawat di udara. Tahun 1998 penulis melanjutkan studi S2 di ITB pada jurusan Teknik Informatika untuk spesialisasi Software Engineering, dimana ilmu yang diperoleh dapat diaplikasikan dalam pekerjaan di lapangan. Tahun 2001 penulis menyelesaikan pendidikan S2 dengan gelar MT mengambil penelitian yang berjudul Pengembangan Perangkat Lunak Simulasi Sistem Perasa Kendali Elevator Pesawat N250-100 dimana penelitian ini juga dilakukan di Engineering Flight Simulator (EFS) N250-100. Tahun 2004 penulis mulai aktif mengajar di Perguruan Tinggi di Bandung diantaranya pernah mengajar di Unikom, ITHB, STT Telkom (kini TelU). Kini penulis penuh mengajar dan menjadi dosen tetap di Politeknik Piksi Ganesha. Bidang keahlian mengajar dan penelitian penulis adalah Artificial Intelligence (AI), Decision Support System (DSS), Software Engineering dan Software Testing. Penulis juga aktif melakukan bimbingan mahasiswa S1 dan publikasi di jurnal-jurnal nasional maupun internasional.

BIODATA PENULIS



Elisawati, S.Kom., M.Kom.

Dosen Program Studi Sistem Informasi
STMIK Dumai

Penulis lulusan STIKI Malang, Program Studi Teknik Informatika pada tahun 2003, dan memperoleh gelar Master Ilmu Komputer di Universitas Putra Indonesia (UPI) Padang pada tahun 2010 dengan konsentrasi Teknologi Informasi. Saat ini penulis bekerja sebagai Dosen Tetap di STMIK Dumai dan aktif mengajar di beberapa perguruan Tinggi yang ada di Dumai.

BIODATA PENULIS



Muhamad Faza Almaliki, S.T.

Lulusan Program Studi S1 Teknik Informatika
Fakultas Teknik Universitas Halu Oleo

Penulis lahir di Kendari tanggal 02 Maret 2001. Penulis adalah lulusan pada Program Studi S1 Teknik Informatika Fakultas Teknik, Universitas Halu Oleo. Menyelesaikan pendidikan di SMPN 1 dan SMAN 4 Kendari, Sulawesi Tenggara pada tahun 2012 dan 2018 serta telah menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika, Universitas Halu Oleo pada awal tahun 2023. Penulis mulai menekuni bidang menulis sejak tahun 2022, yakni ketika penulis sedang dalam akhir tahapan menyelesaikan masa studi.

Pengalaman penulis dalam menyusun sebuah buku dapat dikatakan dimulai dengan buku yang saat ini yang sedang disusun, namun pengalaman dalam bidang penulisan, penulis telah menerbitkan jurnal pada tahun 2022 dan 2023 yang dimana kedua jurnal tersebut masih berkaitan dengan bidang informatika.

BIODATA PENULIS



Citra Nurina Prabiantissa, S.S.T., M.Tr.Kom.
Dosen Program Studi Teknik Informatika
Fakultas Teknik Elektro dan Teknologi Informasi
Institut Teknologi Adhi Tama Surabaya

Penulis lahir di Bondowoso pada tanggal 29 Mei 1994. Penulis merupakan dosen tetap di Institut Teknologi Adhi Tama Surabaya Program Studi Teknik Informatika sejak tahun 2019. Penulis telah menyelesaikan pendidikan S1 Program Studi Teknik Informatika di Politeknik Negeri Malang pada tahun 2016 dan S2 Program Studi Teknik Informatika dan Komputer di Politeknik Elektronika Negeri Surabaya pada tahun 2019. Penulis juga aktif sebagai peneliti dan telah menghasilkan karya ilmiah yang diterbitkan pada jurnal nasional bereputasi dan seminar internasional. Bidang penelitian penulis adalah *digital image processing*, *computer vision* dan *artificial intelligent*.

BIODATA PENULIS

Melissa Indah Fianty, S.Kom., MMSI

Dosen Program Studi Sistem Informasi
Fakultas Teknik dan Informatika Universitas Multimedia
Nusantara

Penulis lahir di Bengkulu tanggal 13 Januari 1992. Penulis adalah dosen tetap pada Program Studi Sistem Informasi Fakultas Teknik dan Informatika, Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada Teknik Informatika dan melanjutkan S2 pada Manajemen Sistem Informasi di Bina Nusantara University.

BIODATA PENULIS



St. Hajrah Mansyur

Dosen tetap pada Fakultas Ilmu Komputer Universitas Muslim Indonesia (Fikom UMI)

Ir. St. Hajrah Mansyur, S.Kom., M.Cs., MTA. Lahir di Ujung Pandang, 19 Januari 1985. Menempuh jenjang pendidikan S1 pada program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Muslim Indonesia (UMI) Makassar tahun 2002, dan melanjutkan pendidikan S2 pada program pascasarjana prodi Ilmu Komputer, Universitas Gadjah Mada Yogyakarta tahun 2010. Karir di mulai sebagai dosen tetap pada Fakultas Ilmu Komputer Universitas Muslim Indonesia (Fikom UMI) mulai tahun 2007 sampai sekarang. Bidang kajian yang menjadi tanggung jawab penulis di Fikom UMI adalah Sistem Informasi. Adapun pelaksanaan tri darma perguruan tinggi, penulis melakukan pengajaran, penelitian, dan pengabdian kepada masyarakat yang sesuai dengan kompetensi di bidang IT. Selain itu, saat ini penulis aktif mendampingi mahasiswa dalam berbagai kegiatan seperti Program Kreativitas Mahasiswa (PKM), dan dipercayakan sebagai Tim Penilai PKM tingkat nasional Kementerian Pendidikan, Kebudayaan, Riset, dan teknologi mulai tahun

2019 hingga sekarang. Sebagai koordinator perguruan tinggi pada kegiatan Merdeka Belajar Kampus Merdeka (MBKM) yang dilaksanakan oleh Kemendikbudristek seperti Kampus Mengajar (KM), Magang dan Studi Independent Bersertifikat (MSIB), dan Program Pertukaran Mahasiswa (PPM).

email : hajrah.mansyur@umi.ac.id

BIODATA PENULIS



Angga Aditya Permana
Dosen Informatika
Universitas Multimedia Nusantara

Angga Aditya Permana (lahir pada Desember 1989 di Jakarta) seorang dosen full time di bidang Computer Science pada Universitas Multimedia Nusantara sejak tahun 2021, fokus penelitian pada kriptografi, steganografi serta keamanan computer, namun pada tahun 2021 sedang mendalami topik bioinformatika dan network science. Angga juga memiliki hobi yaitu touring dan juga bermain bulutangkis, saat ini sedang menjadi mahasiswa program doctoral pada IPB University. Memulai karir pertama kali sebagai Network Enginner tahun 2011 dan memulai profesinya sebagai dosen pada 2013 di kampus swasta yang ada di Jakarta, pernah juga mengajar di kampus negeri yang berada di Jakarta dan Tangerang, juga pernah di undang menjadi dosen tamu untuk mengenalkan bioinformatika di kampus negeri di Jakarta. Terimakasih..

BIODATA PENULIS



Ratna Dewi

Dosen di Universitas Gajah Putih

Penulis Lahir 05 Oktober 1990 di Paya kolak Kecamatan Celala Kabupaten Aceh Tengah Pada tahun 2013 Penulis menyelesaikan Pendidikan S1 Di Sekolah Tinggi Teknik Harapan Medan. Dengan Jurusan SISTEM INFORMASI, kemudian setelah menyelesaikan S1 penulis mengajar di Sekolah SMK Kebangsaan Medan pada tahun 2013 sampai dengan 2016. kemudian di tahun 2018 Penulis Menempuh Pendidikan S2 TEKNIK INFORMATIKA di UNIVERSITAS PUTRA INDONESIA PADANG (YPTK). Dua tahun kemudian Penulis menyelesaikan studi S2 di prodi TEKNIK INFORMATIKA pada tahun 2019.

Di tahun 2020 penulis mengajar di Universitas Gajah Putih di daerah tempat tinggal saya kota Takengon Kabupaten Aceh Tengah. sampai saat ini saya masih menjadi Dosen Tetap. dan untuk mewujudkan karir sebagai dosen profesional, Penulis mulai aktif menulis meneliti dari bidang kepakaran tersebut yaitu tentang ilmu komputer. Harapan penulis ingin memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Email Penulis: dewi90ratna90@gmail.com

BIODATA PENULIS



Risanto Darmawan, MM, M.Kom.

Dosen Program Studi Teknik Informatika
Fakultas Teknik, Universitas Krisnadwipayana

Penulis menamatkan pendidikan Diploma D2 Informatika di Fakultas Politeknik Pertanian IPB Bogor di tahun 1991. Program Sarjana (S1) di Universitas Budi Luhur, Jakarta, Fakultas Teknologi Informasi, Sistem Informasi di tahun 2000 dan Universitas Terbuka, Tangerang Selatan, Fakultas Ekonomi, Prodi Manajemen di 2003. Penulis juga menyelesaikan program Pasca Sarjana (S2) di Universitas Pelita Harapan, Karawaci, Fakultas Business & Management, spesialisasi Marketing di 2003 dan di Sekolah Pascasarjanam, Prodi Ilmu Komputer, IPB Bogor di tahun 2006.

BIODATA PENULIS



Suwito Pomalingo, S.Kom., M.Kom.

Dosen Program Studi Informatika
Fakultas Teknik dan Informatika
Universitas Multimedia Nusantara

Penulis adalah dosen tetap pada Program Studi Informatika Fakultas Teknik dan Informatika Universitas Multimedia Nusantara. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika dan melanjutkan S2 pada Program Magister Informatika Universitas Islam Indonesia.

Saat ini sementara menyelesaikan studi Doktoral dengan topik penelitian di bidang Security Science dengan pendekatan Deep Learning. Penulis menekuni bidang keilmuan meliputi Cyber Security, Malware Analytics, Data Science, Machine Learning, dan Deep Learning, selain itu, penulis juga memiliki beberapa serifikasi internasional di bidang Cyber Security, Digital Forensics, Data Science dan Machine Learning.

BIODATA PENULIS



Moh Safii

Dosen Prodi Ilmu Perpustakaan Universitas Negeri Malang

Penulis lahir di Kota Malang, alumni Teknik Informatika ITS Surabaya dan S2 Ilmu Perpustakaan Universitas Indonesia. Sehari- hari berdinasi di Fakultas Sastra, Prodi Ilmu Perpustakaan Universitas Negeri Malang.

Email : moh.safii@um.ac.id

Web : <http://mohsafii.blog.um.ac.id/>