

METODE DAN PENERAPAN SISTEM PAKAR

**Hasanuddin Sirait, Rini Mayasari, Nono Heryana, Johni S Pasaribu,
Wahyuddin S, Ronald Belferik, Aisyah Mutia Dawis, Andryanto. A,
Edison Ulung Pandapotan Simanuhuruk, Reagan Surbakti Saragih,
Herlina Latipa Sari, Andri Sukmaindrayana, Raimon Efendi**



METODE DAN PENERAPAN SISTEM PAKAR

Hasanuddin Sirait

Rini Mayasari

Nono Heryana

Johni S Pasaribu

Wahyuddin S

Ronald Belferik

Aisyah Mutia Dawis

Andryanto. A

Edison Ulung Pandapotan Simanuhuruk

Reagan Surbakti Saragih

Herlina Latipa Sari

Andri Sukmaindrayana

Raimon Efendi



GET PRESS INDONESIA

METODE DAN PENERAPAN SISTEM PAKAR

Penulis :

Hasanuddin Sirait
Rini Mayasari
Nono Heryana
Johani S Pasaribu
Wahyuddin S
Ronald Belferik
Aisyah Mutia Dawis
Andryanto. A
Edison Ulung Pandapotan Simanuhuruk
Reagan Surbakti Saragih
Herlina Latipa Sari
Andri Sukmaindrayana
Raimon Efendi

ISBN : 978-623-198-532-3

Editor : Mila Sari., S.ST., M.Si

Penyunting : Tri Putri Wahyuni,S.Pd

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

Penerbit : GET PRESS INDONESIA

Anggota IKAPI No. 033/SBA/2022

Redaksi :

Jl. Pasir Sebelah No. 30 RT 002 RW 001
Kelurahan Pasie Nan Tigo Kecamatan Koto Tengah
Padang Sumatera Barat
Website : www.getpress.co.id
Email : adm.getpress@gmail.com

Cetakan pertama, 28 Juli 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 Metode Dan Penerapan Sistem Pakar Ini.

Buku Ini Membahas Pendahuluan, Komponen Utama dan Domain Sistem Pakar, Konsep Dasar Penalaran Sistem Pakar, Elemen dan Struktur Dasar Sistem Pakar, Metode Pengembangan Sistem Pakar, Metode Aktifasi & Rules Sistem Pakar, Metode Perancangan Aplikasi & Tools Sistem Pakar, Penerapan Algoritma K-Nearest Neighbour dalam Sistem Pakar, Penerapan Algoritma Marcov dalam Sistem Pakar, Penerapan Algoritma Naives Bayes dalam Sistem Pakar, Penerapan Algoritma Certainty Factor Dalam Sistem Pakar, Penerapan Algoritma Forward Chaining Dalam Sistem Pakar, Penerapan Metode infrensi Fuzzy dalam Sistem Pakar.

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, 28 Juli 2023

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xii
BAB 1 PENDAHULUAN	1
1.1 Pendahuluan.....	1
1.2 Tujuan Sistem Pakar (<i>Expert System Purpose</i>)	2
1.3 Struktur Sistem Pakar (<i>expert system structure</i>)	5
1.4 Manfaat Sistem Pakar (<i>Expert System Benefits</i>)	7
1.5 Kelebihan Sistem Pakar (<i>Advantages of Expert Systems</i>).....	9
1.6 Kekurangan Sistem Pakar (<i>Disadvantages of Expert Systems</i>).....	10
1.7 Penerapan Sistem Pakar	10
DAFTAR PUSTAKA	13
BAB 2 KOMPONEN UTAMA DAN DOMAIN SISTEM PAKAR.....	15
2.1 Komponen Utama Sistem Pakar	15
2.1.1 Basis Pengetahuan	15
2.1.2 Mesin Inferensi.....	23
2.1.3 Antarmuka Pengguna	25
2.3 Domain Sistem Pakar	27
2.3.1 Pengertian Domain dalam Sistem Pakar.....	27
2.3.2 Pemilihan dan Analisis Domain.....	28
2.3.3 Contoh-Contoh Domain Sistem Pakar	32
2.3.4 Manfaat dan Tantangan dalam Penentuan Domain.....	34
DAFTAR PUSTAKA	36
BAB 3 KONSEP DASAR PENALARAN SISTEM PAKAR.....	39
3.1 Pengertian.....	39
3.1.1 Definisi Penalaran	39

3.1.2 Perbedaan Penalaran Sistem Pakar dengan Kecerdasan Buatan	39
3.2 Jenis-jenis Penalaran dalam Sistem Pakar	40
3.2.1 Penalaran Berbasis Aturan (<i>Rule-Based Reasoning</i>).....	40
3.2.2 Penalaran Berbasis Kasus (<i>Case-Based Reasoning</i>).....	44
3.2.3 Penalaran Berbasis Objek (<i>Object-Based Reasoning</i>).....	46
3.3 Model Penalaran Sistem Pakar	47
3.3.1 Model Klasik	47
3.3.2 Model Probabilistik	51
3.3.3 Model Fuzzy	52
DAFTAR PUSTAKA	55
BAB 4 ELEMEN DAN STRUKTUR DASAR SISTEM PAKAR.....	57
4.1 Pendahuluan.....	57
4.2 Konsep Umum Sistem Pakar.....	58
4.2.1 Pengetahuan dan Pakar	59
4.2.2 <i>Rules</i> (aturan) sebagai Teknik Representasi Pengetahuan.....	59
4.2.3 Apa itu Sistem Pakar?	62
4.3 Anggota tim pengembangan sistem pakar	63
4.4 Elemen dan struktur dasar sistem pakar	66
DAFTAR PUSTAKA	71
BAB 5 METODE PENGEMBANGAN SISTEM PAKAR.....	73
5.1 Pendahuluan.....	73
5.2 Pengertian Metode Penerapan Sistem Pakar	75
5.3 Metode Penalaran Berbasis Aturan (<i>Rule-Based Reasoning</i>).....	76
5.4 Metode Pengetahuan Berbasis Kasus (<i>Case-Based Reasoning</i>).....	78
5.5 Metode Jaringan (<i>Semantic Network</i>)	80

5.6 Metode Jaringan Saraf Tiruan (<i>Artificial Neural Network</i>)	81
5.7 Metode Logika Fuzzy (<i>Fuzzy Logic</i>)	82
5.8 Metode Sistem Berbasis Agen (<i>Agent-Based Systems</i>)	84
5.9 Metode Penalaran (<i>Probabilistic Reasoning</i>)	85
5.10 Metode Genetika (<i>Genetic Algorithms</i>)	86
DAFTAR PUSTAKA	88
BAB 6 METODE AKTIFASI & RULES SISTEM PAKAR	91
6.1 Pengenalan Metode Aktivasi	91
6.1.1 Definisi dan Peran Metode Aktivasi dalam Sistem Pakar	91
6.1.2 Faktor-faktor yang Mempengaruhi Pemilihan Metode Aktivasi	91
6.2 Metode Aktivasi dalam Sistem Pakar	92
6.2.1 Forward Chaining	92
6.2.2 Backward Chaining	93
6.2.3 Certainty Factor	94
6.2.4 Weighted Sum Model	95
6.2.5 Perbandingan dan Pemilihan Metode Aktivasi yang Tepat	95
6.3 Aturan dalam Sistem Pakar	96
6.3.1 Pengenalan aturan dalam Sistem Pakar	98
6.3.2 Identifikasi Pengetahuan Pakar yang Relevan	99
6.4 Strategi Penulisan Aturan yang Efektif dalam Sistem Pakar	100
6.4.1 Format Penulisan Aturan yang Jelas dan Tepat	100
6.4.2 Menangani Ketidakpastian dalam Aturan	102
DAFTAR PUSTAKA	104
BAB 7 METODE PERANCANGAN APLIKASI & TOOLS SISTEM PAKAR	105
7.1 Perancangan Aplikasi	105
7.2 Metode Perancangan Aplikasi	108
7.3 Tools Sistem Pakar	114

7.3.1 Bahasa Pemrograman.....	114
7.3.2 Database.....	115
7.3.3 Web Server	116
7.3.4 Browser	116
DAFTAR PUSTAKA	117
BAB 8 PENERAPAN ALGORITMA K-NEAREST	
NEIGHBOUR DALAM SISTEM PAKAR.....	121
8.1 Pendahuluan.....	121
8.2 Algoritma <i>K-Nearest Neighbour</i> (KNN).....	122
8.2.1 Cara kerja algoritma KNN dalam klasifikasi data..	122
8.2.2 Metrik jarak yang digunakan dalam algoritma KNN	123
8.2.3 Cara kerja algoritma KNN dalam regresi	125
8.3 Penerapan KNN Dalam Berbagai Bidang.....	129
8.3.1 Kesehatan.....	129
8.3.2 Keuangan.....	129
8.3.3 Teknologi.....	129
DAFTAR PUSTAKA	131
BAB 9 PENERAPAN ALGORITMA NAIVE BAYES	
DALAM SISTEM PAKAR.....	133
9.1 Pendahuluan.....	133
9.2 Pengenalan Sistem Pakar	135
9.3 Penerapan Algoritma Neive Bayes dalam Sistem Pakar.	139
9.4 Studi Kasus	147
9.5 Tantangan dalam Implementasi Algoritma Naïve Bayes Dalam Sistem Pakar	152
9.6 Peluang dan Potensi Pengembangan Algoritma Naive Bayes Dalam Sistem Pakar	154
DAFTAR PUSTAKA	158
BAB 10 PENERAPAN ALGORITMA MARKOV	
DALAM SISTEM PAKAR.....	161
10.1 Pendahuluan.....	161

10.2 Dasar-dasar Algoritma Markov.....	162
10.3 Konsep Model Markov.....	164
10.4 Reprerentasi Model Markov.....	166
10.5 Pengambilan Keputusan.....	169
10.6 Evaluasi dan Pemeliharaan.....	170
10.7 Metrik Evaluasi performa model Markov dalam sistem pakar	172
10.8 Pemeliharaan Model Markov.....	174
DAFTAR PUSTAKA	176
BAB 11 PENERAPAN ALGORITMA CERTAINITY	
FACTOR DALAM SISTEM PAKAR	179
11.1 Ketidakpastian (<i>Uncertainty</i>).....	179
11.2 Algoritma <i>Certainty Factor</i> (CF)	181
11.3 Penerapan Algoritma <i>Certainty Factor</i> (CF) dalam Sistem Pakar	183
DAFTAR PUSTAKA	191
BAB 12 PENERAPAN ALGORITMA BACKWARD	
CHAINING DALAM SISTEM PAKAR.....	193
12.1 Sistem Pakar Berbasis Kaidah.....	193
12.2 Strategi Pencarian Pada Mesin Inferensi	194
12.2.1 Runut Balik (backward chaining)	194
12.2.2 Algoritma Runut Balik.....	195
12.3 Analisis dan Perancangan Penerapan Algoritma Runut Balik (<i>Backward Chaining</i>).....	200
12.3.1 Analisis Sistem.....	200
12.3.2 Cara Representasi Pengetahuan	204
12.4 Diagram Konteks dari Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining	210

12.4.1 DFD Level 1 dari Diagram Konteks Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining	211
12.4.2 DFD Level 1 dari Proses 1 Pengolahan Data Pasien	212
12.4.3 DFD Level 1 dari Proses 2 Pengolahan Data Penyakit	213
12.4.4 DFD Level 1 dari Proses 3 Pengolahan Data Gejala	214
12.4.5 DFD Level 1 dari Proses 4 Pengolahan Data Pengetahuan	215
12.4.6 DFD Level 1 dari Proses 6 Pengolahan Laporan.....	216
12.7 Rancangan <i>Entity Relationship Diagram</i> (ERD)	216
12.8 Rancangan Struktur Tabel Basis Data	218
12.9 Rancangan Form dan Laporan.....	220
12.9.1 Rancangan Form Menu Utama	220
12.9.2 Rancangan Form Login	221
12.9.3 Rancangan Form Data Pasien.....	222
12.9.4 Rancangan Form Data Penyakit.....	222
12.9.5 Rancangan Form Data Gejala.....	223
12.9.6 Rancangan Form Data Pengetahuan	223
12.9.7 Rancangan Form Diagnosa	224
12.9.8 Rancangan Form Data Hasil Diagnosa	224
12.9.9 Rancangan Laporan Data Pasien	225
12.9.10 Rancangan Laporan Data Penyakit	225
12.9.11 Rancangan Laporan Data Gejala	226
12.9.12 Rancangan Laporan Data Pengetahuan.....	226
12.9.13 Rancangan Laporan Data Hasil Diagnosa.....	227
12.10 Impelementasi Form dan Laporan	227
12.10.1 Nama Form Yang Dibuat	227
12.10.2 Nama Tabel Yang Dibuat	228

12.10.3 Form Menu Utama	229
12.10.4 Form Login	229
12.10.5 Form Data Pasien.....	230
12.10.6 Form Data Penyakit.....	230
12.10.7 Form Data Gejala.....	231
12.10.8 Form Data Pengetahuan	231
12.10.9 Form Diagnosa	232
12.10.10 Form Hasil Diagnosa	232
12.10.11 Laporan Data Pasien	233
12.10.12 Laporan Data Penyakit	233
12.10.13 Laporan Data Gejala	234
12.10.14 Laporan Data Pengetahuan.....	234
12.10.15 Laporan Data Hasil Diagnosa	235
DAFTAR PUSTAKA	236
BAB 13 PENERAPAN METODE INFERENSY FUZY	
DALAM SISTEM PAKAR	237
13.1 Pendahuluan.....	237
13.2 Konsep Fuzzy Inference System	239
13.3 Penerapan Fuzzy Inference System (Studi Kasus pada pemberian Kredit Perbankan).....	241
13.4 Kelebihan dan Kelemahan Fuzzy Logic	251
DAFTAR PUSTAKA	255
BIODATA PENULIS	

DAFTAR GAMBAR

Gambar 1.1. Siklus Sistem Pakar (<i>expert system diagram</i>).....	7
Gambar 2.1. Komponen Utama Sistem Pakar.....	15
Gambar 2.2. Struktur Basis Pengetahuan.....	18
Gambar 2.3. Representasi Pengetahuan	20
Gambar 2.4. Proses Akuisisi Pengetahuan.....	22
Gambar 3.1. Format Aturan IF-THEN dalam Sistem Pakar	41
Gambar 3.2. Forward Chaining.....	42
Gambar 3.3. Backward Chaining	43
Gambar 3.4. MYCIN	44
Gambar 3.5. Siklus CBR.....	45
Gambar 3.6. Contoh dari Bayesian Network	52
Gambar 3.7. Logika Fuzzy.....	53
Gambar 4.1. Tahapan pengembangan sistem pakar	61
Gambar 4.2. Enam cabang utama Kecerdasan Buatan	62
Gambar 4.3. <i>Stake holders</i> pembuatan proyek ES (sistem pakar).....	64
Gambar 4.4. (a) Model sistem produksi Newell dan Simon dan (b) Struktur dasar sistem pakar berbasis aturan	67
Gambar 4.5. Struktur lengkap dari sistem pakar berbasis aturan	70
Gambar 7.1. Perancangan aplikasi	105
Gambar 7.2. Tim pengembang melakukan identifikasi.....	106
Gambar 7.3. Tim pengembang memilih bahasa pemrograman	107
Gambar 7.4. Metode Agile.....	109
Gambar 7.5. Metode Waterfall	110
Gambar 7.6. Metode Lean UX	112
Gambar 12.1. Contoh Backward Chaining.....	195

Gambar 12.2.	Diagram Alir Algoritma Runut Balik (<i>Backward Chaining</i>).....	199
Gambar 12.3.	Diagram Konteks dari Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining.....	210
Gambar 12.4.	DFD Level 1 dari Diagram Konteks dari Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining.....	211
Gambar 12.5.	DFD Level 1 dari Proses 1 Pengolahan Data Pasien	212
Gambar 12.6.	DFD Level 1 dari Proses 2 Pengolahan Data Penyakit	213
Gambar 12.7.	DFD Level 1 dari Proses 3 Pengolahan Data Gejala	214
Gambar 12.8.	DFD Level 1 dari Proses 4 Pengolahan Data Pengetahuan.....	215
Gambar 12.9.	DFD Level 1 dari Proses 6 Pengolahan Laporan.....	216
Gambar 12.10.	Rancangan <i>Entity Relationship</i> <i>Diagram (ERD)</i>	217
Gambar 12.11.	Rancangan Form Utama.....	221
Gambar 12.12.	Rancangan Form Login.....	221
Gambar 12.13.	Rancangan Form Data Pasien	222
Gambar 12.14.	Rancangan Form Data Penyakit	222
Gambar 12.15.	Rancangan Form Data Gejala	223
Gambar 12.16.	Rancangan Form Data Pengetahuan.....	223
Gambar 12.17.	Rancangan Form Diganosa.....	224
Gambar 12.18.	Rancangan Form Data Hasil Diagnosa	224
Gambar 12.19.	Rancangan Laporan Data Pasien.....	225

Gambar 12.20.	Rancangan Laporan Data Penyakit.....	225
Gambar 12.21.	Rancangan Laporan Data Gejala	226
Gambar 12.22.	Rancangan Laporan Data Pengetahuan	226
Gambar 12.22.	Rancangan Laporan Data Hasil Diagnosa	227
Gambar 12.23.	Form Menu Utama.....	229
Gambar 12.24.	Form Login.....	229
Gambar 12.25.	Form Data Pasien	230
Gambar 12.26.	Form Data Penyakit	230
Gambar 12.27.	Form Data Gejala	231
Gambar 12.28.	Form Data Pengetahuan	231
Gambar 12.29.	Form Diagnosa.....	232
Gambar 12.30.	Form Hasil Diagnosa	232
Gambar 12.38.	Laporan Data Pasien.....	233
Gambar 12.39.	Laporan Data Penyakit.....	233
Gambar 12.40.	Laporan Data Gejala.....	234
Gambar 12.41.	Laporan Data Pengetahuan.....	234
Gambar 12.42.	Laporan Data Hasil Diagnosa	135
Gambar 13.1.	Arsitektur <i>Fuzzy Inference System</i>	246
Gambar 13.2.	Model FIS Tipe Sugeno Pada Aplikasi Matlab.....	247
Gambar 13.3.	Fungsi Keanggotaan <i>Input</i> Untuk Variabel <i>Plafond</i>	248

DAFTAR TABEL

Tabel 1.1. Perbandingan Pakar dengan Sistem Pakar	9
Tabel 2.1. Perbandingan Metode Struktur Basis Pengetahuan.....	16
Tabel 2.2. Perbandingan antara forward chaining dan backward chaining.....	24
Tabel 2.3. Perbandingan tipe antarmuka pengguna	26
Tabel 3.1. Perbandingan Forward dan Backward Chaining.....	48
Tabel 3.2. Contoh Penerapan Aturan IF-THEN dalam MYCIN	49
Tabel 3.3. Kelebihan dan Kekurangan Model Penalaran Klasik	50
Tabel 3.4. Contoh aturan fuzzy.....	54
Tabel 11.1. Tingkat Keyakinan Certainty Factor (CF)	184
Tabel 11.2. Nilai CF gejala terhadap Penyakit.....	184
Tabel 12.1. Data Pasien	200
Tabel 12.2. Data Gejala Penyakit	202
Tabel 12.3. Tabel Data Pasien.....	218
Tabel 12.4. Tabel Data Penyakit	218
Tabel 12.5 Tabel Data Gejala.....	219
Tabel 12.6. Tabel Data Pengetahuan.....	219
Tabel 12.7. Tabel Data Diagnosa.....	220
Tabel 12.8. Nama Form Yang Dibuat.....	227
Tabel 12.9. Nama Tabel Yang Dibuat	228
Tabel 13.1. Himpunan <i>fuzzy</i> Data Calon debitur kredit	246
Tabel 13.2. Himpunan <i>Fuzzy Plafond</i> Yang Diajukan Untuk <i>Input</i> 1	249
Tabel 13.3. <i>Output Fuzzy</i> Persentase Keputusan Kredit.....	249
Tabel 13.4. <i>Rule</i> Data Untuk Persentase Keputusan Pemberian Kredit	251

BAB 1

PENDAHULUAN

Oleh Hasanuddin Sirait

1.1 Pendahuluan

Sistem pakar (*expert system*) sering disebut dengan AI (*Artificial Intelligence*) merupakan suatu program komputer atau informasi yang memiliki kemampuan seperti yang dilakukan para pakar. Kepekaran tersebut terdiri dari beberapa bidang pengetahuan atau kemampuan manusia. Yang dimaksud dengan pakar yaitu seseorang yang memiliki keahlian atau ilmu pengetahuan pada salah bidang ilmu atau keahlian tertentu, contohnya : keahlian dokter, keahlian psikolog, keahlian mekanik, keahlian analisis, dan lainnya. Bentuk umum dari pada sistem pakar terdiri suatu program/perangkat lunak/software yang dirancang berdasarkan suatu aturan yang detail dari analisis informasi, serta sebagai suatu penalaran mengenai suatu masalah spesifik dipecahkan dalam suatu analisis matematik dari masalah tersebut. Perangkat lunak ini pertama kali dikembangkan oleh ilmuan di Universitas Standford dengan istilah Dendral oleh E. Feigenbaum tahun 1960, sistem ini dapat menentukan struktur molekul organik dan rumus kimia serta data spektrografi ikatan kimia pada molekul yang ada, kemudian dikembangkan kembali tahun 1970 dengan struktur molekul besar. Dan pada tahun 1980 dikembangkan menjadi suatu sistem pakar tujuan komersial dengan istilah AI (*Artificial Inteligen*) atau sistem cerdas (*Expert system*) (Ahyuna and Aryasa 2017).

Dengan perkembangan yang sangat pesat maka sistem pakar juga dapat memberikan rekomendasi sebagai solusi dari permasalahan untuk dapat dilakukan tindakan atau kesimpulan.

Sistem Kerja dari Sistem pakar adalah dengan memanfaatkan fungsi penalaran dalam mengambil simpulan berdasarkan data yang ada, serta melalui big data dengan menggunakan salah satu dan beberapa algoritma (Usmanto 2017).

Sistem pakar dikembangkan dari beberapa metode dan algoritma dengan meniru suatu kepakaran/keahlian oleh manusia. Penerapan sistem pakar akan secara otomatis dapat memberikan kesimpulan atau alternatif atau solusi dengan melakukan perhitungan atau logika secara cepat dari suatu analisis algoritma, baik melalui big data maupun aritmatika yang diterapkan (Muttaqin, Harmonvikler Dumoharis Lumban Raja, Adhi Prasetyo, Harizahayu Rahmadini Darwas, Teguh Priyantoro, and Fergie Joanda Kaunang, Hartanto Tantriawan, Hasanuddin Sirait Zelfi Gustiana, Darsin, Andryanto. A 2022).

Dalam buku ini akan dibahas tuntas mengenai : Komponen Utama dan Domain Sistem Pakar; Konsep Dasar Penalaran Sistem Pakar; Elemen dan Struktur Dasar Sistem Pakar; Metode Pengembangan Sistem Pakar; Metode Aktifasi & Rules Sistem Pakar; Metode Perancangan Aplikasi & Tools Sistem Pakar; Penerapan Algoritma K-Nearest Neighbour dalam Sistem Pakar; Penerapan Algoritma Markov dalam Sistem Pakar; Penerapan Algoritma Naives Bayes dalam Sistem Pakar; Penerapan Algoritma *Certainty Factor* Dalam Sistem Pakar; Penerapan Algoritma Forward Chaining Dalam Sistem Pakar; Penerapan Metode inferensi Fuzzy dalam Sistem Pakar. Sehingga buku ini dapat menjadi referensi pengetahuan baru bagi pembaca dalam membuat suatu keputusan dengan baik.

1.2 Tujuan Sistem Pakar (*Expert System Purpose*)

Dibuatnya sistem pakar agar memiliki dapat memberi rekomendasi rangkaian dari suatu tindakan (*action*) atau perilaku (*behaviour*) kepada pengguna sehingga dapat melakukan sistem koreksi atau keputusan yang tepat dan akurat (Indit Rahmawati

2016). Sistem Pakar juga memanfaatkan dan kapabilitas proses penalaran untuk mencapai simpulan berdasarkan pengolahan data dan fakta yang ada melalui suatu database yang besar dengan sistem pengolahan melalui algoritma tertentu. Beberapa kegunaan dibuatnya sistem pakar adalah sebagai berikut :

1. Melakukan interpretasi, *expert system/AI (artificial intelligence)* yang bertujuan melakukan proses peninjauan data mentah (*raw data*) yang terdiri dari angka, huruf/kalimat, suara dan gambar sampai kepada kesimpulan atau deskripsi. Sistem ini melakukan proses citra, contohnya : Aplikasi search engine, aplikasi ground water, Aplikasi Pajak P3B (Persetujuan Penghindaran Pajak Berganda), dan lainnya.
2. Melakukan Prediksi, sistem ini mampu memberikan informasi akibat dari situasi atau fenomena berdasarkan data, contohnya prediksi terkait data demografi, *finance*, ekonomi, cuaca/iklim, dan lainnya.
3. Sebagai Piagnosis, sistem dapat menentukan penyebab terjadinya kesalahan dari suatu diagnosa pada situasi yang kompleks berdasarkan gejala yang ada, sehingga dapat dilihat diagnosis yang tepat, contohnya aplikasi medis seperti : Medscape, MIMS, Prognosis, Calculator IMT, dan lainnya.
4. Sebagai Perancangan dan Disain, sistem ini dapat menentukan dan membuat rancangan konfigurasi terkait komponen-komponen sistem yang ada dan dicocokkan dengan kinerja tertentu dalam memenuhi kendala tertentu seperti :
 - a. Perancangan desain bangunan/animasi dimensi seperti : *SketchUp, SmartDraw, ArchiCad, Lumion* dan lainnya.
 - b. Penentu lapangan pekerjaan, seperti : *evernote, Laspas, Lua, Trello*, dan lainnya.

- c. Meningkatkan kualitas kerja seperti : *Google Tasks, Grammarly Keyboard, evernote, Doodle, RescueTime, oTranscribe, stay focus* dan lainnya.
5. Sebagai Perencanaan, *expert system* dapat dilakukan untuk merencanakan serangkaian tindakan yang mendapat suatu umpan balik dari rencana yang diharapkan bertujuan pada tahap kondisi dan keadaan awal yang ada, contohnya penggunaan aplikasi SINCAN (Aplikasi sistem Perencanaan), aplikasi AKSARA, aplikasi Perencanaan Keuangan seperti : *Money Lover, Money Manager, Spendee Budged* dan *Money Tracker, Blue Coin*, dan lainnya.
6. Sebagai Monitoring, sistem ini memberikan hasil pengamatan berdasarkan suatu kondisi dan keadaan yang diharapkan, contohnya *computer aided monitoring system (CAMS)*, aplikasi jaringan seperti : *Netmonk, SolarWinds, Paessler PRTG, Zabbix, ManageEngine OpManager, Nagios* dan lainnya.
7. Sebagai *Debugging*, sistem ini mampu untuk melakukan proses indentifikasi dan melakukan penghapusan jika terjadi kesalahan, sehingga dapat membantu para programmer untuk membangun suatu aplikasi seperti aplikasi GitBook untuk android, PyCharm untuk Python, Node.js dan Dinotes untuk Visual Studio.
8. Sebagai Instruksi, mempunyai kemampuan untuk menganalisa dan mendeteksi terhadap pemahaman dari suatu domain subjek, sehingga dalam memberikan instruks, contohnya dalam merancang suatu aplikasi. Sehingga instruksi dapat diketahui dalam penerapannya, dan lain sebagainya.
9. Sebagai Kontrol, Sistem ini memiliki kemampuan untuk mengatur pola dan proses dari suatu objek sesuai dengan lingkungannya. Contohnya adalah kontrol terhadap

interpretasi, perbaikan/ revarasi, dan prediksi pada sistem mekanik.

1.3 Struktur Sistem Pakar (*expert system structure*)

Struktur dari pengembangan sistem pakar, terdiri dari beberapa komonen serta tersusun dengan baik dan terstruktur dan dapat pembentuk suatu sistem informasi yang komprehensif (Usmanto 2017). Beberapa bagian dari arsitektur sistem pakar adalah sebagai berikut :

1. *User Interface (Antar muka Pengguna)*

Antar muka atau *interface* merupakan sebuah mekanisme yang digunakan sebagai sarana untuk berkomunikasi dan berinteraksi dengan pengguna (*user*). Sebagai antarmuka user akan menerima informasi dari pengguna, dan mengubah ke dalam instruksi yang dapat diterima oleh suatu sistem.

2. *Knowledge Base (berbasis Pengetahuan)*

Knowledge Base mengandung pemahaman mengenai formulasi dan skema penyelesaian masalah, sehingga alur dan skema yang ada dapat menjadi suatu pedoman dari kerja sistem.

3. *Knowledge Acquisition (Pemindahan Pengetahuan)*

Knowledge acquisition merupakan proses transformasi, akumulasi, dan transfer *knowledge* pada keahlian untuk dapat menyelesaikan permasalahan dengan membantuk suatu aplikasi komputer. Pada tahap ini, *engineer* bertugas untuk menyerap segala pengetahuandlam bentuk struktur algoritma pengolahan data yang tepat, sehingga terbentuk suatu aplikasi yang dapat memberikan informasi/kesimpulan sesuai yang dilakukan oleh seorang pakar.

4. *Inference Engine (Mesin Inferensi)*

Komponen ini memiliki mekanisme penalaran dan pola pikir yang dapat digunakan oleh para pakar untuk memecahkan suatu masalah dengan baik. Mesin inferensi dimaksud merupakan program komputer untuk dapat memberikan metodologi pada suatu pekerjaan, kemudian diolah menjadi kesimpulan.

5. *Workplace/Blackboard (Tempat Kerja/Papan tulis)*

Workplace merupakan suatu kumpulan dari pekerjaan yang digunakan, dan akan merekam setiap kejadian yang ada berupa perancangan keputusan sementara dari kejadian.

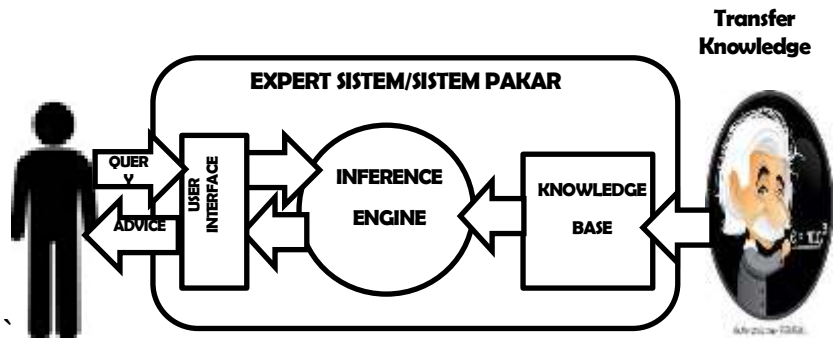
6. *Explanation/Description Facility (Deskripsi/Penjelasan)*

Deskripsi/Penjelasan merupakan komponen tambahan dari sistem pakar yang berguna meningkatkan penggunaannya, serta melacak proses dari *sistem pakar* secara interaktif dan berkelanjutan.

7. *Perbaikan Pengetahuan (Improved Knowledge)*

Perbaikan sistem pakar merupakan komponen dari suatu sistem pakar dengan kemampuan analisis yang baik untuk dapat meningkatkan kinerja sistem pakar. Kemampuan tersebut terdiri atas : keahlian dalam pembelajaran yang terkomputerisasi, *update system* pengetahuan, pengetahuan *recovery system* dan pengetahuan tentang *maintenance system*.

Sesuai dengan penjelasan diatas menunjukkan relevansi dari suatu sistem pakar untuk dapat diterapkan dalam suatu aplikasi. Berikut gambar siklus proses sistem pakar :



Gambar 1.1. Siklus Sistem Pakar (*expert system diagram*)

1.4 Manfaat Sistem Pakar (*Expert System Benefits*)

Sesuai dengan fungsinya, sistem pakar memiliki manfaat bagi pengguna untuk membantu (*assistant*) dalam penyelesaian masalah atau memberi *alternative solution* dalam memberi suatu keputusan. Berikut ini beberapa manfaat dari sistem pakar :

1. Sebagai orang awam dapat mengerjakan pekerjaan sebagaimana ahli/pakar mengerjakannya.
2. Dapat melakukan kegiatan yang serupa secara berulang dan otomatis serta tidak mengalami kebosanan atau kejenuhan.
3. Dapat menyimpan data pengetahuan (*knowledge*) dan keahlian (*skill*) para pakar kedalam suatu database menjadi suatu big data sebagai dasar (dasar) proses dari solusi.
4. Dapat mengambil suatu keputusan dan dapat melestarikan keahlian para pakar (terutama yang termasuk keahlian langka/jarang).
5. Dapat melakukan proses dalam lingkungan yang berbahaya jika dilakukan manusia secara konvensional langsung.
6. Dapat berkerja sesuai dengan instruksi secara terus menerus tanpa mengurangi kualitas dan fungsinya.

7. Memiliki kemampuan untuk bekerja berdasarkan data dan proses yang lengkap sehingga dapat menghasilkan suatu keputusan dengan tidak mengalami keragu-raguan, karena telah memiliki suatu big data dan sistem algoritma proses pembacaannya dengan cepat dan benar.
8. Pengguna dapat merespon pada jawaban dengan hasil **'tidak tahu'** atau **'tidak yakin'** atau **'Benar'** atau **'Salah'** pada satu atau lebih pertanyaan, sehingga input dapat mempengaruhi hasil/*output* dari kesimpulan.
9. Tidak memerlukan biaya jika tidak digunakan, sehingga penggunaannya dapat dioptimalkan.
10. Dapat di administrasikan sesuai kebutuhan dengan cepat dan tidak memerlukan banyak biaya.
11. Dapat memecahkan masalah lebih cepat dengan menggunakan data yang sama secara *continue*.
12. Nilai toleransi dapat ditetapkan pada semua proses, sehingga akurasi kesimpulan mendekati kesempurnaan.
13. Meningkatkan kualitas pekerjaan kepada pengguna, karena dapat memberi alternative dengan cepat serta dilakukan dengan konsisten dalam mengurangi kesalahan.
14. Meningkatkan kapabilitas pada sistem terkomputerisasi serta cepat dan dapat melakukan *shared*/membagikan kepada peralatan lain secara terintegrasi.
15. Mampu menyediakan visualisasi atau simulasi untuk menghindari kesalahan atau resiko yang terjadi.
16. Dapat digunakan dengan baik oleh para pengguna pemula, karena memiliki penjelasan/bantuan (*help*) sehingga dapat berfungsi sebagai guru.
17. Dapat direkomendasikan sebagai jejak digital untuk waktu yang tidak terhingga karena memiliki keterkaitan dengan lainnya.

18. Dapat membaca jejak digital dari sistem lain yang sudah terintegrasi yaitu dengan menggunakan metode proses penerapan algoritma tertentu.

Dari penjelasan diatas maka berikut perbandingan pakar dengan sistem pakar (*expert system*) dari berbagai faktor :

Tabel 1.1. Perbandingan Pakar dengan Sistem Pakar

Faktor	Pakar	Sistem Pakar
<i>Time availability</i>	Hari Kerja	Setiap Saat
Duplication of results	Variable	Cepat
Reference source	Terbatas	<i>integrated system</i>
Make a decision	Manusiawi	Sistem
Accuracy	Variable	Data
Geografis	Lokal/tertentu	Dimana saja
Keamanan	Tidak tergantikan	Dapat diganti
Perishable/dapat habis	Ya	Tidak
Performansi	Variable	Konsisten
Kecepatan	Variable	Konsisten
Biaya	Tinggi	Terjangkau

1.5 Kelebihan Sistem Pakar (*Advantages of Expert Systems*)

Kelebihan sistem dalam hal penyajian informasi adalah sebagai berikut (Wulandari, Novi Yohanes 2018) :

1. Meningkatkan produktivitas kerja, yang mana dapat membantu dalam menyelesaikan setiap pekerjaan dalam waktu yang lebih cepat.
2. Mampu meningkatkan kualitas dari sisi pemberian nasihat yang lebih konsisten.

3. Memiliki tingkat keandalan yang relatif tinggi, serta dapat bekerja secara *real time*

1.6 Kekurangan Sistem Pakar (*Disadvantages of Expert Systems*)

Dalam aplikasinya/penerapannya, sistem pakar tetap memiliki kekurangan antara lain :

1. Kendala dalam mendapatkan pengalaman para pakar dari pengetahuan yang baru, sebagaimana penerapan pengetahuan baru pakar sudah pasti memiliki pengalaman-pengalaman baru. Sehingga pendekatan dalam pengalaman pengetahuan baru harus menggunakan beberapa metode yang saling keterkaitan dikarenakan memiliki fenomena yang berbeda dan berubah/tidak sama dengan menggunakan variable baru.
2. Proses pembuatan pakar, memerlukan biaya yang besar karena tetap memperhatikan faktor kualitas dan pengetahuan terbaik dan baru, sehingga hasilnya memiliki tingkat kepuasan lebih sempurna.
3. Nilai kepuasan sistem pakar tidak memiliki kebenaran 100%, namun masih memerlukan tahap pengujian secara berkala untuk dapat menghasilkan yang lebih baik. Sistem Pakar hanya sebagai rekomendasi atau acuan/intusi dalam menetapkan suatu keputusan.

1.7 Penerapan Sistem Pakar

Berikut ini terdapat beberapa contoh penerapan dari sistem pakar dalam kehidupan sehari-hari :

1. Sistem pakar pada bidang analisa.
 - a. Dendral, aplikasi sistem pakar digunakan mengidentifikasi struktur molekul pada campuran kimia yang tidak dikenal.
 - b. MYCIN, Aplikasi sistempaar untuk mendiagnosa berbagai jenis penyakit.

- c. Prospector, aplikasi sistem pakar pada bidang geologi.
 - d. XCON dan XSEL, aplikasi sistem pakar mengkonfigurasi sistem komputer dengan skala besar.
 - e. CADUCEUS, aplikasi sistem pakar untuk mendeteksi penyakit.
 - f. PUFF, Aplikasi sistem pakar untuk mengukur fungsi paru-paru.
 - g. Toxic Master, dibuat oleh Universitas Bina Nusantara untuk mendeteksi kandungan racun yang diserap oleh tubuh manusia.
2. Sistem pakar bidang Industri/Manufaktur.
 - a. Pride (*punch roll interactive design expert/environmen*), aplikasi sistem pengaturan kertas untuk mesin fotocopy.
 - b. WTFA (*wood truss fabrication application*), untuk membantu sistem pakar, sering disebut *shell* Sistem Pakar SPS (*semi intelligen process selector*).
 - c. *Contionuous caster steel Miil Schedulling Apprication*, aplikasi sistem pakar menggunakan algoritma *fruzzy logic* yang dibuat untuk sistem monitoring *online* dan penjadwalan *continuous caster steel mill* pada industri baja.
 3. Sistem pakar pengontrol pada industri.
 - a. *Aluminium Foil Rolling Flatness Control Application*, sistem pakar ini dibuat untuk mengontrol wadah aluminium foil secara otomatis.
 - b. *Blass Furnace heat control application*, untuk mengontrol tingkat panas pada furnace (perapian) sehingga panas dapat terkendali.
 4. Sistem pakar bidang pendidikan dan ilmu pengetahuan. Sehingga para pakar dapat meningkatkan pengetahuan dengan berbagai pengujian dengan tidak langsung.

Saat ini aplikasi sistem pakar sangat banyak digunakan pada bidang riset dan pendidikan. Baik dalam analisa kata, suara warna, gambar dan lainnya melalui bank data seperti *search engine, multiple language translations, voice to language conversion, convert voice to text, image to sound, face recognition, voice recognition* dan lainnya. Penerapan sistem pakar pada bidang pendidikan dan ilmu pengetahuan saat ini hampir mendekat mendekati sempurna.

DAFTAR PUSTAKA

- Ahyuna, Ahyuna, and Komang Aryasa. 2017. "Sistem Pakar Diagnosa Dan Tatalaksana Penyakit Demam Berdarah Dengue Menggunakan Jaringan Saraf Tiruan Backpropagation." *E-JURNAL JUSITI: Jurnal Sistem Informasi dan Teknologi Informasi* 6(1): 1-11. <http://ejurnal.dipaneegara.ac.id/index.php/jusiti/article/view/40>.
- Indit Rahmawati. 2016. "Pemanfaatan Media Pembelajaran." *El-Wasathiya: Jurnal Studi Agama* 5(1): 1-56. <https://ejournal.poltektegal.ac.id/index.php/siklus/article/view/298%0Ahttp://repositorio.unan.edu.ni/2986/1/5624.pdf%0Ahttp://dx.doi.org/10.1016/j.jana.2015.10.005%0Ahttp://www.biomedcentral.com/1471-2458/12/58%0Ahttp://ovidsp.ovid.com/ovidweb.cgi?T=JS&P>.
- Muttaqin, Khairunnisa Samosir, Nurzaenab Harmonvikler Dumoharis Lumban Raja, Adhi Prasetyo, Harizahayu Rahmadini Darwas, Teguh Priyantoro, and Janner Simarmata Fergie Joanda Kaunang, Hartanto Tantriawan, Hasanuddin Sirait Zelvi Gustiana, Darsin, Andryanto. A. 2022. Yayasan Kita Menulis *BIG DATA: Informasi Dalam Dunia Digital*. 1st ed. ed. Yayasan Kita Menulis. Medan: Yayasan Kita Menulis. <https://kitamenulis.id/2022/04/28/big-data-informasi-dalam-dunia-digital/>.
- Usmanto, BudiTri Susilowati. 2017. "EXPERT JURNAL SISTEM INFORMASI." *FAKULTAS ILMU KOMPUTER BANDAR LAMPUNG* 7(2). <https://media.neliti.com/media/publications/346000-perancangan-prototype-teknologi-smart-bu-41bf1e71.pdf>.

Wulandari, Novi Yohanes, Siregar Armstrong. 2018. 1 Jurnal Ilmu Komputer *Sistem Pakar Untuk Mendiagnosa Dan Klasifikasi Anak Berkebutuhan Khusus Dengan Backward Dan Forward Chaining Pada Slb Cahaya Pertiwi Di Bekasi*. Jurnal Ilm. ed. Armstrong Yohanes Siregar. BEKASI, JAW BARAT: Jurnal Ilmu Komputer-JIK.

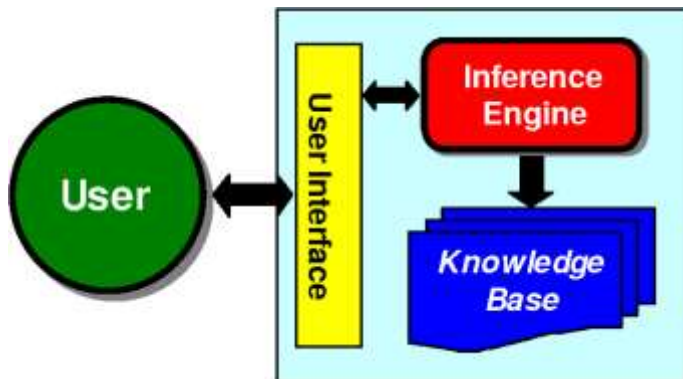
BAB 2

KOMPONEN UTAMA DAN DOMAIN SISTEM PAKAR

Oleh Rini Mayasari

2.1 Komponen Utama Sistem Pakar

Komponen utama dalam sistem pakar mencakup basis pengetahuan, mesin inferensi, dan antarmuka pengguna. Setiap komponen ini memiliki peran penting dalam pengembangan dan operasionalisasi sistem pakar.



Gambar 2.1. Komponen Utama Sistem Pakar

2.1.1 Basis Pengetahuan

Basis pengetahuan adalah komponen penting dalam sistem pakar yang menyimpan seluruh pengetahuan yang telah dikumpulkan dari pakar dalam bidang tertentu (Liao, 2005). Fungsi utama basis pengetahuan adalah sebagai sumber informasi

dan pengetahuan untuk sistem pakar dalam melakukan proses inferensi dan penyelesaian masalah.

Basis pengetahuan biasanya terdiri dari fakta dan aturan. Fakta adalah pernyataan tentang dunia yang diterima sebagai benar, sedangkan aturan adalah pedoman untuk melakukan tindakan berdasarkan fakta tertentu (Giarratano & Riley, 2004). Dalam konteks sistem pakar, aturan biasanya dinyatakan dalam bentuk "IF kondisi THEN aksi".

Struktur dan organisasi basis pengetahuan sangat penting dalam mempengaruhi efisiensi dan efektivitas sistem pakar. Ada beberapa cara dalam struktur basis pengetahuan, seperti pengetahuan berbasis aturan, pengetahuan berbasis frame, dan pengetahuan berbasis semantik. Pilihan metode ini biasanya tergantung pada jenis masalah dan pengetahuan yang digunakan dalam sistem pakar.

Tabel 2.1. Perbandingan Metode Struktur Basis Pengetahuan

Metode Struktur	Deskripsi	Keuntungan	Kekurangan
Aturan	Basis pengetahuan berbasis aturan terdiri dari himpunan aturan "IF-THEN" yang mencerminkan pengetahuan dari seorang pakar.	Mudah dipahami dan ditransfer dari pakar ke sistem, cocok untuk masalah yang strukturnya jelas.	Tidak cocok untuk masalah dengan banyak pengecualian atau struktur yang kompleks.
Frame	Pengetahuan berbasis frame menggunakan struktur data yang	Memfasilitasi organisasi dan representasi pengetahuan	Lebih sulit untuk dibangun dan dikelola dibandingkan

Metode Struktur	Deskripsi	Keuntungan	Kekurangan
	mengelompokkan objek terkait dan atribut-atributnya.	yang lebih kompleks, baik untuk situasi yang mengharuskan representasi objek dan sifatnya.	dengan model berbasis aturan.
Semantik	Pengetahuan berbasis semantik menggunakan jaringan semantik untuk merepresentasikan pengetahuan, biasanya dalam bentuk graf.	Sangat baik untuk representasi pengetahuan yang melibatkan hubungan banyak-ke-banyak, memberikan struktur yang lebih fleksibel dan ekspresif.	Kompleks dan membutuhkan waktu dan sumber daya yang signifikan untuk dibangun dan dikelola.

Pembangunan basis pengetahuan adalah proses yang membutuhkan waktu dan sumber daya yang signifikan, seringkali melibatkan proses elicitation pengetahuan dari pakar dan pengetahuan tersebut harus ditransformasikan menjadi format yang bisa dipahami oleh sistem pakar.

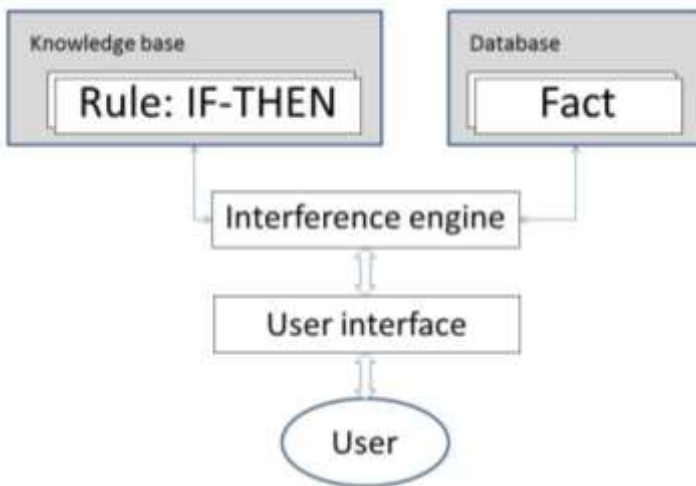
1. Struktur Basis Pengetahuan

Struktur basis pengetahuan (*Knowledge Base Structure*) merupakan bagian vital dalam sistem pakar. Basis pengetahuan menyimpan semua informasi yang digunakan

oleh sistem pakar untuk memecahkan masalah dan membuat keputusan. Struktur basis pengetahuan terdiri dari kumpulan fakta dan aturan yang diterapkan oleh sistem pakar untuk menarik kesimpulan (Giarratano & Riley, 2004).

Struktur basis pengetahuan umumnya terbagi menjadi dua komponen utama:

- a. **Fakta**, yang merupakan informasi konkret dan spesifik mengenai domain masalah. Misalnya, dalam sistem pakar medis, fakta bisa berupa gejala yang dihadapi oleh pasien.
- b. **Aturan**, yang merupakan pengetahuan heuristik, biasanya dalam bentuk "jika...maka..." (If...Then...), yang membantu sistem pakar dalam menarik kesimpulan berdasarkan fakta yang ada.



Gambar 2.2. Struktur Basis Pengetahuan

Pengetahuan dalam basis pengetahuan biasanya ditambahkan dan diperbarui oleh pakar domain. Penggunaan sistem pakar, dalam banyak kasus, akan melibatkan penambahan dan pembaruan informasi dalam basis

pengetahuan untuk menjaga relevansi dan akurasi sistem (Russell & Norvig, 2021).

Struktur basis pengetahuan dalam sistem pakar dapat bervariasi tergantung pada jenis sistem pakar, domain pengetahuan, dan metode inferensi yang digunakan. Dalam beberapa kasus, basis pengetahuan dapat mencakup pengetahuan prosedural, yaitu serangkaian langkah-langkah yang harus diikuti untuk mencapai kesimpulan tertentu.

2. Representasi Pengetahuan

Representasi pengetahuan adalah bagian penting dari sistem pakar yang berfungsi untuk menyimpan dan mengorganisir informasi agar dapat digunakan oleh mesin inferensi. Dalam konteks ini, representasi pengetahuan merujuk pada metode yang digunakan oleh sistem pakar untuk merumuskan dan menyimpan pengetahuan (Liu, 2012).

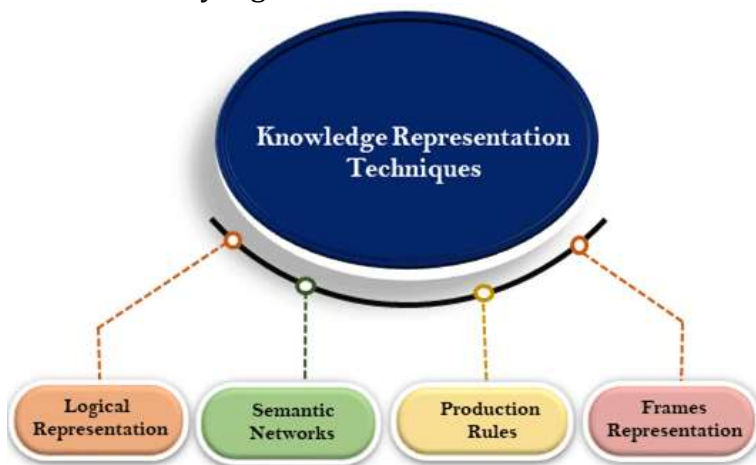
Representasi pengetahuan adalah suatu metode yang digunakan untuk merencanakan dan menyimpan pengetahuan pada sistem pakar. Tujuannya adalah untuk memodelkan dunia nyata sehingga sistem komputer dapat memahaminya. Metode representasi pengetahuan ini biasanya digunakan dalam sistem berbasis pengetahuan, kecerdasan buatan dan aplikasi terkait lainnya.

Ada beberapa teknik representasi pengetahuan yang umum digunakan, antara lain:

- a. Logika: Teknik ini menggunakan sistem formal yang didasarkan pada prinsip logika untuk mewakili pengetahuan. Teknik ini memanfaatkan pernyataan logika untuk menggambarkan informasi tentang dunia dan menggunakan inferensi logika untuk mendeduksi fakta baru dari pernyataan tersebut.
- b. Frame: Teknik ini merupakan teknik representasi pengetahuan yang berorientasi objek di mana

pengetahuan direpresentasikan sebagai set entitas dan hubungan antara entitas-entitas tersebut. Frame adalah kumpulan atribut dan nilai yang mendeskripsikan suatu objek atau konsep.

- c. Jaringan Semantik: Teknik ini mewakili pengetahuan dalam bentuk graf. Dalam jaringan semantik, konsep direpresentasikan sebagai node dan hubungan antara konsep direpresentasikan sebagai tepi.
- d. Aturan Produksi: Teknik ini menggunakan aturan "IF-THEN" untuk mewakili pengetahuan. Aturan ini digunakan untuk membuat keputusan berdasarkan informasi yang tersedia.



Gambar 2.3. Representasi Pengetahuan

Untuk menggunakan representasi pengetahuan ini secara efektif dalam sistem pakar, ada beberapa hal yang perlu dipertimbangkan:

- a. Pilihan Teknik Representasi Pengetahuan: Teknik yang dipilih tergantung pada jenis masalah yang ingin diselesaikan. Beberapa masalah mungkin lebih mudah

diwakili menggunakan logika atau aturan, sedangkan masalah lainnya mungkin lebih cocok dengan jaringan semantik atau frame. Dalam banyak kasus, sistem pakar akan menggunakan lebih dari satu teknik representasi pengetahuan.

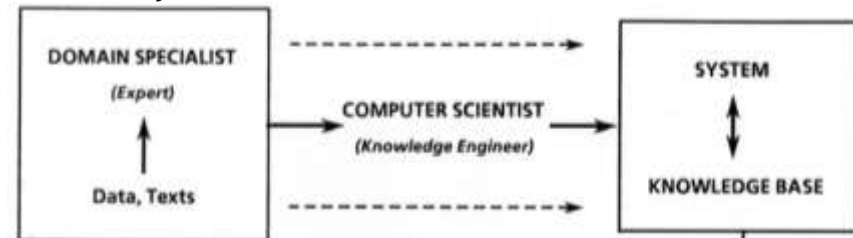
- b. Penyimpanan dan Pengambilan Pengetahuan: Dalam sistem pakar, pengetahuan harus disimpan dalam cara yang memungkinkan untuk pencarian yang efisien dan akurat. Teknik penyimpanan akan sangat bergantung pada teknik representasi pengetahuan yang dipilih.
- c. Pembaruan Pengetahuan: Sistem pakar harus mampu memperbarui dan memodifikasi basis pengetahuannya. Ini mungkin termasuk penambahan pengetahuan baru, modifikasi pengetahuan yang ada, atau penghapusan pengetahuan yang tidak lagi relevan atau akurat.
- d. Keterbatasan dan Tantangan: Setiap teknik representasi pengetahuan memiliki keterbatasan dan tantangan sendiri. Misalnya, logika mungkin sulit untuk mewakili pengetahuan yang tidak pasti atau ambigu, sedangkan jaringan semantik dan frame mungkin menjadi rumit dan sulit untuk dikelola saat jumlah konsep dan hubungan meningkat.

3. Acquisition dan Maintenance Pengetahuan

Proses acquisition dan maintenance pengetahuan merupakan bagian krusial dalam sistem pakar. Acquisition pengetahuan merujuk pada proses pengumpulan dan mengekstrakan pengetahuan dari pakar dalam bidang tertentu, sementara maintenance pengetahuan berkaitan dengan proses update, validasi, dan peningkatan kualitas pengetahuan yang telah diakuisisi (Durkin, 1994).

Akuisisi Pengetahuan

Pada dasarnya, akuisisi pengetahuan adalah proses mengambil pengetahuan dari pakar dan mentransfernya ke dalam sistem pakar. Proses ini seringkali melibatkan pakar, pengguna, dan engineer sistem pakar dalam dialog interaktif (Aamodt & Plaza, 1994).



Gambar 2.4. Proses Akuisisi Pengetahuan

Teknik-teknik dalam proses akuisisi pengetahuan, termasuk wawancara, observasi, dan analisis dokumen, biasanya digunakan untuk mengumpulkan pengetahuan dari pakar. Pengetahuan yang diakuisisi kemudian direpresentasikan dalam bentuk yang dapat dimengerti oleh sistem pakar, seperti rules, frame, atau semantic networks (Giarratano & Riley, 2004).

Maintenance Pengetahuan

Maintenance pengetahuan, di sisi lain, adalah proses yang melibatkan peninjauan dan pembaruan pengetahuan yang ada dalam sistem pakar. Ini dilakukan untuk memastikan bahwa pengetahuan dalam sistem selalu up-to-date dan akurat (Marakas, 2003). Proses maintenance juga membantu dalam deteksi dan koreksi kesalahan atau ketidaksesuaian dalam basis pengetahuan.

Sejalan dengan perkembangan dan perubahan di dunia nyata, pengetahuan dalam sistem pakar harus diupdate dan diperbarui secara berkala. Proses ini melibatkan evaluasi dan

validasi pengetahuan, yang bisa dilakukan melalui pengujian sistem, feedback dari pengguna, dan konsultasi dengan pakar.

2.1.2 Mesin Inferensi

Mesin inferensi, juga dikenal sebagai 'engine' sistem pakar, merupakan komponen penting yang bertanggung jawab atas proses penalaran dan penarikan kesimpulan dari basis pengetahuan yang ada dalam sistem (Durkin, 1994). Mesin inferensi adalah bagian dari sistem yang memungkinkan penyelesaian masalah dengan cara mengaplikasikan logika pada pengetahuan yang tersedia. Dengan kata lain, mesin inferensi adalah komponen yang "memahami" dan menerapkan aturan pada data masukan untuk mencapai kesimpulan atau solusi.

1. Jenis-jenis Mesin Inferensi

Mesin inferensi umumnya dapat dibagi menjadi dua tipe utama: forward chaining dan backward chaining (Giarratano and Riley, 2004).

- a. Forward chaining: Dalam proses ini, mesin inferensi mulai dari data atau fakta yang diketahui dan berusaha menuju ke kesimpulan. Ini sering digunakan dalam situasi di mana kita ingin "mengenali" sesuatu atau mengetahui apa yang bisa kita simpulkan dari data yang kita miliki.
- b. Backward chaining: Ini adalah pendekatan yang dimulai dari tujuan atau hipotesis dan bergerak mundur untuk mencari fakta atau data yang mendukung atau membuktikan hipotesis tersebut. Ini biasanya digunakan ketika kita memiliki kesimpulan atau tujuan tertentu yang ingin dicapai dan perlu mencari bukti untuk mendukungnya.

Tabel 2.2. Perbandingan antara forward chaining dan backward chaining

Forward Chaining	Backward Chaining
Mulai dari fakta yang diketahui	Mulai dari hipotesis atau tujuan
Mencari kesimpulan	Mencari bukti

2. Mekanisme Inferensi

Mekanisme inferensi adalah metode yang digunakan mesin inferensi untuk menerapkan aturan-aturan yang ada dalam basis pengetahuan. Mekanisme ini mencakup proses pengambilan keputusan tentang aturan mana yang harus diterapkan, kapan harus diterapkan, dan bagaimana hasil dari aturan tersebut mempengaruhi penalaran selanjutnya (Russell and Norvig, 2009).

Biasanya, mesin inferensi memiliki suatu mekanisme untuk mengatasi konflik antara aturan. Misalnya, jika ada dua aturan yang kontradiktif, mesin inferensi akan memiliki cara untuk menentukan aturan mana yang harus diterapkan berdasarkan kriteria tertentu, seperti tingkat kepercayaan atau prioritas aturan.

3. Proses Penalaran

Proses penalaran melibatkan penggunaan aturan atau pengetahuan untuk mencapai kesimpulan. Dalam konteks mesin inferensi, proses penalaran mencakup identifikasi fakta atau data yang relevan, aplikasi aturan, penilaian hasil, dan iterasi proses ini sampai ditemukan solusi atau semua aturan telah diaplikasikan (Luger and Stubblefield, 2004). Dalam konteks ini, proses penalaran adalah jantung dari sistem pakar, yang memungkinkan sistem untuk meniru kemampuan ahli manusia dalam memecahkan masalah yang kompleks.

Perlu dicatat bahwa metode penalaran yang digunakan dapat bervariasi tergantung pada jenis sistem pakar dan karakteristik masalah yang ditangani. Proses penalaran dalam

sistem pakar merupakan topik penelitian yang sedang berkembang, dan penelitian lanjutan diperlukan untuk mengembangkan metode baru dan lebih efisien.

2.1.3 Antarmuka Pengguna

Antarmuka pengguna (User Interface, UI) adalah bagian dari sistem pakar yang berfungsi sebagai perantara antara pengguna dan sistem itu sendiri. Antarmuka pengguna memungkinkan pengguna untuk berinteraksi dengan sistem, memasukkan data atau pertanyaan, dan menerima hasil dari sistem dalam bentuk yang dapat dimengerti. Antarmuka pengguna yang baik sangat penting untuk memastikan sistem pakar mudah digunakan dan memberikan informasi dengan jelas dan efektif (Luger & Stubblefield, 2004).

1. Tipe-tipe Antarmuka Pengguna

Ada beberapa tipe antarmuka pengguna yang biasanya digunakan dalam sistem pakar, antara lain:

- a. Antarmuka Berbasis Teks: Pengguna memasukkan teks dan menerima respons dalam bentuk teks. Tipe ini biasanya digunakan dalam sistem pakar awal dan masih digunakan untuk aplikasi tertentu yang tidak memerlukan interaksi grafis.
- b. Antarmuka Grafis (Graphical User Interface, GUI): Pengguna berinteraksi dengan sistem melalui elemen visual seperti ikon, jendela, dan tombol. GUI membuat sistem lebih intuitif dan mudah digunakan, khususnya untuk pengguna non-teknis.
- c. Antarmuka Berbasis Suara: Pengguna berinteraksi dengan sistem melalui suara, biasanya melalui perintah suara atau percakapan. Tipe ini digunakan dalam aplikasi seperti asisten digital dan sistem telepon otomatis.

- d. Antarmuka Berbasis Sentuhan: Pengguna berinteraksi dengan sistem melalui sentuhan, biasanya pada layar sentuh. Ini sering digunakan dalam perangkat mobile seperti smartphone dan tablet.

Tabel 2.3. Perbandingan tipe antarmuka pengguna

Tipe Antarmuka Pengguna	Deskripsi	Contoh Penggunaan
Antarmuka Berbasis Teks	Pengguna memasukkan teks dan menerima respons dalam bentuk teks	Konsultasi medis melalui input teks
Antarmuka Grafis (GUI)	Pengguna berinteraksi melalui elemen visual seperti ikon dan jendela	Sistem pakar untuk diagnostik penyakit
Antarmuka Berbasis Suara	Pengguna berinteraksi dengan sistem melalui perintah suara atau percakapan	Asisten virtual seperti Siri atau Alexa
Antarmuka Berbasis Sentuhan	Pengguna berinteraksi dengan sentuhan pada layar sentuh	Aplikasi pada perangkat mobile seperti smartphone atau tablet

2. Desain Antarmuka Pengguna yang Efektif

Desain antarmuka pengguna yang efektif penting untuk memastikan sistem pakar mudah digunakan dan memberikan informasi dengan jelas. Ada beberapa prinsip yang biasanya

digunakan dalam mendesain antarmuka pengguna yang efektif, termasuk:

- a. Konsistensi: Elemen antarmuka harus konsisten dalam penggunaan warna, font, layout, dan terminologi.
- b. Simplicity: Antarmuka harus dirancang se-simple mungkin, tanpa mengorbankan fungsi penting.
- c. Feedback: Sistem harus memberikan feedback yang jelas dan tepat waktu kepada pengguna.
- d. Toleransi Kesalahan: Desain antarmuka harus mencegah kesalahan sebanyak mungkin dan memungkinkan pengguna untuk memulihkan diri dari kesalahan dengan mudah.
- e. Dokumentasi: Dokumentasi yang baik dan bantuan online harus tersedia untuk membantu pengguna memahami cara menggunakan sistem.

2.3 Domain Sistem Pakar

2.3.1 Pengertian Domain dalam Sistem Pakar

Domain dalam konteks Sistem Pakar merujuk pada area pengetahuan yang spesifik atau bidang masalah yang diambil sebagai fokus dalam pengembangan sistem pakar. Setiap sistem pakar memiliki domain yang berbeda-beda, tergantung pada tujuan dan kebutuhan aplikasi yang ingin diatasi. Domain ini mencakup pengetahuan eksplisit yang dapat digunakan oleh sistem pakar untuk memberikan solusi, rekomendasi, atau penjelasan dalam bidang yang ditentukan.

Pemilihan domain yang tepat sangat penting dalam pengembangan sistem pakar. Sebagai pengembang, Anda harus memilih domain yang memiliki kompleksitas dan kekhususan yang memungkinkan pengetahuan spesialis dalam domain tersebut dapat diimplementasikan secara efektif ke dalam sistem pakar.

Misalnya, dalam pengembangan sistem pakar untuk diagnosis penyakit pada tanaman, domainnya adalah bidang

pertanian dan hortikultura. Dalam domain ini, pengetahuan spesialis mengenai jenis-jenis penyakit, gejala-gejalanya, faktor-faktor penyebab, serta metode pengendalian harus diidentifikasi dan diwujudkan dalam basis pengetahuan sistem pakar.

Untuk mendapatkan pemahaman yang mendalam tentang domain yang dipilih, pengembang sistem pakar dapat melakukan analisis mendalam melalui studi pustaka, wawancara dengan ahli domain, atau observasi langsung. Hasil dari analisis ini akan membantu mengidentifikasi pengetahuan yang relevan, aturan-aturan penalaran, serta variabel-variabel yang harus diperhatikan dalam pengembangan sistem pakar.

2.3.2 Pemilihan dan Analisis Domain

Pemilihan dan analisis domain yang tepat merupakan langkah awal yang penting dalam membangun sistem pakar yang efektif dan berkualitas.

Pemilihan domain harus didasarkan pada kebutuhan dan tujuan sistem pakar yang akan dikembangkan. Pertama-tama, identifikasi masalah atau permasalahan yang ingin diselesaikan oleh sistem pakar. Setelah itu, lakukan evaluasi terhadap domain yang relevan dengan permasalahan tersebut. Beberapa faktor yang perlu dipertimbangkan dalam pemilihan domain antara lain ketersediaan data dan pengetahuan yang cukup, kompleksitas domain, dan dampak sosial dan ekonomi dari penerapan sistem pakar.

Setelah pemilihan domain, langkah selanjutnya adalah menganalisis domain secara mendalam. Analisis domain bertujuan untuk memahami karakteristik, aturan, dan pengetahuan yang terkait dengan domain tersebut. Beberapa metode yang dapat digunakan dalam analisis domain antara lain wawancara dengan pakar domain, pengamatan langsung, studi literatur, dan analisis data historis.

Dalam melakukan analisis domain, penting untuk mengidentifikasi dan mendokumentasikan pengetahuan eksplisit yang dimiliki oleh pakar domain. Pengetahuan eksplisit adalah pengetahuan yang dapat diekspresikan dalam bentuk aturan atau fakta yang jelas. Selain itu, juga perlu diperhatikan pengetahuan implisit yang dimiliki oleh pakar domain. Pengetahuan implisit adalah pengetahuan yang tidak terlalu mudah diungkapkan secara langsung, tetapi dapat diambil dari pengalaman dan intuisi pakar.

1. Faktor-faktor dalam Pemilihan Domain

Pemilihan domain yang tepat merupakan langkah penting dalam pengembangan sistem pakar. Terdapat beberapa faktor yang harus dipertimbangkan dalam pemilihan domain untuk memastikan kesuksesan implementasi sistem pakar.

- a. Relevansi: Salah satu faktor utama dalam pemilihan domain adalah relevansi terhadap masalah yang ingin diselesaikan. Domain yang dipilih harus berkaitan langsung dengan permasalahan yang akan diatasi oleh sistem pakar. Hal ini penting agar sistem pakar dapat memberikan solusi yang akurat dan bermanfaat.
- b. Ketersediaan Pengetahuan: Pemilihan domain juga harus mempertimbangkan ketersediaan pengetahuan yang relevan. Dalam beberapa kasus, pengetahuan yang diperlukan untuk membangun sistem pakar mungkin sulit didapatkan atau terbatas. Dalam hal ini, perlu dilakukan evaluasi terhadap ketersediaan pengetahuan dan apakah dapat diakses dengan mudah.
- c. Kompleksitas Domain: Kompleksitas domain juga merupakan faktor yang harus diperhatikan. Pemilihan domain yang terlalu kompleks dapat mengakibatkan kesulitan dalam menggambarkan pengetahuan secara formal dan mengimplementasikannya dalam sistem pakar. Sebaliknya, pemilihan domain yang terlalu

sederhana mungkin tidak memberikan tantangan yang cukup bagi sistem pakar.

- d. Keterbatasan Pengetahuan Manusia: Faktor lain yang perlu dipertimbangkan adalah keterbatasan pengetahuan manusia. Sistem pakar harus mampu merepresentasikan pengetahuan yang dimiliki oleh para ahli dan menggantikan keahlian mereka dalam memecahkan masalah. Oleh karena itu, pemilihan domain harus mempertimbangkan sejauh mana pengetahuan manusia dapat didekati dan digantikan.

Pemilihan domain yang tepat sangat penting dalam pengembangan sistem pakar untuk memastikan keberhasilan implementasi. Relevansi dengan permasalahan yang ingin diselesaikan, ketersediaan pengetahuan yang relevan, kompleksitas domain, dan keterbatasan pengetahuan manusia adalah beberapa faktor yang harus dipertimbangkan. Dengan memperhatikan faktor-faktor ini, dapat dihasilkan sistem pakar yang efektif dan mampu memberikan solusi yang akurat dan bermanfaat.

2. Metode Analisis Domain

Metode Analisis Domain adalah pendekatan sistematis yang digunakan untuk memahami dan menganalisis karakteristik dan kebutuhan domain yang akan diimplementasikan dalam sistem pakar. Tujuan dari metode ini adalah untuk mendapatkan pemahaman yang mendalam tentang domain tersebut sehingga dapat merancang dan mengembangkan sistem pakar yang efektif dan relevan.

Ada beberapa metode yang dapat digunakan dalam analisis domain, antara lain:

- a. Wawancara dengan Ahli Domain: Metode ini melibatkan interaksi langsung dengan ahli domain

yang memiliki pengetahuan dan pengalaman yang mendalam dalam bidang yang akan dijadikan domain sistem pakar. Wawancara ini bertujuan untuk memperoleh pemahaman yang lebih baik tentang domain, masalah yang umum terjadi, dan proses penyelesaiannya. Dalam wawancara, penting untuk mengajukan pertanyaan yang relevan dan mendalam untuk mendapatkan informasi yang akurat dan bermanfaat.

- b. Observasi Lapangan: Metode ini melibatkan pengamatan langsung terhadap kegiatan dan proses yang terjadi di lapangan. Dengan mengamati interaksi antara ahli domain dan objek domain, dapat diperoleh pemahaman yang lebih baik tentang konteks domain, aturan penalaran yang digunakan, serta faktor-faktor yang memengaruhi pengambilan keputusan. Observasi ini dapat dilakukan secara langsung atau melalui rekaman video atau audio.
- c. Studi Literatur: Metode ini melibatkan penelitian dan analisis literatur yang relevan dengan domain yang akan diimplementasikan. Literatur dapat berupa buku, jurnal ilmiah, artikel, dan dokumen teknis lainnya. Dengan mempelajari literatur terkait, dapat diperoleh wawasan tentang teori, konsep, metode, dan praktik terbaru dalam domain yang bersangkutan. Hal ini membantu dalam memahami basis pengetahuan yang relevan yang harus dimasukkan ke dalam sistem pakar.
- d. Analisis Data Historis: Jika data historis tersedia, metode ini dapat digunakan untuk menganalisis data yang ada dan menemukan pola atau tren yang berguna dalam domain. Analisis ini dapat melibatkan teknik statistik, seperti analisis regresi atau analisis korelasi, untuk mengidentifikasi hubungan antara variabel-

variabel dalam domain. Informasi ini dapat digunakan untuk mengembangkan aturan-aturan penalaran yang akan digunakan dalam sistem pakar.

Selama proses analisis domain, penting untuk mencatat dan menganalisis temuan-temuan yang relevan. Hal ini membantu dalam merumuskan kebutuhan sistem pakar dan merancang komponen-komponen yang sesuai dengan domain yang akan diimplementasikan.

2.3.3 Contoh-Contoh Domain Sistem Pakar

Contoh-contoh domain sistem pakar mencakup berbagai bidang yang telah memanfaatkan kecerdasan buatan untuk memecahkan masalah yang kompleks. Dalam berbagai industri, sistem pakar telah digunakan untuk memberikan solusi yang akurat dan mendalam dalam domain yang spesifik. Berikut ini adalah beberapa contoh domain sistem pakar yang telah sukses diterapkan:

1. **Diagnosa Medis:** Salah satu contoh penerapan sistem pakar yang populer adalah dalam bidang diagnosa medis. Sistem pakar di bidang ini dapat membantu dokter dalam mengidentifikasi penyakit berdasarkan gejala yang dialami oleh pasien. Sistem pakar akan menganalisis gejala yang dilaporkan oleh pasien dan memberikan diagnosis yang akurat berdasarkan pengetahuan medis yang tersimpan dalam basis pengetahuan. Dalam hal ini, sistem pakar dapat mempercepat proses diagnosis dan membantu dokter dalam membuat keputusan yang tepat mengenai perawatan yang diperlukan. Contoh sistem pakar yang terkenal di bidang ini adalah MYCIN, yang digunakan untuk diagnosa penyakit infeksi.
2. **Perencanaan Keuangan:** Sistem pakar juga telah diterapkan dalam bidang perencanaan keuangan untuk membantu

individu atau perusahaan dalam mengelola keuangan mereka dengan lebih baik. Sistem pakar dapat memberikan saran dan rekomendasi mengenai investasi, pengelolaan hutang, perencanaan pensiun, dan pengaturan keuangan lainnya berdasarkan informasi yang diberikan oleh pengguna. Sistem pakar ini akan menganalisis data keuangan dan memberikan solusi yang disesuaikan dengan tujuan dan kebutuhan individu atau perusahaan. Hal ini dapat membantu mengoptimalkan keputusan keuangan dan mengurangi risiko kerugian. Sebagai contoh, sistem pakar dapat memberikan rekomendasi tentang alokasi portofolio investasi berdasarkan profil risiko dan tujuan keuangan seseorang.

3. Manajemen Sumber Daya Manusia: Sistem pakar juga digunakan dalam bidang manajemen sumber daya manusia (SDM) untuk membantu dalam pengambilan keputusan terkait dengan manajemen tenaga kerja. Sistem pakar dalam SDM dapat membantu dalam seleksi karyawan, penilaian kinerja, pengembangan karir, dan pengelolaan konflik. Sistem pakar akan menggabungkan informasi tentang karyawan, kebijakan perusahaan, dan pengetahuan manajemen SDM untuk memberikan rekomendasi yang akurat dan objektif. Misalnya, sistem pakar dapat membantu dalam menilai kinerja karyawan berdasarkan kriteria yang telah ditentukan dan memberikan saran untuk pengembangan karir berdasarkan hasil evaluasi tersebut.

Contoh-contoh di atas hanya sebagian kecil dari berbagai domain yang telah memanfaatkan sistem pakar. Dalam masing-masing contoh tersebut, sistem pakar dapat memberikan keuntungan seperti akurasi, kecepatan, dan pengetahuan yang mendalam dalam pengambilan keputusan.

Melalui penerapan teknologi ini, berbagai bidang telah dapat memperoleh solusi yang lebih efisien dan efektif.

2.3.4 Manfaat dan Tantangan dalam Penentuan Domain

Penentuan domain dalam sistem pakar merupakan langkah penting dalam pengembangan sistem pakar. Domain merujuk pada area spesifik atau bidang pengetahuan di mana sistem pakar akan diterapkan. Pemilihan domain yang tepat akan memberikan manfaat yang signifikan, sementara tantangan yang mungkin muncul dalam proses penentuan domain harus diatasi untuk mencapai kesuksesan implementasi sistem pakar.

Ada beberapa manfaat yang dapat diperoleh dari penentuan domain yang baik dalam sistem pakar. Pertama, penentuan domain yang tepat akan memastikan relevansi dan aplikabilitas sistem pakar. Dengan memilih domain yang sesuai, sistem pakar dapat memberikan solusi yang spesifik dan relevan terhadap masalah atau pertanyaan yang diajukan. Misalnya, jika sistem pakar dikembangkan untuk diagnosis penyakit kulit, penentuan domain yang tepat akan memungkinkan sistem pakar memberikan diagnosis yang akurat dan rekomendasi perawatan yang sesuai.

Selain itu, penentuan domain yang tepat juga dapat meningkatkan efisiensi dan kecepatan sistem pakar. Dengan membatasi cakupan domain, sistem pakar dapat fokus pada pengetahuan yang relevan dan menghindari kompleksitas yang tidak perlu. Hal ini dapat menghasilkan proses penalaran yang lebih cepat dan akurat.

Namun, penentuan domain dalam sistem pakar juga dapat menghadapi beberapa tantangan. Salah satu tantangan utama adalah keterbatasan sumber daya pengetahuan. Untuk membangun sistem pakar yang efektif, diperlukan pengetahuan yang memadai dalam domain yang ditentukan. Namun, tidak selalu mudah untuk mengumpulkan pengetahuan yang komprehensif

dan terstruktur dalam domain yang spesifik. Keterbatasan pengetahuan dapat menghambat kemampuan sistem pakar untuk memberikan solusi yang akurat dan terpercaya.

Selain itu, perubahan dalam domain juga dapat menjadi tantangan. Teknologi dan pengetahuan dapat berkembang dengan cepat, dan domain yang relevan dengan sistem pakar dapat mengalami perubahan seiring waktu. Oleh karena itu, perlu adanya mekanisme untuk memperbarui dan mengelola pengetahuan dalam sistem pakar agar tetap sesuai dengan perubahan domain.

Dalam penentuan domain, penting untuk melakukan analisis yang teliti dan mempertimbangkan berbagai faktor yang relevan. Beberapa faktor yang perlu dipertimbangkan meliputi tujuan sistem pakar, ketersediaan pengetahuan, kebutuhan pengguna, dan potensi aplikasi sistem pakar dalam domain tertentu. Dengan melakukan penilaian yang komprehensif terhadap manfaat dan tantangan dalam penentuan domain, pengembang sistem pakar dapat mengarahkan upaya mereka ke arah yang tepat dan mencapai kesuksesan implementasi.

DAFTAR PUSTAKA

- Aamodt, A. and Plaza, E. 1994. 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *AI Communications*, 7(1), pp. 39-59.
- Bramer, M. 2007. *Principles of data mining*. Springer Science & Business Media.
- Durkin, J. 1994. *Expert Systems: Design and Development*. New York: Macmillan Publishing Company.
- Giarratano, J., & Riley, G. 2004. *Expert Systems: Principles and Programming*. Thomson Course Technology.
- Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. 1983. *Building expert systems*. Addison-Wesley Longman Publishing Co., Inc..
- Liu, R., Rong, Y., & Peng, Z. 2020. A review of medical artificial intelligence. *Global Health Journal*, 4(2), 42-45.
- Liao, S. H. 2005. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert Systems with Applications*, 28(1), 93-103.
- Liu, X. and Li, Y., 2012. *Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies*.
- Luger, G. F., and Stubblefield, W. A. 2004. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (5th Edition). Addison-Wesley.
- Marakas, G. M. 2003. *Decision Support Systems in the Twenty-First Century*. 2nd edn. Upper Saddle River, NJ: Prentice Hall.
- Negnevitsky, M. 2005. *Artificial Intelligence: A Guide to Intelligent Systems* (2nd ed.). Pearson Education.
- Park, H., Sim, M. K., & Choi, D. G. 2020. An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications*, 158, 113573.
- Rich, E., & Knight, K. 1991. *Artificial Intelligence*. McGraw-Hill.

- Russell, S., & Norvig, P. 2021. Artificial Intelligence: A Modern Approach. Pearson.
- Russell, S., and Norvig, P. 2009. Artificial Intelligence: A Modern Approach (3rd Edition). Prentice Hall.
- Shneiderman, B., & Plaisant, C. 2005. Designing the user interface: strategies for effective human-computer interaction. Pearson.
- Zhang, D., & Zhou, L. 2004. Discovering golden nuggets: data mining in financial application. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 34(4), 513-522.

BAB 3

KONSEP DASAR PENALARAN SISTEM PAKAR

Oleh Nono Heryana

3.1 Pengertian

3.1.1 Definisi Penalaran

Penalaran dapat diartikan sebagai proses di mana suatu kesimpulan atau keputusan dicapai melalui logika berdasarkan set informasi yang ada (Hurley, 2018). Penalaran sangat penting dalam sistem pakar karena sistem ini menggunakan penalaran untuk meniru kemampuan manusia dalam memecahkan masalah atau membuat keputusan. Penalaran ini biasanya dilakukan melalui serangkaian aturan atau pengetahuan yang telah diprogram sebelumnya dalam sistem.

Penalaran dalam konteks ini merujuk pada proses di mana sistem pakar membuat keputusan atau mencapai kesimpulan berdasarkan informasi dan aturan yang ada dalam basis pengetahuannya.

3.1.2 Perbedaan Penalaran Sistem Pakar dengan Kecerdasan Buatan

Sistem pakar dan kecerdasan buatan (AI) keduanya adalah bidang dalam ilmu komputer yang berfokus pada pembuatan sistem yang mampu melakukan tugas-tugas yang biasanya membutuhkan intervensi manusia. Meskipun keduanya bertujuan untuk membuat sistem yang bisa 'berpikir' seperti manusia, ada beberapa perbedaan penting antara cara sistem pakar dan AI melakukan penalaran.

Sistem pakar menggunakan basis pengetahuan khusus dan aturan yang telah ditentukan untuk menarik kesimpulan atau membuat keputusan. Penalaran dalam sistem pakar biasanya bersifat deterministik dan berfokus pada area pengetahuan yang sangat spesifik (Giarratano and Riley, 2004).

Sebaliknya, kecerdasan buatan menerapkan algoritma dan model matematis untuk belajar dari data dan membuat prediksi atau keputusan. AI tidak selalu memerlukan aturan yang telah ditentukan dan dapat beradaptasi dengan situasi baru melalui proses pembelajaran (Russell and Norvig, 2021). Contoh umum penalaran dalam AI adalah pembelajaran mesin dan pembelajaran mendalam (*deep learning*).

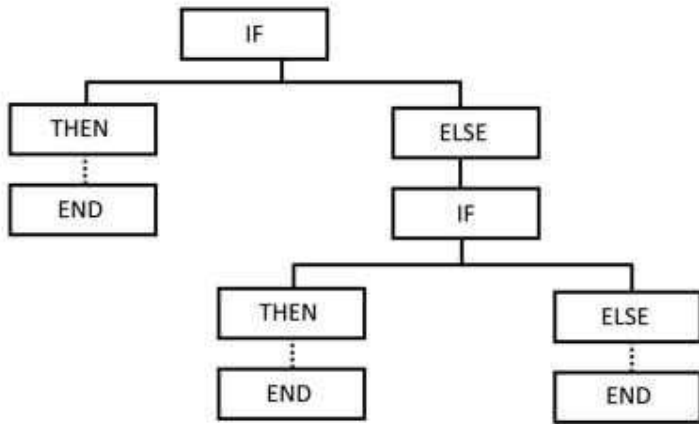
3.2 Jenis-jenis Penalaran dalam Sistem Pakar

Sistem pakar menghasilkan penalaran atau solusi berdasarkan sejumlah metode. Beberapa jenis penalaran yang umum digunakan dalam sistem pakar meliputi penalaran berbasis aturan, penalaran berbasis kasus, dan penalaran berbasis objek.

3.2.1 Penalaran Berbasis Aturan (*Rule-Based Reasoning*)

Penalaran berbasis aturan atau Rule-Based Reasoning adalah metode penalaran yang menggunakan seperangkat aturan atau pola yang telah ditentukan sebelumnya untuk membuat kesimpulan atau prediksi. Dalam konteks sistem pakar, aturan ini biasanya ditentukan oleh pakar di bidang tertentu dan disimpan dalam basis pengetahuan.

Aturan biasanya ditulis dalam bentuk "IF...THEN", dengan "IF" merujuk kepada kondisi dan "THEN" merujuk kepada tindakan yang harus diambil atau kesimpulan yang harus dibuat jika kondisi tersebut terpenuhi. Misalnya, "JIKI suhu ruangan lebih dari 30 derajat MAKA nyalakan AC".

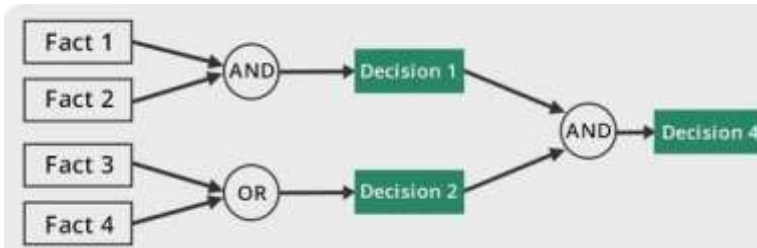


Gambar 3.1. Format Aturan IF-THEN dalam Sistem Pakar

Dua metode utama dalam penalaran berbasis aturan adalah forward chaining dan backward chaining.

1. *Forward Chaining*

Forward chaining adalah salah satu metode penalaran dalam sistem pakar berbasis aturan, di mana proses penalaran dimulai dari set fakta yang diketahui dan mencoba mencapai suatu kesimpulan atau tujuan. Konsep ini sering digunakan dalam sistem berbasis pengetahuan yang menangani penyelesaian masalah dengan pengetahuan yang spesifik (Russell & Norvig, 2021). Metode ini seringkali dipilih ketika sistem pakar menghadapi lingkungan dengan jumlah tujuan yang banyak dan tidak jelas, serta ketika sistem diharapkan untuk memberikan penalaran yang dapat ditelusuri oleh pengguna.



Gambar 3.2. Forward Chaining

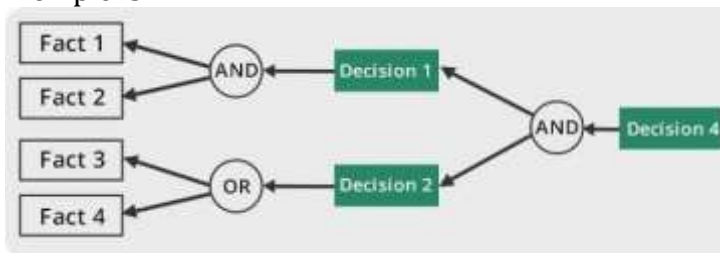
Forward chaining bekerja dengan cara memeriksa aturan yang berlaku pada fakta yang dikenal. Setelah memeriksa dan menemukan aturan yang cocok, sistem pakar kemudian melakukan tindakan yang ditentukan oleh aturan tersebut, yang biasanya melibatkan penambahan fakta baru ke basis pengetahuan sistem. Proses ini berlanjut sampai tidak ada aturan lagi yang cocok atau sampai sistem mencapai kesimpulan yang diinginkan.

Sebagai contoh, mari kita pertimbangkan sistem pakar yang digunakan untuk mendiagnosis penyakit berdasarkan gejala. Sistem ini mungkin memiliki aturan seperti "jika pasien memiliki demam tinggi dan batuk kering, maka pasien mungkin memiliki penyakit A." Dalam kasus ini, forward chaining akan memulai dengan fakta yang diketahui (misalnya, pasien memiliki demam tinggi dan batuk kering), menemukan aturan yang cocok (aturan tentang penyakit A), dan kemudian menambahkan fakta baru ke basis pengetahuan (yaitu, pasien mungkin memiliki penyakit A).

Meskipun forward chaining adalah teknik yang sangat efektif dalam berbagai aplikasi, metode ini juga memiliki beberapa keterbatasan. Salah satunya adalah bahwa metode ini bisa menjadi cukup kompleks dan memakan waktu jika terdapat banyak aturan atau jika aturan-aturan tersebut saling berinteraksi dengan cara yang kompleks (Jackson, 1999).

2. Backward Chaining

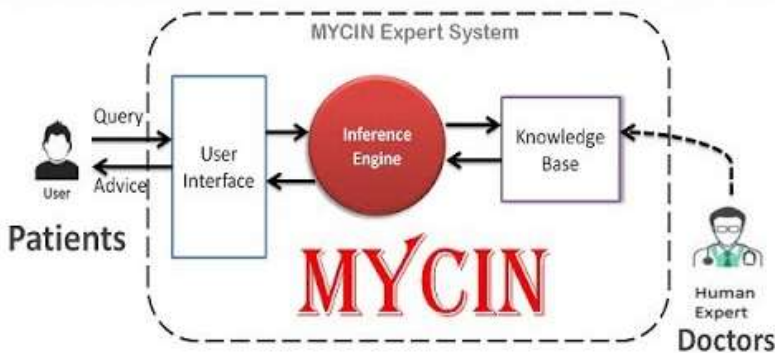
Backward chaining adalah metode penalaran lain dalam sistem pakar berbasis aturan. Dalam metode ini, sistem mulai dari suatu tujuan atau hipotesis dan kemudian mencari aturan yang dapat membuktikan atau mendukung tujuan tersebut berdasarkan set fakta yang diketahui. Backward chaining biasanya digunakan dalam situasi di mana sistem pakar memiliki tujuan yang jelas dan terfokus, tetapi menghadapi lingkungan dengan jumlah fakta awal yang besar dan kompleks.



Gambar 3.3. Backward Chaining

Dalam backward chaining, sistem akan terlebih dahulu melihat tujuan atau hasil yang diinginkan dan mencari aturan yang memiliki konsekuensi yang sesuai dengan tujuan tersebut. Jika premis atau kondisi aturan tersebut belum dipenuhi, sistem akan mencari aturan lain yang dapat memenuhi kondisi tersebut. Proses ini berlanjut hingga sistem menemukan aturan yang kondisinya telah dipenuhi oleh fakta yang ada atau tidak menemukan aturan lain yang dapat dipakai.

Metode ini sangat efektif digunakan dalam sistem diagnostik, seperti sistem pakar medis, di mana tujuannya biasanya jelas (misalnya, menentukan diagnosis pasien) tetapi jalan untuk mencapai tujuannya mungkin tidak (Stefik, 2014).



Gambar 3.4. MYCIN

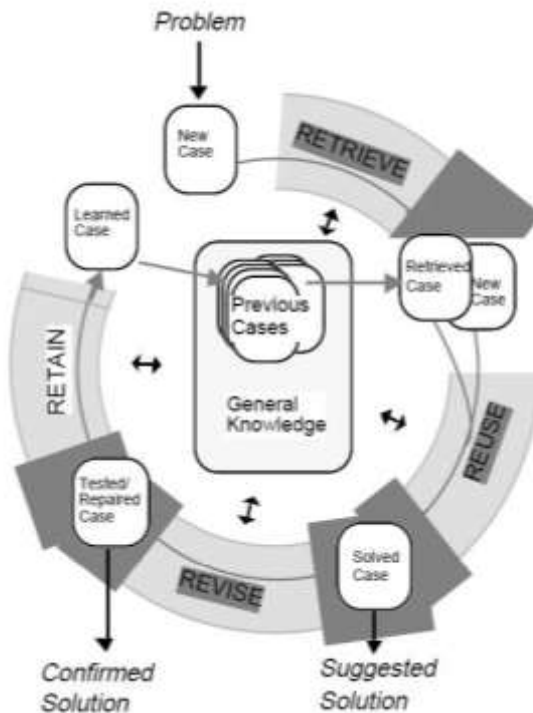
Contoh penerapan backward chaining dalam sistem pakar adalah sistem pakar medis MYCIN. MYCIN adalah sistem yang dirancang untuk mendiagnosis infeksi bakteri dan merekomendasikan antibiotik. Dalam sistem ini, tujuan (diagnosis) ditentukan terlebih dahulu, dan sistem mencoba mencapai tujuan tersebut dengan menanyakan sejumlah pertanyaan kepada pengguna (Shortliffe, 2012).

Namun, perlu dicatat bahwa backward chaining memiliki beberapa keterbatasan. Metode ini bisa menjadi kompleks dan memakan waktu jika terdapat banyak aturan dan tujuan yang tidak jelas. Selain itu, jika terdapat lebih dari satu aturan yang cocok, sistem harus dapat memilih aturan mana yang harus diterapkan terlebih dahulu, yang bisa menjadi tantangan tersendiri.

3.2.2 Penalaran Berbasis Kasus (*Case-Based Reasoning*)

Penalaran Berbasis Kasus (*Case-Based Reasoning*, CBR) adalah pendekatan yang berpusat pada pemecahan masalah dengan cara mencari dan memanfaatkan solusi dari kasus-kasus sebelumnya yang relevan dan mirip (Aamodt & Plaza, 1994). Konsep dasar CBR adalah bahwa individu atau sistem sering kali

memecahkan masalah baru berdasarkan pengetahuan dari pengalaman sebelumnya. Misalnya, dokter mungkin akan mendiagnosis penyakit pasien berdasarkan kasus-kasus serupa yang pernah mereka temui sebelumnya (Richter & Weber, 2016). CBR memiliki empat langkah utama dalam prosesnya: retrieve, reuse, revise, dan retain, yang juga dikenal sebagai siklus 4R (Aamodt & Plaza, 1994).



Gambar 3.5. Siklus CBR

1. *Retrieve*: Pada tahap ini, sistem akan mencari kasus sebelumnya yang mirip dengan kasus baru. Kasus-kasus ini biasanya disimpan dalam basis pengetahuan.

2. *Reuse*: Jika kasus sebelumnya ditemukan, solusi dari kasus tersebut akan dicoba digunakan untuk menyelesaikan kasus baru.
3. *Revise*: Solusi tersebut akan diuji dan jika diperlukan, akan direvisi atau disesuaikan untuk memenuhi kebutuhan kasus baru.
4. *Retain*: Kasus baru dan solusinya yang berhasil akan disimpan dalam basis pengetahuan untuk digunakan pada masa yang akan datang.

Salah satu aplikasi utama CBR adalah dalam bidang diagnostik medis. Dalam hal ini, CBR digunakan untuk membantu dokter dalam proses diagnostik dengan mencari kasus serupa dari pasien sebelumnya dan menggunakan pengetahuan tersebut untuk membantu dalam diagnosis pasien saat ini.

CBR memiliki kelebihan dalam memanfaatkan pengetahuan dari pengalaman sebelumnya dan dapat beradaptasi dengan kasus baru dengan cara memperbarui basis pengetahuan. Namun, CBR juga memiliki keterbatasan, terutama dalam penanganan kasus-kasus yang tidak memiliki kesamaan dengan kasus sebelumnya (Bichindaritz & Marling, 2006).

3.2.3 Penalaran Berbasis Objek (*Object-Based Reasoning*)

Penalaran berbasis objek atau Object-Based Reasoning adalah salah satu metode penalaran dalam sistem pakar yang mengorganisasikan pengetahuan dan melakukan penalaran berdasarkan objek dan atributnya (Wang, 2010). Objek dalam hal ini bisa merujuk ke berbagai entitas di dunia nyata yang memiliki properti atau karakteristik tertentu.

Struktur dasar dalam penalaran berbasis objek adalah konsep objek dan atributnya. Dalam sistem pakar, objek dapat direpresentasikan sebagai kumpulan data dan fungsi yang saling terkait. Setiap objek memiliki atribut yang menjelaskan sifat dan

karakteristik objek tersebut. Misalnya, jika objek adalah "mobil," maka atribut dapat mencakup "merk," "model," "warna," dan "tahun pembuatan."

Metode ini seringkali digunakan dalam sistem pakar yang bekerja dengan struktur data kompleks atau domain pengetahuan yang melibatkan banyak entitas yang berbeda (Schreiber dkk., 1999). Dengan penalaran berbasis objek, sistem pakar dapat membuat inferensi atau kesimpulan berdasarkan hubungan antara objek dan atributnya.

Sebagai contoh, sistem pakar medis dapat menggunakan penalaran berbasis objek untuk mendiagnosa penyakit berdasarkan gejala (objek) dan tingkat keparahan (atribut). Sistem ini kemudian melakukan penalaran berdasarkan hubungan antara gejala dan tingkat keparahan dengan berbagai penyakit yang mungkin.

Sistem ini menggabungkan kekuatan representasi pengetahuan yang fleksibel dan penalaran heuristik untuk mendukung proses pengambilan keputusan. Namun, penalaran berbasis objek mungkin lebih sulit untuk dipahami dan diimplementasikan dibandingkan dengan metode penalaran lainnya karena struktur dan hubungan antara objek dan atribut dapat menjadi sangat kompleks (Studer, Benjamins, & Fensel, 1998).

3.3 Model Penalaran Sistem Pakar

3.3.1 Model Klasik

Model penalaran klasik dalam sistem pakar sering juga disebut Model klasik dalam penalaran sistem pakar merupakan metode penalaran yang paling dasar dan sering digunakan. Model ini mengandalkan pengetahuan yang sudah ditentukan dan dibentuk menjadi aturan atau heuristik dalam sebuah basis pengetahuan (Knowledge Base). Dalam model klasik, sistem pakar akan menggunakan aturan-aturan ini untuk melakukan penalaran

dan mencapai kesimpulan berdasarkan masukan yang diberikan oleh pengguna.

Pada model klasik, biasanya menggunakan dua pendekatan penalaran, yaitu penalaran maju (*forward chaining*) dan penalaran mundur (*backward chaining*). *Forward chaining* adalah pendekatan penalaran dimana sistem pakar mulai dari kumpulan fakta dan berusaha mencapai kesimpulan, sedangkan *backward chaining* adalah pendekatan di mana sistem pakar mulai dari suatu kesimpulan yang dicari dan berusaha mencari fakta-fakta yang mendukung kesimpulan tersebut (Luger & Stubblefield, 2004).

Tabel 3.1. Perbandingan Forward dan Backward Chaining

Kriteria	Forward Chaining	Backward Chaining
Pendekatan	Data-driven, dimulai dari fakta dan bergerak menuju kesimpulan.	Goal-driven, dimulai dari tujuan dan bergerak mundur untuk mencari fakta yang mendukungnya.
Ketika Digunakan	Ketika kita memiliki banyak data dan ingin menarik kesimpulan dari data tersebut.	Ketika kita memiliki tujuan spesifik dan ingin mencari bukti atau fakta yang mendukung tujuan tersebut.
Kompleksitas	Dapat menjadi kompleks jika jumlah data atau fakta yang ada sangat besar.	Dapat menjadi kompleks jika rantai alasan yang dibutuhkan untuk mencapai tujuan sangat panjang.
Contoh Aplikasi	Diagnostik medis, sistem monitoring, penalaran dalam permainan berbasis aturan.	Sistem konsultasi, penyelesaian masalah dalam domain yang sempit, seperti program debugging.

Kriteria	Forward Chaining	Backward Chaining
Kelebihan	Sangat baik dalam menemukan semua solusi yang mungkin dan mengeksplorasi konsekuensi dari fakta yang diketahui.	Sangat efisien jika tujuan yang dicapai dengan baik didefinisikan dan jumlah tujuan yang mungkin adalah sedikit.
Kekurangan	Mungkin memerlukan banyak waktu dan sumber daya jika jumlah aturan dan fakta sangat besar.	Mungkin tidak efisien jika terdapat banyak tujuan potensial atau jika tujuan tidak dapat dicapai.

Untuk mendemonstrasikan model klasik, kita dapat melihat pada sistem pakar medis seperti MYCIN yang dikembangkan pada tahun 1970-an. MYCIN adalah sistem pakar yang dibuat untuk mendiagnosis penyakit seperti meningitis dan infeksi darah. Sistem ini menggunakan serangkaian aturan IF-THEN untuk melakukan diagnosis berdasarkan gejala yang diinput oleh dokter (Shortliffe & Buchanan, 1975).

Tabel 3.2. Contoh Penerapan Aturan IF-THEN dalam MYCIN

No	Premis (Jika...)	Konklusi (Maka...)
1	Organisme adalah Staphylococcus aureus dan sensitif terhadap Penicillin	Rekomendasikan Penicillin
2	Organisme adalah Staphylococcus epidermidis	Rekomendasikan Novobiocin
3	Organisme adalah Streptococcus dan pasien alergi terhadap Penicillin	Rekomendasikan Erythromycin
4	Lokasi infeksi adalah CNS dan organisme adalah Pseudomonas aeruginosa	Rekomendasikan Gentamicin + Carbenicillin

Dalam penerapan model klasik, ada beberapa kelemahan dan kelebihan. Kelebihan dari model klasik antara lain adalah sederhana dan mudah untuk diterapkan, serta dapat memberikan solusi yang cepat dalam kasus-kasus tertentu. Sedangkan kelemahan dari model ini adalah ketidakmampuannya untuk menangani ketidakpastian dan kompleksitas dalam beberapa kasus, serta kesulitan dalam memperbarui basis pengetahuan (Durkin, 1994).

Tabel 3.3. Kelebihan dan Kekurangan Model Penalaran Klasik

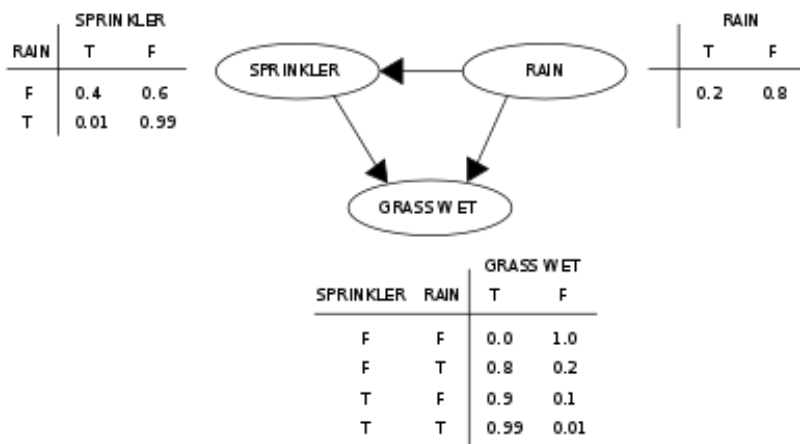
Kriteria	Kelebihan	Kekurangan
Kejelasan	Aturan dan inferensi sangat jelas dan eksplisit. Penalaran dapat diikuti dan ditinjau.	Kejelasan ini mungkin tidak sesuai dengan situasi yang melibatkan ketidakpastian atau ambiguitas.
Ketepatan	Model klasik menyediakan hasil yang pasti dan konsisten asalkan informasi yang diberikan akurat.	Jika data tidak akurat atau tidak lengkap, model mungkin menghasilkan kesimpulan yang salah atau tidak dapat memberikan kesimpulan sama sekali.
Kerumitan	Model klasik seringkali lebih mudah untuk dipahami dan ditinjau.	Penanganan situasi yang kompleks atau yang melibatkan banyak variabel mungkin menjadi sulit atau tidak praktis.
Penanganan Ketidakpastian	Model klasik biasanya tidak dirancang untuk	Inabilitas untuk menangani ketidakpastian berarti

Kriteria	Kelebihan	Kekurangan
	menangani ketidakpastian.	bahwa model klasik mungkin tidak efektif dalam banyak situasi dunia nyata.
Fleksibilitas	Model klasik dapat dengan mudah ditambahkan dan diubah asalkan aturan dan inferensi baru tetap konsisten dengan yang ada.	Kekurangan fleksibilitas dalam menghadapi ketidakpastian atau ambiguitas berarti bahwa model klasik mungkin perlu diperluas atau diganti dengan model yang lebih canggih dalam banyak aplikasi.

3.3.2 Model Probabilistik

Model Probabilistik adalah suatu model penalaran dalam sistem pakar yang menggunakan teori probabilitas untuk mengambil keputusan. Konsep utama dari model probabilistik adalah menggabungkan pengetahuan pakar dengan informasi statistik untuk memprediksi hasil berdasarkan sejumlah kemungkinan (Russell and Norvig, 2021).

Salah satu contoh paling populer dari model penalaran ini adalah Bayes' Net atau Bayesian Network. Bayesian Network merupakan representasi grafis dari dependensi antara variabel dalam suatu sistem. Dalam Bayesian Network, simpul merepresentasikan variabel, dan panah antara simpul menunjukkan hubungan sebab-akibat. Setiap simpul memiliki tabel probabilitas kondisional yang mengkalkulasi probabilitas variabel, mengingat kondisi variabel orang tuanya. Bayesian Network sangat cocok untuk sistem pakar yang memiliki banyak variabel dan relasi yang kompleks antara variabel-variabel tersebut (Pearl, 1988).



Gambar 3.6. Contoh dari Bayesian Network

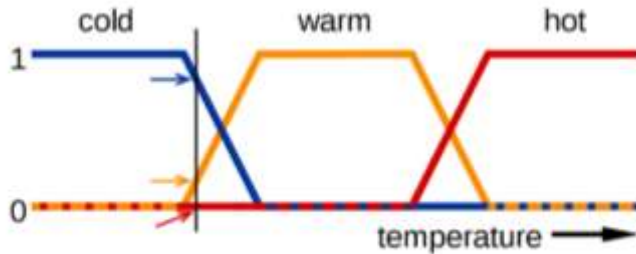
Untuk melakukan penalaran atau inferensi dalam Bayesian Network, biasanya digunakan dua metode: inferensi eksak dan inferensi perkiraan. Inferensi eksak, seperti algoritma eliminasi variabel, memberikan jawaban yang tepat tapi bisa menjadi sangat rumit dan mahal secara komputasi dalam jaringan yang besar. Sebaliknya, metode perkiraan seperti sampling memberikan jawaban yang mendekati benar dan biasanya lebih cepat dalam jaringan yang besar (Koller and Friedman, 2009).

Model probabilistik adalah pilihan yang baik dalam sistem pakar yang harus menangani ketidakpastian. Namun, mereka juga memiliki tantangan dalam skala dan kompleksitas. Pelatihan model probabilistik dapat memerlukan sejumlah besar data dan komputasi intensif, dan dapat menjadi sangat kompleks ketika banyak variabel terlibat (Murphy, 2012).

3.3.3 Model Fuzzy

Penalaran fuzzy atau logika fuzzy adalah salah satu model penalaran dalam sistem pakar yang dibuat oleh Lotfi Zadeh pada tahun 1965. Konsep ini berawal dari ide bahwa tidak semua hal

dalam dunia ini beroperasi dalam keadaan hitam dan putih atau benar dan salah, namun ada keadaan di antara keduanya yang biasa kita sebut sebagai "abu-abu" atau fuzzy (Zadeh, 1965). Model ini sangat berguna dalam mengatasi permasalahan yang melibatkan ketidakpastian atau ambiguitas.



Gambar 3.7. Logika Fuzzy

Sebuah sistem berbasis penalaran fuzzy biasanya terdiri dari tiga komponen utama, yaitu basis pengetahuan, mekanisme penalaran, dan defuzzifikasi. Basis pengetahuan berisi aturan-aturan fuzzy yang didefinisikan dalam bentuk IF-THEN. Mekanisme penalaran bertugas untuk mengevaluasi aturan-aturan tersebut berdasarkan informasi yang diberikan. Defuzzifikasi adalah proses untuk mengubah output fuzzy menjadi nilai crisp yang dapat dipahami (Ross, 2009).

Sebagai contoh, jika kita memiliki sistem pakar yang menggunakan logika fuzzy untuk memberikan rekomendasi aktivitas berdasarkan cuaca, aturan fuzzy mungkin terlihat seperti ini: "Jika cuaca 'cukup panas' maka 'jangan lari di luar ruangan'." Dalam contoh ini, 'cukup panas' adalah variabel fuzzy yang mungkin memiliki banyak interpretasi berbeda. Menggunakan model penalaran fuzzy, sistem ini bisa menangani variasi interpretasi dan memberikan rekomendasi yang tepat.

Tabel 3.4. Contoh aturan fuzzy

Aturan	Jika Suhu...	Maka Kecepatan Kipas...
1	Dingin	Lambat
2	Hangat	Sedang
3	Panas	Cepat

Logika fuzzy telah banyak digunakan dalam berbagai aplikasi, termasuk sistem kontrol, prediksi cuaca, pengolahan citra, dan bahkan dalam sistem pakar medis untuk mendukung pengambilan keputusan (Carlsson dkk., 2012).

Model Fuzzy dalam sistem pakar biasanya melibatkan proses berikut:

1. **Fuzzifikasi:** Mengubah nilai input yang tegas menjadi derajat keanggotaan fuzzy. Ini melibatkan penentuan fungsi keanggotaan yang sesuai.
2. **Penilaian Aturan:** Penentuan derajat kebenaran aturan fuzzy berdasarkan input yang difuzzykan.
3. **Defuzzifikasi:** Mengubah output fuzzy menjadi nilai tegas

Pada dasarnya, model penalaran fuzzy memungkinkan sistem pakar untuk lebih baik dalam menangani ketidakpastian dan ambiguitas, yang seringkali muncul dalam pengambilan keputusan di dunia nyata.

DAFTAR PUSTAKA

- Aamodt, A., & Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), 39-59.
- Bichindaritz, I., & Marling, C. 2006. Case-based reasoning in the health sciences: What's next? *Artificial Intelligence in Medicine*, 36(2), 127-135.
- Carlsson, C., Fedrizzi, M. and Fullér, R., 2012. Fuzzy logic in management (Vol. 66). Springer Science & Business Media.
- Durkin, J. 1994. Expert systems: design and development. Prentice Hall.
- Giarratano, J. & Riley, G. 2004. Expert Systems: Principles and Programming. Course Technology.
- Hurley, P.J., 2014. A concise introduction to logic. Cengage Learning.
- Jackson, P., 1999. Introduction to Expert Systems. Harlow. Reading, Menlo Park Addison-Wesley.
- Koller, D. and Friedman, N., 2009. Probabilistic graphical models: principles and techniques. Cambridge, MA: MIT press.
- Luger, G. F., & Stubblefield, W. A. 2004. Artificial intelligence: structures and strategies for complex problem solving. Pearson Education.
- Murphy, K.P., 2012. Machine learning: a probabilistic perspective. Cambridge, MA: MIT press.
- Pearl, J., 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. San Francisco, CA: Morgan Kaufmann Publishers.
- Richter, M. M., & Weber, R. O. 2013. Case-based reasoning: a textbook. Springer Science & Business Media.
- Ross, T.J., 2009. Fuzzy logic with engineering applications. John Wiley & Sons.
- Russell, S.J., 2021. Artificial Intelligence: A Modern Approach, 4th Global ed.. Pearson Education, Inc..

- Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van De Velde, W., & Wielinga, B. 1999. Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press.
- Shortliffe, E. ed., 2012. Computer-based medical consultations: MYCIN (Vol. 2). Elsevier.
- Shortliffe, E. H., & Buchanan, B. G. 1975. A model of inexact reasoning in medicine. *Mathematical Biosciences*, 23(3-4), 351-379.
- Stefik, M., 2014. Introduction to knowledge systems. Elsevier.
- Studer, R., Benjamins, V. R., & Fensel, D. 1998. Knowledge engineering: principles and methods. *Data & Knowledge Engineering*, 25(1-2), 161-197.
- Wang, P. P. 2010. Advances in cognitive informatics and cognitive computing. Springer Science & Business Media.
- Zadeh, L. A., 1965. Fuzzy sets. *Information and control*, 8(3), 338-353.

BAB 4

ELEMEN DAN STRUKTUR DASAR SISTEM PAKAR

Oleh Johani S Pasaribu

4.1 Pendahuluan

Sistem pakar (*Expert Systems*, ES) adalah sistem berbasis pengetahuan dimana sebelumnya merupakan salah satu bidang penelitian dalam Kecerdasan Buatan (*Artificial Intelligence*, AI) dapat didefinisikan sebagai perangkat lunak intensif pengetahuan yang dapat melakukan beberapa tugas yang biasanya membutuhkan keahlian manusia. Sistem pakar digunakan untuk memecahkan masalah spesifik tertentu dan setiap langkah penalaran untuk masalah tertentu ditentukan oleh pakar manusia secara profesional. Jadi, sistem pakar berperilaku sebagai sistem penasihat buatan untuk masalah tertentu.

Meskipun AI digunakan dalam berbagai aplikasi komersial saat ini, aplikasi sistem pakar terkadang dianggap sebagai “AI” juga. Setelah sistem pakar dipindahkan dari laboratorium penelitian selama awal 1980-an, mereka menjadi lebih populer dan menemukan beberapa bidang aplikasi seperti teknik, kimia, kedokteran, industri, dan banyak lainnya.

Proses konstruksi sistem pakar dengan pengetahuan domain khusus didefinisikan sebagai rekayasa pengetahuan. Sistem pakar berbasis pengetahuan berisi pengetahuan yang diperoleh dari majalah, buku, atau dari wawancara dengan pakar bidang tertentu. Sistem pakar sebagian besar lebih disukai karena menghasilkan solusi yang masuk akal bahkan untuk beberapa masalah tidak terstruktur yang tidak memiliki solusi algoritmik

yang efisien (Giarratano, Joseph C.; Riley, 2004). Selain sistem pakar klasik, ada sistem pakar hybrid saat ini yang menggunakan teknik seperti jaringan syaraf tiruan dan algoritme genetika.

4.2 Konsep Umum Sistem Pakar

Sistem pakar pertama dibangun dengan mewawancarai seorang pakar dan berusaha untuk menangkap pengetahuan, maka diberi istilah "sistem pakar." ES adalah program komputer, yang dibuat dengan memanfaatkan pengalaman pakar domain. ES melakukan fungsi seperti mengajukan pertanyaan dan menjelaskan alasannya. Antarmuka pengguna dari sistem semacam ini berlanjut dengan cara menjawab pertanyaan oleh pengguna akhir.

Kernel sistem pakar memiliki dua komponen utama, yaitu basis pengetahuan dan mesin inferensi. Basis pengetahuan berisi pengetahuan tentang domain pakar. Ini mungkin diwakili oleh fakta sederhana, atau dengan representasi yang lebih kompleks seperti bingkai. Ada juga aturan yang secara eksplisit mewakili keterampilan atau pengetahuan pakar tentang domain yang sedang dipertimbangkan. Sistem pakar menggunakan pengetahuan ini dengan memanfaatkan komponen utama kedua, yaitu mesin inferensi yang memiliki beberapa peran termasuk menentukan bagaimana alasan sistem menggunakan aturan IF-THEN dalam basis pengetahuan. Setelah basis pengetahuan dibangun, ES dapat membuat kesimpulan. Bentuk inferensi yang paling umum adalah *forward and backward chaining* (rangkaiannya maju dan mundur). Proses bergerak maju dari fakta yang diketahui ke kesimpulan yang mengikutinya disebut *forward chaining* (rantai maju). Alternatifnya, proses bekerja mundur dari hipotesis ke fakta-fakta yang diketahui yang mendukungnya, disebut *backward chaining* (rangkaiannya mundur).

4.2.1 Pengetahuan dan Pakar

Pengetahuan dalam sistem pakar dapat berupa pemahaman teoretis atau praktis tentang subjek atau domain tertentu, keahlian ataupun pemahaman yang tersedia dari buku, jurnal dan ahli. Orang-orang yang memiliki pengetahuan atau pemahaman teoretis maupun praktis tentang subjek tertentu disebut sebagai ahli atau pakar. Istilah sistem pakar, sistem berbasis pengetahuan, dan sistem pakar berbasis pengetahuan sering digunakan secara sinonim. Kebanyakan orang menggunakan istilah sistem pakar hanya karena lebih singkat, meskipun mungkin tidak ada keahlian dalam sistem pakar, yang ada hanya pengetahuan.

Pengetahuan seorang ahli khusus untuk satu area masalah (*problem domain*) sebagai lawan dari teknik pemecahan masalah umum. Area masalah seperti sains, mesin, komputer, kedokteran, keuangan, atau elektronik yang dapat ditangani oleh seorang ahli merupakan bidang masalah khusus. Sistem pakar dibuat untuk menjadi pakar dalam satu area masalah saja seperti misalnya, biasanya seseorang tidak mengharapkan pemain catur dapat memiliki pengetahuan atau keahlian tentang kedokteran. Keahlian dalam satu bidang tidak secara otomatis terbawa dalam bidang lainnya.

4.2.2 Rules (aturan) sebagai Teknik Representasi Pengetahuan

Pengetahuan tentang sistem pakar dapat direpresentasikan dalam beberapa cara. Salah satu metode umum untuk merepresentasikan pengetahuan adalah dalam bentuk aturan tipe IF THEN, seperti:

**IF <anteseden> THEN
<kesimpulan>**

Contohnya:

IF zat x adalah tembaga

AND suhu x adalah suhu kamar
THEN fase x padat

Jika ada fakta zat x adalah tembaga dan suhu x adalah suhu kamar. Aturannya dipenuhi dan melakukan tindakan fase x padat.

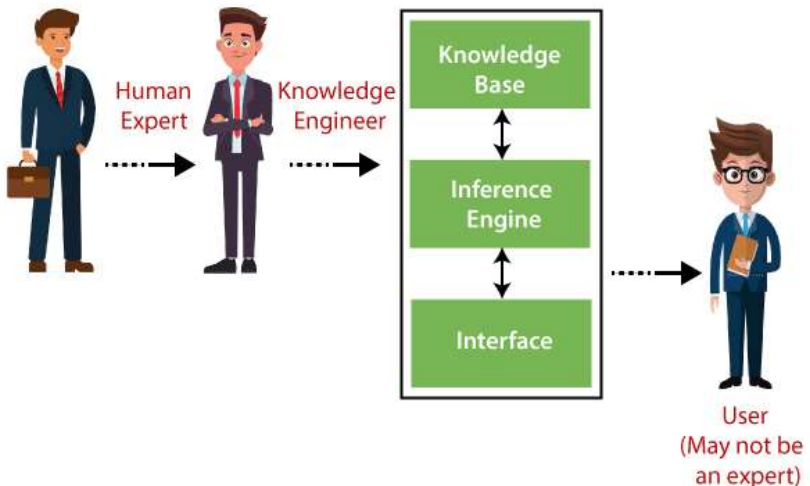
Bagian IF didefinisikan sebagai anteseden yang berisi fakta dan bagian THEN didefinisikan sebagai kesimpulan yang berisi tujuan. Bagian anteseden di mana fakta dieksekusi terlebih dahulu disebut sebagai *data-driven* atau *forward chaining* dan bagian kesimpulan di mana bagian tujuan dari aturan IF-THEN dieksekusi terlebih dahulu disebut sebagai *goal-driven* atau *backward chaining*. Pendekatan *forward chaining* (rantai maju) umumnya lebih disukai dalam kasus pengumpulan data yang sedikit dan mahal dan pendekatan *backward chaining* (rantai mundur) lebih disukai ketika hasil tertentu diperlukan sehubungan dengan jumlah data yang besar (Abraham, 2005).

Meskipun ini adalah contoh yang sangat sederhana, banyak sistem pakar telah dibangun dengan mengungkapkan pengetahuan pakar dalam aturan. Faktanya, pendekatan berbasis pengetahuan untuk mengembangkan sistem pakar telah sepenuhnya menggantikan pendekatan AI awal tahun 1950-an dan 1960-an yang mencoba menggunakan teknik penalaran canggih tanpa ketergantungan pada pengetahuan. Beberapa jenis alat sistem pakar seperti CLIPS memungkinkan objek serta aturan. Pengetahuan dapat dikemas dalam aturan dan objek. Aturan dapat mencocokkan pola pada objek dan juga fakta. Alternatifnya, objek dapat beroperasi secara independen dari aturan.

Sistem pakar klasik mewujudkan pengetahuan tidak tertulis yang harus diekstraksi dari seorang pakar melalui wawancara ekstensif dengan *knowledge engineer* (insinyur pengetahuan) dalam jangka waktu yang lama. Proses membangun sistem pakar disebut *knowledge engineering* (rekayasa pengetahuan) dan dilakukan oleh *knowledge engineer*. Rekayasa pengetahuan

mengacu pada perolehan pengetahuan dari pakar atau sumber lain dan pengkodeannya dalam sistem pakar.

Tahapan umum dalam pengembangan sistem pakar diilustrasikan pada Gambar 4.1. *Knowledge Engineer* (insinyur pengetahuan) pertama-tama membuat dialog dengan pakar untuk memperoleh pengetahuan pakar. Tahapan ini analog dengan seorang perancang sistem dalam pemrograman konvensional yang mendiskusikan kebutuhan sistem dengan klien untuk siapa program akan dibuat. *Knowledge engineer* kemudian mengkodekan kecerdasan (pengetahuan) secara eksplisit dalam *Knowledge Base* (basis pengetahuan). Kemudian *Human expert* (pakar) mengevaluasi sistem pakar tersebut dan memberikan saran atau kritik kepada *Knowledge Engineer*. Proses ini berulang hingga kinerja sistem dinilai memuaskan oleh pakar.



Gambar 4.1. Tahapan pengembangan sistem pakar
(Sumber: (Javatpoint, 2011))

4.2.3 Apa itu Sistem Pakar?

Dalam kecerdasan buatan (AI), sistem pakar adalah aplikasi teknologi kecerdasan buatan yang meniru kemampuan pakar dalam pengambilan keputusan. Sistem pakar bernalar atau membuat kesimpulan (*inferences*) dengan cara yang sama seperti seorang pakar melakukan penalaran (*reasoning*) pengetahuan atau menyimpulkan solusi dari suatu masalah. Yaitu, mengingat beberapa fakta, kesimpulan logis disimpulkan dengan alasan yang mengikuti. Sistem pakar seperti halnya seorang pakar, memecahkan masalah yang kompleks dengan *reasoning* (penalaran) pengetahuan (kecerdasan), dan bukan dengan mengikuti prosedur *developer* (pengembang) seperti pada pemrograman konvensional. Sistem pakar pertama dibuat pada tahun 1970-an dan kemudian berkembang pada tahun 1980-an. Sistem pakar adalah salah satu bentuk perangkat lunak AI pertama yang berhasil dari 6 cabang AI (Neelam Tyagi, 2021).



Gambar 4.2. Enam cabang utama Kecerdasan Buatan
(Sumber: (Neelam Tyagi, 2021))

Gambar 4.2 menunjukkan beberapa bidang AI. Pada tahun 1992, Profesor Edward Albert Feigenbaum dari Universitas Stanford, seorang profesor ilmu komputer di Universitas Stanford

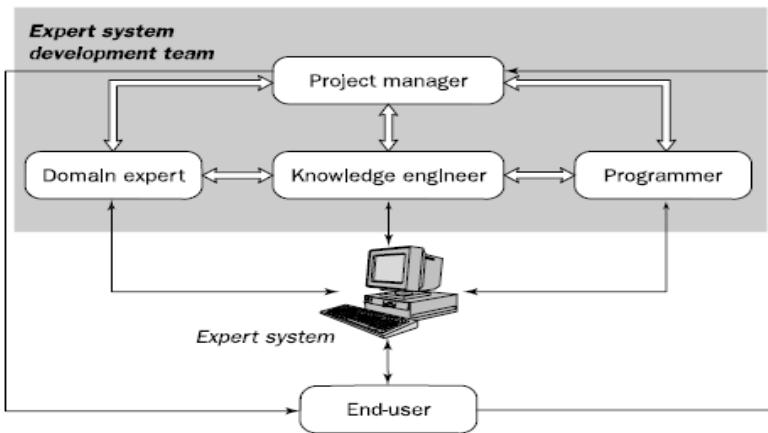
dan pelopor teknologi ES, menyatakan bahwa dunia komputer sedang bergerak maju dari yang awalnya melakukan pemrosesan data menuju "pemrosesan pengetahuan". Sistem pakar merupakan program komputer cerdas yang menggunakan pengetahuan dan prosedur inferensi untuk memecahkan masalah yang cukup sulit dimana membutuhkan keahlian manusia yang signifikan untuk solusinya. Hal ini dimungkinkan karena komputer memiliki potensi untuk melakukan lebih dari perhitungan dasar dan memecahkan masalah yang kompleks berkat teknologi prosesor dan arsitektur komputer baru (Lutkevich, 2018).

Sistem pakar menggunakan pengetahuan khusus secara ekstensif untuk memecahkan masalah pada tingkat pakar. Namun, istilah sistem pakar sering diterapkan saat ini untuk setiap sistem yang menggunakan teknologi sistem pakar. Teknologi sistem pakar dapat mencakup bahasa, program, dan perangkat keras khusus sistem pakar yang dirancang untuk membantu pengembangan dan pelaksanaan sistem pakar.

4.3 Anggota tim pengembangan sistem pakar

Karakteristik dalam tim pengembangan sistem pakar adalah (Irawen, 2018): Secara umum tim pengembangan sistem pakar terdiri dari *knowledge engineer* (insinyur pengetahuan), *programmer* (pemrogram), *domain expert* (pakar domain), *project manager* (pimpinan/manajer proyek), dan *end-user* (klien atau pengguna). Diperlukan komitmen dan kerja sama dari tim pengembangan sistem pakar sehingga sistem pakar yang dihasilkan memang cukup efektif, andal (*reliable*), sesuai dengan keinginan pengguna (*robust*) dan efisien.

Hubungan dasar dalam tim pengembangan dirangkum dalam Gambar 4.3.



Gambar 4.3. *Stake holders* pembuatan proyek ES (sistem pakar)
(Sumber: (Irawen, 2018))

1. *Domain expert* (pakar bidangnya) merupakan orang yang mempunyai kecerdasan, ahli dan terampil dalam permasalahan spesifik di bidang tertentu seperti kedokteran (penyakit), geologi, komputer dan sebagainya. Pakar tersebut harus meluangkan waktu untuk proyek ES tersebut, mampu mengomunikasikan pengetahuannya dan bekerja sama dengan anggota tim lainnya.
2. *Knowledge engineer* (insinyur pengetahuan) merupakan orang yang terlibat dalam tahap analisis, desain, implementasi dan pengujian ES. *Knowledge engineer* mewawancarai *domain expert* untuk mengetahui bagaimana masalah tertentu diselesaikan. Melalui interaksi tersebut, *knowledge engineer* menentukan metode penalaran apa yang digunakan oleh *domain expert* dalam menangani fakta dan aturan serta bagaimana merepresentasikannya dalam ES. Selain itu, *knowledge engineer* juga memilih beberapa perangkat lunak pengembangan atau melihat bahasa pemrograman untuk menyandikan pengetahuan

(kadang kala menyandikannya sendiri). Jadi *knowledge engineer* terlibat mulai dari tahap awal yaitu analisis hingga tahap implementasi dan *delivery* (penyerahan akhir) sistem pakar bahkan juga setelah proyek selesai, *knowledge engineer* terlibat dalam *maintenance* (pemeliharaan sistem). Tentunya disini *knowledge engineer* dibantu anggota tim lainnya tersebut.

3. *Programmer* (pemrogram) yaitu orang yang menulis kode untuk program komputer dan aplikasi. Mereka juga terlibat dalam memelihara, men-debug, dan memecahkan masalah sistem dan perangkat lunak untuk memastikan semuanya berjalan dengan lancar. Peran pemrogram ES adalah menulis kode yang sesuai dengan pengetahuan yang diberikan oleh pakar (*domain expert*). Sistem pakar telah dibangun menggunakan berbagai bahasa pemrograman umum serta khusus. LISP dan PROLOG telah digunakan secara luas. OPS-5 juga populer di kalangan pemrogram berbasis aturan. OPS adalah produk dari Instructable Production System Project di CMU pada tahun 1975. Keterampilan dan kualifikasi pemrogram secara umum adalah kemampuan untuk bekerja secara mandiri, pemahaman mendalam tentang C, Pascal, FORTRAN dan Basic, pengetahuan luas tentang sistem operasi, dokumentasi perangkat lunak dan pengalaman debug dan kemampuan untuk fokus menghasilkan karya berkualitas tinggi.
4. *Project manager/PM* (manajer proyek) adalah yang bertanggung jawab untuk mengembangkan ide proyek menjadi rencana yang dapat ditindaklanjuti dengan tujuan yang jelas. *Project manager* (PM) bekerja dengan klien dan pemangku kepentingan (*stake holders*) untuk menetapkan ekspektasi untuk proyek dan menentukan hasil yang

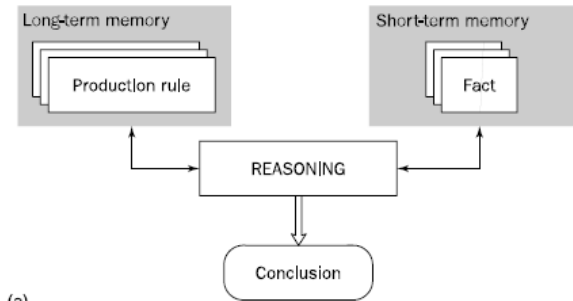
diharapkan. PM juga mengidentifikasi tugas-tugas yang diperlukan untuk menyelesaikan proyek.

5. *End-user* (pengguna akhir) adalah pengguna langsung produk sistem pakar yang dikembangkan oleh tim tersebut. Pengguna akhir memainkan peran yang sangat penting dalam pengembangan produk. Pengguna akhir memberikan umpan balik kepada pengembang, yang membantu memastikan bahwa produk perangkat lunak ES benar-benar memang bermanfaat oleh orang yang membutuhkannya. Keterlibatan pengguna akhir memastikan suara mereka didengar dan dapat secara aktif membentuk dan meningkatkan layanan yang akan mereka gunakan.

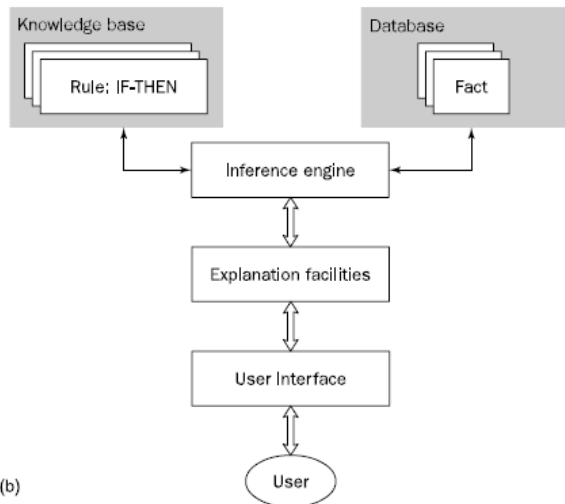
4.4 Elemen dan struktur dasar sistem pakar

Pada tahun 1972, Allen Newell dan Herbert A. Simon dari *Carnegie-Mellon University* mengusulkan model sistem produksi yang menjadi fondasi sistem pakar berbasis aturan modern yaitu teori bahwa orang memecahkan masalah melalui pencarian heuristik melalui ruang masalah (Newell, A. and Simon, 1972). Pendekatan Newell dan Simon, yang didasarkan pada hipotesis ruang masalah, mengemukakan bahwa tingkat pengetahuan merasionalisasikan perilaku dalam hal alasan bahwa seorang agen harus percaya bahwa tindakan tertentu akan mengarah pada pencapaian tujuan tertentu. Dalam pengertian ini pengetahuan adalah sarana untuk mencapai tujuan, sumber daya untuk perilaku. Manusia memecahkan masalah dengan menerapkan pengetahuan (dinyatakan sebagai aturan produksi) untuk masalah tertentu yang diwakili oleh informasi masalah. Aturan produksi disimpan dalam memori jangka panjang dan informasi atau fakta masalah disimpan dalam memori jangka pendek. Hal ini dapat dilihat seperti pada Gambar 4.4.

RBES (*Rule-Based Expert Systems*) atau Sistem pakar berbasis aturan memiliki lima komponen: *knowledge base*, *database*, *inference engine*, *explanation facilities*, dan *user interface* (Giarratano, Joseph C.; Riley, 2004).



(a)



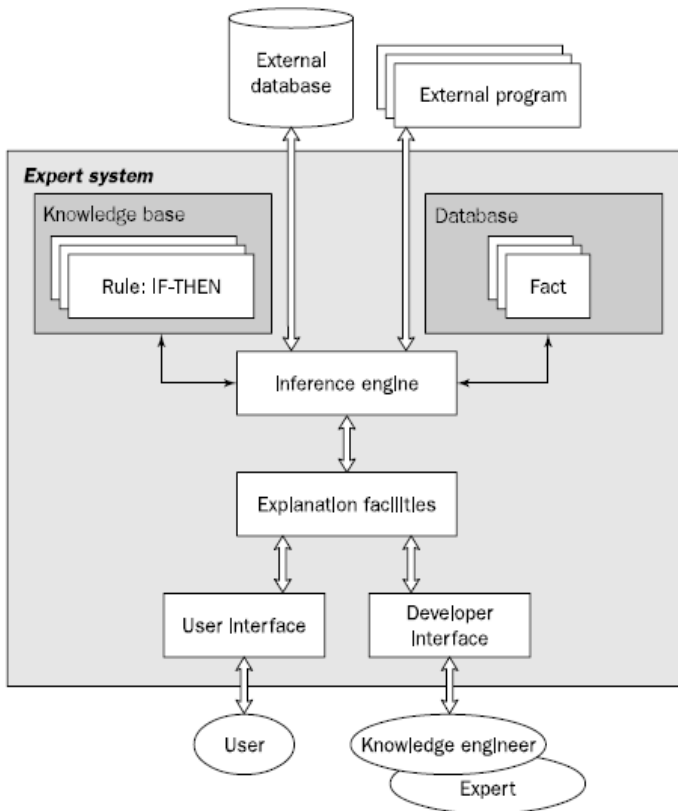
(b)

Gambar 4.4. (a) Model sistem produksi Newell dan Simon dan (b) Struktur dasar sistem pakar berbasis aturan (Sumber: (Ogidan, E. T., Dimililer, K., & Ever, 2018))

1. *Knowledge base* (Basis pengetahuan) berisi pengetahuan (kecerdasan) pakar dalam area tertentu. Basis pengetahuan dalam sistem pakar mewakili fakta dan aturan. Ini berisi pengetahuan dalam domain tertentu bersama dengan aturan untuk memecahkan masalah dan membentuk prosedur yang relevan dengan domain tersebut. Setiap aturan menentukan hubungan, rekomendasi, arahan, strategi atau heuristik dan memiliki struktur IF (kondisi) THEN (tindakan). Kesimpulan ini adalah tanggapan sistem pakar terhadap pertanyaan pengguna untuk keahlian.
2. *Database* (Basis data) berisikan database fakta yang digunakan oleh aturan artinya mencakup sekumpulan fakta yang digunakan untuk *knowledge base* (basis pengetahuan).
3. *Inference engine* (Mesin inferensi). Fungsi paling dasar dari mesin inferensi adalah untuk memperoleh data yang relevan dari basis pengetahuan, menafsirkannya, dan menemukan solusi untuk masalah pengguna. Mesin inferensi juga memiliki kemampuan penjelasan dan debugging. Jadi *inference engine* (mesin inferensi) menghubungkan aturan yang diberikan dalam *knowledge base* (basis pengetahuan) dengan fakta yang disediakan dalam *database* (basis data).
4. *Explanation facilities* (Fasilitas penjelasan). Seperti namanya, modul ini membantu memberikan penjelasan kepada pengguna tentang kesimpulan yang dicapai.
5. *User interface* (antarmuka pengguna). Komponen ini sangat penting bagi pengguna non-ahli untuk berinteraksi dengan sistem pakar dan mencari solusi.
Kelima komponen ini sangat penting untuk setiap sistem pakar berbasis aturan. Mereka merupakan intinya, tetapi ada beberapa komponen tambahan.

6. *External interface* (Antarmuka eksternal). Antarmuka eksternal memungkinkan sistem pakar untuk berinteraksi dengan database dan program lain. Antarmuka pengembang (*developer interface*) memungkinkan pengembang untuk mengedit dengan basis pengetahuan, aturan, dan fakta. Editor teks menyediakan fasilitas semacam notepad untuk memasukkan input.

Struktur lengkap dari sistem pakar berbasis aturan ditunjukkan pada Gambar 4.5. *Developer interface* (Antarmuka pengembang) biasanya mencakup editor basis pengetahuan, alat bantu debugging, dan fasilitas input/output. Semua kerangka sistem pakar menyediakan editor teks sederhana untuk memasukkan dan memodifikasi aturan, dan untuk memeriksa format dan ejaan yang benar.



Gambar 4.5. Struktur lengkap dari sistem pakar berbasis aturan
(Sumber: (Irawen, 2018))

Banyak sistem pakar juga menyertakan fasilitas pembukuan untuk memantau perubahan yang dibuat oleh insinyur pengetahuan atau pakar. Jika aturan diubah, editor akan secara otomatis menyimpan tanggal perubahan dan nama orang yang membuat perubahan untuk referensi selanjutnya. Ini sangat penting ketika sejumlah insinyur pengetahuan dan pakar memiliki akses ke basis pengetahuan dan dapat memodifikasinya.

DAFTAR PUSTAKA

- Abraham, A. 2005. 'Rule-Based Expert Systems', in *Handbook of Measuring System Design*. John Wiley & Sons Ltd, pp. 909–919.
- Giarratano, Joseph C.; Riley, G. D. 2004. *Expert Systems: Principles and Programming*. Fourth Edi. Course Technology.
- Irawen. 2018. *Expert System in Artificial Intelligence, AI*. Available at: <https://www.ntirawen.com/2018/09/expert-system-in-artificial-intelligence.html> (Accessed: 17 June 2023).
- Javatpoint. 2011. *What is an Expert System?*, javatpoint. Available at: <https://www.javatpoint.com/expert-systems-in-artificial-intelligence> (Accessed: 18 June 2023).
- Lutkevich, B. 2018. *DEFINITION expert system*, TechTarget. Available at: <https://www.techtarget.com/searchenterpriseai/definition/expert-system> (Accessed: 18 June 2023).
- Neelam Tyagi. 2021. *6 Major Branches of Artificial Intelligence (AI), analyticssteps*. Available at: <https://www.analyticssteps.com/blogs/6-major-branches-artificial-intelligence-ai> (Accessed: 17 June 2023).
- Newell, A. and Simon, H. A. 1972. *Human Problem Solving*. Englewood Cliffs, NJ.: Prentice-Hall.
- Ogidan, E. T., Dimililer, K., & Ever, Y. K. 2018. 'Machine Learning for Expert Systems in Data Analysis', in *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, p. 5. doi: 10.1109/ismsit.2018.8567251.

BAB 5

METODE PENGEMBANGAN SISTEM PAKAR

Oleh Wahyuddin S

5.1 Pendahuluan

Metode pengembangan sistem pakar adalah suatu pendekatan yang digunakan untuk merancang, meningkatkan, serta menerapkan sistem pakar. Sistem pakar merupakan sistem komputer yang dirancang untuk meniru kemampuan dan pengetahuan seorang pakar manusia dalam suatu domain tertentu. Metode pengembangan sistem pakar bertujuan untuk memperoleh pengetahuan atau pemahaman dari pakar manusia dan mentransfernya ke dalam suatu sistem komputer.

Pendahuluan dalam metode pengembangan sistem pakar meliputi beberapa langkah awal yang perlu dilakukan sebelum memulai proses pengembangan. Langkah-langkah ini membantu dalam memahami kebutuhan sistem pakar dan memastikan keberhasilan pengembangannya. Berikut adalah beberapa pendahuluan yang umumnya dilakukan dalam metode pengembangan sistem pakar:

1. **Identifikasi Masalah**

Tahapan ini melibatkan serta mengidentifikasi masalah atau tugas yang akan diselesaikan oleh sistem pakar. Dalam hal ini, perlu diketahui secara jelas apa yang menjadi fokus sistem pakar, apakah itu diagnosa penyakit, pengambilan keputusan, atau pemecahan masalah dalam suatu domain khusus.

2. **Pengumpulan Pengetahuan**

Langkah ini melibatkan identifikasi pakar manusia yang memiliki pengetahuan dan pengalaman dalam domain yang relevan. Pengetahuan ini kemudian dikumpulkan melalui wawancara, observasi, atau dokumen referensi. Tujuannya adalah untuk memahami pengetahuan yang dimiliki oleh pakar manusia dan menggali informasi yang relevan dengan masalah yang ingin diselesaikan.

3. Representasi Pengetahuan

Setelah pengetahuan dikumpulkan, langkah berikutnya yaitu mewakili pengetahuan tersebut dalam bentuk yang dapat dipahami oleh sistem komputer. Representasi pengetahuan dapat dilakukan menggunakan berbagai teknik, seperti aturan produksi, jaringan semantik, atau logika predikat.

4. Pengembangan Sistem

Tahap ini mengaitkan pengembangan perangkat lunak yang akan menjadi inti dari sistem pakar. Hal ini meliputi pembuatan basis pengetahuan, implementasi aturan produksi, pengembangan antarmuka pengguna, dan pengujian sistem.

5. Evaluasi dan Validasi

Setelah sistem pakar dikembangkan, perlu dilakukan evaluasi dan validasi untuk memastikan bahwa sistem tersebut dapat bekerja dengan baik dan menghasilkan hasil yang akurat. Evaluasi melibatkan pengujian sistem dengan menggunakan kasus-kasus yang diketahui jawabannya oleh pakar manusia. Hasil sistem pakar dibandingkan dengan jawaban yang diharapkan untuk menentukan keakuratan dan kualitas sistem (Dawis et al., 2022).

Pendahuluan dalam metode pengembangan sistem pakar ini penting karena memastikan pemahaman yang tepat terhadap masalah yang ingin diselesaikan, memperoleh pengetahuan yang

relevan, dan merancang sistem yang sesuai. Dengan melakukan langkah-langkah ini dengan cermat, proses pengembangan sistem pakar dapat berjalan dengan lebih efisien dan menghasilkan sistem pakar yang efektif dalam mendukung pengambilan keputusan atau pemecahan masalah (Wahyuddin, Rismayani, et al., 2023).

5.2 Pengertian Metode Penerapan Sistem Pakar

Metode pengembangan sistem pakar adalah serangkaian pendekatan, prosedur, dan teknik yang digunakan untuk merancang, mengembangkan, dan mengimplementasikan sistem pakar. Tujuannya adalah untuk mentransfer pengetahuan dari pakar manusia ke dalam sebuah sistem komputer yang dapat meniru kemampuan dan pengetahuan pakar tersebut.

Metode pengembangan sistem pakar melibatkan langkah-langkah yang terstruktur dan sistematis untuk mengumpulkan pengetahuan pakar, merepresentasikan pengetahuan tersebut dalam bentuk yang dapat dimengerti oleh komputer, mengimplementasikan pengetahuan ke dalam suatu sistem perangkat lunak, dan menguji serta memvalidasi kinerja sistem pakar yang dikembangkan.

Beberapa metode umum yang sering digunakan dalam pengembangan sistem pakar antara lain:

1. Metode Tahap Berulir (*Waterfall Method*)

Metode ini mengikuti pendekatan linier dan terstruktur, dengan langkah-langkah pengembangan yang dilakukan secara berurutan, mulai dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan.

2. Metode Prototyping

Metode ini melibatkan pembuatan prototipe sistem pakar yang berfungsi sebagai representasi awal sistem. Prototipe ini dapat digunakan untuk memvalidasi dan

mengklarifikasi kebutuhan sistem, serta memberikan pemahaman yang lebih baik terhadap pengetahuan pakar.

3. Metode Pengetahuan Berbasis Kasus (*Case-Based Reasoning*)

Metode ini menggunakan pendekatan yang didasarkan pada kasus-kasus sebelumnya yang telah dipecahkan oleh pakar manusia. Sistem pakar akan mengambil pengetahuan dari kasus-kasus yang ada untuk memecahkan masalah yang serupa.

4. Metode Penalaran Berbasis Aturan (*Rule-Based Reasoning*)

Metode ini menggunakan aturan produksi yang terdiri dari kondisi dan tindakan. Sistem pakar akan menggunakan aturan-aturan ini untuk menganalisis fakta-fakta masalah dan menghasilkan kesimpulan atau rekomendasi.

5. Metode Jaringan Semantik (*Semantic Network*)

Metode ini menggunakan representasi pengetahuan dalam bentuk grafik yang terdiri dari simpul-simpul yang saling terhubung. Setiap simpul mewakili konsep atau objek, sedangkan hubungan antar simpul mewakili relasi antara konsep atau objek tersebut (Samosir, Wahyuddin, & ..., 2022).

Pemilihan metode pengembangan sistem pakar tergantung pada kompleksitas masalah yang ingin diselesaikan, ketersediaan pengetahuan pakar, serta karakteristik dan kebutuhan sistem yang akan dikembangkan.

5.3 Metode Penalaran Berbasis Aturan (*Rule-Based Reasoning*)

Metode Penalaran Berbasis Aturan (*Rule-Based Reasoning*) adalah salah satu metode pengembangan sistem pakar yang populer. Metode ini didasarkan pada penggunaan aturan produksi yang terdiri dari kondisi dan tindakan. Aturan produksi

menggambarkan hubungan antara kondisi (premis) dengan tindakan (konklusi) yang harus diambil dalam suatu sistem pakar. Dalam metode penalaran berbasis aturan, sistem pakar menggunakan aturan-aturan tersebut untuk menganalisis fakta-fakta masalah yang diberikan dan mencocokkannya dengan aturan-aturan yang ada dalam basis pengetahuan. Jika kondisi dalam aturan-aturan terpenuhi, maka sistem pakar akan mengambil tindakan yang terkait dengan aturan tersebut.

Proses penalaran dalam metode ini melibatkan langkah-langkah berikut:

1. Penyampaian Fakta

Sistem pakar menerima fakta-fakta masalah yang diberikan oleh pengguna atau sistem lain yang terkait. Fakta-fakta ini dapat berupa input dari pengguna, data sensor, atau informasi yang diambil dari sumber lain.

2. Pencocokan Aturan

Sistem pakar mencocokkan fakta-fakta masalah dengan aturan-aturan yang ada dalam basis pengetahuan. Setiap aturan mengandung kondisi (IF) yang harus dipenuhi agar tindakan (THEN) dapat diambil.

3. Penalaran

Jika kondisi dalam suatu aturan terpenuhi, maka tindakan yang terkait dengan aturan tersebut dijalankan. Proses ini dapat melibatkan pemrosesan beberapa aturan secara simultan.

4. Menghasilkan Kesimpulan

Setelah melalui serangkaian penalaran, sistem pakar menghasilkan kesimpulan atau rekomendasi berdasarkan tindakan yang diambil sesuai dengan aturan-aturan yang terpenuhi.

Metode penalaran berbasis aturan memiliki kelebihan dalam representasi pengetahuan yang mudah dipahami,

fleksibilitas dalam memodifikasi aturan, dan kemampuan dalam menangani pengetahuan yang tidak terstruktur. Namun, metode ini juga memiliki batasan dalam mengatasi ketidakpastian dan kompleksitas dalam pengetahuan (Wahyuddin, Nasution, et al., 2023; Wibowo, Wahyuddin, Darwas, et al., 2023).

Dalam pengembangan sistem pakar, penggunaan metode penalaran berbasis aturan dapat sangat berguna dalam berbagai domain, seperti diagnosis medis, sistem pengambilan keputusan, pemecahan masalah teknis, dan lainnya.

5.4 Metode Pengetahuan Berbasis Kasus (*Case-Based Reasoning*)

Metode Pengetahuan Berbasis Kasus (*Case-Based Reasoning*) adalah sebuah pendekatan dalam kecerdasan buatan yang memanfaatkan pengetahuan dan pengalaman dari kasus-kasus sebelumnya untuk memecahkan masalah pada kasus-kasus baru yang serupa. Metode ini berfokus pada penggunaan pengetahuan yang disimpan dalam basis kasus untuk memberikan solusi atau rekomendasi yang relevan.

Dalam sistem pakar berbasis kasus, basis kasus berfungsi sebagai penyimpanan pengetahuan yang terdiri dari sejumlah kasus yang relevan dengan domain yang sedang dipelajari. Setiap kasus terdiri dari dua komponen utama: deskripsi masalah atau gejala (*simptom*) yang dihadapi dan solusi (*diagnosis* atau *tindakan*) yang diambil untuk menyelesaikan masalah tersebut. Proses penalaran dalam metode pengetahuan berbasis kasus melibatkan beberapa langkah:

1. Pemulihan (*retrieval*)

Berdasarkan deskripsi masalah yang diberikan pada kasus baru, sistem mencari kasus yang paling mirip atau relevan dalam basis kasus yang telah ada.

2. Reutilisasi (*reuse*)

Solusi dari kasus yang mirip atau relevan tersebut digunakan sebagai panduan awal dalam menyelesaikan kasus baru. Solusi ini dapat berupa aturan, penalaran, atau strategi yang diambil dari kasus sebelumnya.

3. Revisi (*revise*)

Solusi yang ditemukan dari kasus sebelumnya dianalisis dan disesuaikan dengan kondisi unik dari kasus baru. Revisi ini melibatkan penyesuaian atau penambahan informasi yang relevan untuk mengoptimalkan solusi.

4. Retensi (*retain*)

Solusi yang telah direvisi dan digunakan untuk menyelesaikan kasus baru disimpan kembali ke basis kasus. Dengan demikian, basis kasus terus berkembang dan diperbarui seiring bertambahnya pengalaman (Permana et al., 2023).

Metode pengetahuan berbasis kasus memiliki beberapa keuntungan. Pertama, metode ini memanfaatkan pengetahuan yang ada dalam basis kasus, sehingga tidak memerlukan proses pembangunan pengetahuan dari awal. Kedua, metode ini dapat menangani situasi di mana solusi atau aturan belum diketahui dengan pasti atau tidak dapat dirumuskan secara tepat. Ketiga, metode ini mampu mempelajari dari pengalaman dan menyesuaikan solusi sesuai dengan kasus baru.

Namun, metode pengetahuan berbasis kasus juga memiliki beberapa tantangan. Pemilihan kasus yang relevan dan proses pengambilan keputusan yang efisien dapat menjadi permasalahan. Selain itu, penyimpanan dan pengelolaan basis kasus yang besar juga dapat menjadi tugas yang kompleks (Wibowo, Wahyuddin, Permana, et al., 2023).

5.5 Metode Jaringan (*Semantic Network*)

Metode Jaringan Semantik (*Semantic Network*) adalah salah satu metode representasi pengetahuan dalam sistem pakar. Metode ini menggunakan grafik berupa jaringan simpul-simpul yang saling terhubung untuk menggambarkan hubungan antara konsep atau objek dalam suatu domain pengetahuan.

Dalam metode jaringan semantik, setiap simpul dalam grafik mewakili konsep atau objek tertentu, sedangkan hubungan antara simpul-simpul tersebut mewakili relasi atau koneksi antara konsep atau objek tersebut. Hubungan ini biasanya diberi label untuk menunjukkan jenis relasi yang ada, seperti "adalah bagian dari", "memiliki atribut", "berhubungan dengan", dan lain sebagainya.

Contoh sederhana dari jaringan semantik adalah sebagai berikut:

```
[Kucing]
|
[Hewan]
/  \
[Mamalia] [Karnivora]
```

Dalam contoh di atas, terdapat tiga simpul yang mewakili konsep "Kucing", "Hewan", "Mamalia", dan "Karnivora". Hubungan antara simpul-simpul tersebut menunjukkan bahwa "Kucing" adalah subkonsep dari "Hewan", "Mamalia" juga merupakan subkonsep dari "Hewan", dan "Kucing" juga merupakan subkonsep dari "Mamalia".

Metode jaringan semantik memungkinkan representasi pengetahuan yang lebih kompleks dengan menggabungkan banyak simpul dan hubungan yang saling terkait. Informasi dapat diakses melalui navigasi melalui simpul-simpul dalam jaringan, sehingga memungkinkan pemodelan hubungan yang lebih kompleks di

antara konsep-konsep dalam domain pengetahuan. Keuntungan dari metode jaringan semantik termasuk kemudahan pemahaman visual, kemampuan untuk menangkap relasi kompleks, dan fleksibilitas dalam memodifikasi dan mengubah struktur jaringan. Namun, metode ini juga memiliki keterbatasan dalam menangani ketidakpastian dan kompleksitas pengetahuan yang lebih besar. Metode jaringan semantik sering digunakan dalam sistem pakar yang berfokus pada domain yang memiliki struktur pengetahuan yang kaya, seperti sistem pakar di bidang linguistik, ontologi, dan pengenalan pola (Sihotang et al., 2023).

5.6 Metode Jaringan Saraf Tiruan (*Artificial Neural Network*)

Metode Jaringan Saraf Tiruan (*Artificial Neural Network*) adalah sebuah pendekatan dalam kecerdasan buatan yang terinspirasi oleh struktur dan fungsi jaringan saraf biologis. Metode ini digunakan untuk memodelkan dan meniru cara kerja otak manusia dalam mempelajari, mengenali pola, dan membuat keputusan. Jaringan saraf tiruan terdiri dari sejumlah besar unit pemrosesan sederhana yang disebut neuron buatan atau node. Neuron-neuron ini saling terhubung dalam jaringan dan berkomunikasi melalui koneksi-koneksi yang memiliki bobot numerik. Setiap neuron menerima input dari neuron lainnya, menggabungkannya, dan menghasilkan output berdasarkan fungsi aktivasi yang ditentukan.

Proses pembelajaran dalam jaringan saraf tiruan terjadi melalui penyesuaian bobot koneksi antara neuron-neuron. Dalam fase pembelajaran, jaringan saraf tiruan diberikan contoh-contoh data latihan yang memiliki pasangan input-output yang diinginkan. Jaringan secara iteratif menghitung outputnya, membandingkannya dengan output yang diharapkan, dan menyesuaikan bobot-bobot koneksi agar kesalahan antara output aktual dan output yang diharapkan diminimalkan. Proses ini biasa

disebut sebagai proses pelatihan (training) atau pembelajaran jaringan. Setelah jaringan saraf tiruan telah dilatih dengan data latihan yang cukup, ia dapat digunakan untuk mengenali dan memproses data baru yang belum pernah dilihat sebelumnya. Jaringan ini dapat digunakan dalam berbagai tugas, seperti pengenalan pola, klasifikasi, regresi, optimisasi, dan lainnya (Wahyuddin, Sudipa, et al., 2023).

Keuntungan utama dari metode jaringan saraf tiruan adalah kemampuannya untuk mempelajari pola-pola yang kompleks dan non-linear dari data. Mereka juga dapat beradaptasi dengan perubahan dalam data masukan atau lingkungan. Selain itu, jaringan saraf tiruan juga dapat melakukan pengolahan paralel yang efisien, sehingga cocok untuk aplikasi yang membutuhkan pemrosesan yang cepat. Namun, metode jaringan saraf tiruan juga memiliki beberapa keterbatasan. Jumlah besar parameter atau bobot yang harus disesuaikan dalam proses pembelajaran dapat memerlukan waktu dan sumber daya komputasi yang besar. Selain itu, interpretasi dan pemahaman tentang bagaimana jaringan saraf tiruan membuat keputusan masih merupakan tantangan aktif dalam bidang penelitian. Penggunaan metode jaringan saraf tiruan dapat ditemukan dalam berbagai bidang, termasuk pengenalan wajah, pengenalan suara, pengenalan tulisan tangan, prediksi harga saham, diagnosa medis, kendali robot, dan lainnya (Santoso et al., 2023).

5.7 Metode Logika Fuzzy (*Fuzzy Logic*)

Metode Logika Fuzzy (*Fuzzy Logic*) adalah sebuah pendekatan dalam kecerdasan buatan yang menggabungkan konsep matematika fuzzy untuk mengatasi ketidakpastian dan ambiguitas dalam pengambilan keputusan. Fuzzy Logic digunakan untuk memodelkan dan mengendalikan sistem yang memiliki karakteristik yang tidak dapat dengan jelas dibedakan antara kategori-kategori yang berbeda.

Konsep dasar dalam logika fuzzy adalah memperluas keadaan "benar" atau "salah" dalam logika Boolean menjadi kontinum yang mencakup tingkat keanggotaan atau kebenaran. Dalam logika fuzzy, suatu pernyataan atau variabel dapat memiliki derajat keanggotaan di antara nilai-nilai 0 dan 1, yang mengindikasikan sejauh mana pernyataan tersebut benar atau tidak benar. Dalam logika fuzzy, aturan-aturan dinyatakan dalam bentuk aturan fuzzy (*fuzzy rules*) yang terdiri dari premis fuzzy (*fuzzy antecedents*) dan konklusi fuzzy (*fuzzy consequents*). Premis fuzzy menggambarkan hubungan antara variabel input dengan fungsi keanggotaan yang disebut himpunan fuzzy, sedangkan konklusi fuzzy menggambarkan hubungan antara variabel output dengan fungsi keanggotaan (Muttaqin, Yuswardi, et al., 2023).

Proses penalaran dalam logika fuzzy melibatkan kombinasi dan penyebaran nilai kebenaran antara premis dan konklusi berdasarkan fungsi keanggotaan yang ditentukan. Metode ini menghasilkan keluaran yang bersifat kabur atau probabilistik, yang menggambarkan sejauh mana suatu keputusan atau tindakan harus diambil. Logika fuzzy memiliki kelebihan dalam mengatasi ketidakpastian dan kompleksitas dalam pengambilan keputusan. Dengan memperkenalkan derajat keanggotaan, logika fuzzy memungkinkan representasi yang lebih fleksibel dan lebih dekat dengan pemikiran manusia. Hal ini membuatnya cocok untuk menghadapi masalah di mana pengukuran dan data yang ada tidak tepat atau dalam jumlah terbatas. Penerapan logika fuzzy dapat ditemukan dalam berbagai bidang, termasuk sistem kendali, pengambilan keputusan, pengenalan pola, sistem pakar, robotika, dan banyak lagi. Dalam sistem kendali fuzzy, misalnya, logika fuzzy digunakan untuk mengendalikan perangkat dengan mempertimbangkan variabel input yang tidak pasti atau kurang tepat, sehingga menghasilkan respons yang adaptif dan dapat disesuaikan dengan kondisi lingkungan.

Metode Jaringan Saraf Tiruan (*Artificial Neural Network*) adalah sebuah pendekatan dalam kecerdasan buatan yang terinspirasi oleh struktur dan fungsi jaringan saraf biologis. Metode ini digunakan untuk memodelkan dan meniru cara kerja otak manusia dalam mempelajari, mengenali pola, dan membuat keputusan (Wahyuddin, Heryana, et al., 2023).

5.8 Metode Sistem Berbasis Agen (*Agent-Based Systems*)

Metode Sistem Berbasis Agen (*Agent-Based Systems*) adalah pendekatan dalam pengembangan sistem yang memodelkan entitas individu yang disebut agen. Setiap agen memiliki kemampuan untuk berinteraksi dengan lingkungannya dan melakukan tindakan berdasarkan tujuannya dan pengetahuannya. Metode ini bertujuan untuk memahami dan mensimulasikan perilaku kolektif yang muncul dari interaksi antara agen-agenya. Dalam sistem berbasis agen, setiap agen dianggap sebagai entitas yang otonom, yang dapat memperoleh informasi dari lingkungannya, memproses informasi tersebut, dan mengambil keputusan atau tindakan berdasarkan pengetahuannya dan tujuannya. Setiap agen juga memiliki keterbatasan dan kemampuan yang unik, serta bisa berinteraksi dengan agen lain dalam sistem.

Sistem berbasis agen memungkinkan pemodelan sistem yang kompleks dan adaptif, di mana perilaku sistem secara keseluruhan muncul dari interaksi antara agen-individu. Selain itu, metode ini dapat menggambarkan dinamika yang rumit, interaksi nonlinier, dan munculnya perilaku kolektif yang sulit dijelaskan melalui pendekatan konvensional. Penerapan sistem berbasis agen dapat ditemukan dalam berbagai bidang, termasuk simulasi sosial, analisis lalu lintas, ekonomi, ekologi, dan banyak lagi. Contohnya, dalam simulasi sosial, agen dapat mewakili individu dalam masyarakat, dan interaksi antara agen dapat digunakan untuk

memodelkan dinamika kelompok, penyebaran informasi, dan fenomena sosial lainnya.

Keuntungan dari metode sistem berbasis agen termasuk kemampuan untuk memodelkan sistem yang kompleks, adaptif, dan dinamis, serta mampu menggambarkan perilaku kolektif yang muncul. Selain itu, pendekatan ini memungkinkan analisis efek dari interaksi antara entitas individu dalam sistem. Namun, sistem berbasis agen juga memiliki tantangan, seperti kompleksitas dalam mendesain dan mengatur perilaku agen, serta sulitnya memvalidasi dan menguji sistem secara menyeluruh karena sifat dinamisnya. Diperlukan penelitian dan pemodelan yang cermat untuk memastikan representasi yang akurat dari entitas dan interaksi dalam sistem (Muttaqin, Simarmata, et al., 2023).

5.9 Metode Penalaran (*Probabilistic Reasoning*)

Metode Penalaran Probabilistik (*Probabilistic Reasoning*) adalah sebuah pendekatan dalam kecerdasan buatan yang menggabungkan konsep probabilitas untuk memodelkan ketidakpastian dalam pengambilan keputusan dan penalaran. Metode ini didasarkan pada teori probabilitas yang memungkinkan pengukuran dan perhitungan tingkat keyakinan atau kebenaran suatu pernyataan.

Dalam penalaran probabilistik, setiap pernyataan atau variabel diwakili oleh distribusi probabilitas yang mengindikasikan sejauh mana pernyataan tersebut benar. Distribusi probabilitas ini mencerminkan tingkat keyakinan atau kebenaran berdasarkan informasi yang tersedia. Metode penalaran probabilistik menggunakan aturan-aturan probabilitas dan teorema Bayes untuk memperbarui distribusi probabilitas berdasarkan bukti atau informasi baru yang diperoleh. Dengan menggabungkan informasi dan memperbarui tingkat keyakinan, metode ini dapat menghasilkan kesimpulan yang lebih akurat dan

mempertimbangkan ketidakpastian dalam pengambilan keputusan.

Contoh penerapan metode penalaran probabilistik adalah dalam sistem pengenalan suara. Dalam hal ini, metode ini digunakan untuk mengukur probabilitas bahwa suara yang dideteksi cocok dengan pola suara yang diketahui sebelumnya. Berdasarkan probabilitas tersebut, sistem dapat mengambil keputusan tentang identitas suara yang dideteksi. Keuntungan dari metode penalaran probabilistik termasuk kemampuan untuk mengatasi ketidakpastian dan ketidaksempurnaan data, serta kemampuan untuk menggabungkan dan memperbarui informasi secara efektif. Pendekatan ini dapat digunakan dalam berbagai aplikasi yang melibatkan pengambilan keputusan yang berdasarkan bukti dan tingkat keyakinan, seperti diagnosis medis, prediksi cuaca, analisis risiko, dan lainnya. Namun, metode penalaran probabilistik juga memiliki beberapa tantangan, seperti perluasan ruang pencarian yang kompleks dan komputasi yang intensif. Selain itu, perlu pemodelan yang akurat dari distribusi probabilitas yang sesuai dengan sifat domain yang dikaji untuk menghasilkan hasil yang optimal (Wahyuddin, Nasution, et al., 2023).

5.10 Metode Genetika (*Genetic Algorithms*)

Metode Genetika (*Genetic Algorithms*) adalah sebuah pendekatan dalam komputasi yang terinspirasi oleh prinsip evolusi biologis. Metode ini digunakan untuk menyelesaikan masalah optimisasi dan pencarian di berbagai domain dengan memanfaatkan konsep seleksi alami, pewarisan genetik, dan rekombinasi.

Dalam metode genetika, solusi masalah direpresentasikan sebagai individu dalam populasi. Setiap individu memiliki kromosom yang terdiri dari serangkaian gen atau nilai-nilai yang merepresentasikan karakteristik solusi. Pada awalnya, populasi

terdiri dari sejumlah individu yang dihasilkan secara acak atau berdasarkan pengetahuan awal. Proses evolusi dimulai dengan evaluasi kualitas setiap individu dalam populasi menggunakan fungsi evaluasi yang mengukur sejauh mana individu tersebut memenuhi kriteria keberhasilan atau tujuan yang diinginkan. Individu yang memiliki kualitas yang lebih tinggi cenderung memiliki nilai fitness yang lebih tinggi (Mayefis et al., 2023).

Selanjutnya, individu-individu dengan nilai fitness yang lebih tinggi memiliki peluang yang lebih besar untuk dipilih sebagai "orangtua" untuk menghasilkan "keturunan" baru. Proses ini mencakup operasi seperti seleksi, *crossover* (rekombinasi), dan mutasi. Dalam operasi *crossover*, bagian-bagian kromosom dari dua individu dipertukarkan untuk menghasilkan keturunan baru. Sedangkan operasi mutasi mengubah nilai-gen individu secara acak. Setelah generasi baru terbentuk, proses evaluasi dan seleksi berlanjut untuk mengevolusikannya. Proses ini berlangsung berulang-ulang hingga tercapai solusi yang diinginkan atau kriteria berhenti lainnya, seperti mencapai jumlah generasi maksimum atau konvergensi ke solusi yang stabil.

Metode genetika memiliki kelebihan dalam menangani masalah optimisasi yang kompleks dan non-linear. Metode ini mampu menjelajahi ruang pencarian secara paralel dan menemukan solusi yang baik meskipun dalam ruang pencarian yang besar. Metode genetika telah diterapkan dalam berbagai bidang, termasuk desain perangkat lunak, optimisasi parameter, pemodelan prediktif, dan lainnya. Namun, metode genetika juga memiliki keterbatasan. Proses evolusi yang berulang dapat memakan waktu yang lama, terutama untuk masalah yang kompleks. Selain itu, representasi individu dan pemilihan operator genetik yang tepat juga merupakan tantangan dalam merancang algoritma genetika yang efektif (Haris et al., 2023).

DAFTAR PUSTAKA

- Dawis, A. M., Himawan, I. S., Meidelfi, D., Ikhrum, F., Intan, I., Harun, R., ... Purnomo, R. 2022. *Artificial Intelligence: Konsep Dasar Dan Kajian Praktis*. TOHAR MEDIA.
- Haris, M. S., Nurlaila, Q., Ratnadewi, R., Friadi, J., Hendrawan, S. A., Meidelfi, D., ... Dewi, A. R. S. 2023. *PENGANTAR DATA SCIENCE*. Penerbit Widina.
- Mayefis, R., Yusnanto, T., Heryana, N., Aulia, A. P., Permana, A. A., Aisa, S., ... Sihombing, F. A. H. 2023. *Rekayasa Perangkat Lunak*. Global Eksekutif Teknologi.
- Muttaqin, M., Simarmata, J., AKP, A. M., Nurzaenab, N., Ashari, I. F., Wahyuddin, S., ... Arifin, S. R. 2023. *Cloud Computing: Konsep dan Implementasi*. Yayasan Kita Menulis.
- Muttaqin, M., Yuswardi, Y., Warni, E., Wahyuddin, S., Bagaskara, R., Ambarwari, A., ... Sinambela, M. 2023. *Konsep Dasar Kecerdasan Buatan*. Yayasan Kita Menulis.
- Permana, A. A., Wahyuddin, S., Santoso, L. W., Wibowo, G. W. N., Wardhani, A. K., Wahidin, A. J., ... Wijayanti, R. R. 2023. *Machine Learning*. Global Eksekutif Teknologi.
- Samosir, K., Wahyuddin, S., & ... 2022. *Sistem Basis Data*. books.google.com.
- Santoso, L. W., Fauzan, R., Kristiana, R., Puspita, S., Nugroho, J. W., & Wahyuddin, S. 2023. *Manajemen Sains*. Global Eksekutif Teknologi.
- Sihotang, J. I., Arni, S., Darsin, D., Ashari, I. F., Siregar, M. N. H., Fadhillah, Y., ... Wirapraja, A. 2023. *Kontrol dan Audit TI*. Yayasan Kita Menulis.
- Wahyuddin, S., Heryana, N., Waworuntu, A., Permana, A. A., Wijayanti, R. R., Pomalingo, S., ... Istiono, W. 2023. *Kontrol Dan Audit Teknologi Informasi*. Global Eksekutif Teknologi.

- Wahyuddin, S., Nasution, M. B. K., Permana, A. A., Dewi, R., Hendarsyah, D., Pakarbudi, A., ... Hermila, A. 2023. *Audit Sistem Informasi*. Global Eksekutif Teknologi.
- Wahyuddin, S., Rismayani, R., Sihotang, J. I., Aisa, S., Gunawan, H., Tamsir, N., ... 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., ... Santoso, L. W. 2023. *Data Mining*. Global Eksekutif Teknologi.
- Wibowo, S. H., Wahyuddin, S., Darwas, R., Pasaribu, J. S., Anggoro, D., Azizah, N., ... Safii, M. 2023. *Sistem Informasi*. Global Eksekutif Teknologi.
- Wibowo, S. H., Wahyuddin, S., Permana, A. A., Sembiring, S., Wahidin, A. J., Nugroho, J. W., ... Adhicandra, I. 2023. *Teknologi Digital Di Era Modern*. Global Eksekutif Teknologi.

BAB 6

METODE AKTIFASI & RULES

SISTEM PAKAR

Oleh Ronald Belferik

6.1 Pengenalan Metode Aktivasi

6.1.1 Definisi dan Peran Metode Aktivasi dalam Sistem Pakar

Metode aktivasi adalah metode untuk mengumpulkan informasi dari pengguna atau sumber lain. Sistem pakar menggunakan pertanyaan terstruktur untuk membuat keputusan yang lebih akurat.

Penggunaan metode aktivasi dalam sistem pakar memiliki beberapa keuntungan. **Pertama**, metode aktivasi memungkinkan sistem pakar untuk mendapatkan informasi yang lebih lengkap dan akurat dari pengguna atau sumber lainnya.

Kedua, penggunaan metode aktivasi dapat membantu mengurangi waktu yang dibutuhkan oleh sistem pakar untuk memberikan solusi dalam waktu yang lebih cepat.

6.1.2 Faktor-faktor yang Mempengaruhi Pemilihan Metode Aktivasi

Beberapa faktor dapat mempengaruhi pilihan metode aktivasi dalam sistem pakar. Faktor-faktor antara lain kompleksitas masalah yang sedang dipecahkan, jumlah dan kualitas data yang tersedia, serta tingkat akurasi dan kecepatan yang diinginkan. Faktor-faktor lain yang dapat dipertimbangkan yakni biaya dan ketersediaan sumber daya yang diperlukan untuk menerapkan metode aktivasi tertentu, serta keahlian dan preferensi perancang dan pengguna sistem pakar.

6.2 Metode Aktivasi dalam Sistem Pakar

6.2.1 Forward Chaining

Metode ini melibatkan proses penyebaran informasi dari fakta-fakta awal ke aturan-aturan yang terkait dalam basis pengetahuan sistem pakar untuk mencapai tujuan tertentu. Dengan kata lain Forward Chaining melakukan pencocokan fakta atau pernyataan, dimulai dari fakta terlebih dahulu (Menggunakan IF) untuk menguji kebenaran hipotesis. (Wiji, 2015). Berikut adalah konsep dan implementasi forward chaining:

Konsep Forward Chaining:

1. **Fakta Awal:** Dimulai dengan fakta-fakta awal atau data masukan yang disediakan oleh pengguna atau sistem eksternal.
2. **Aturan:** Aturan-aturan ini terdiri syarat dan aksi yang harus diambil jika kondisi terpenuhi.
3. **Evaluasi Kondisi:** evaluasi kondisi yang terkait dengan setiap aturan dalam basis pengetahuan. Kondisi-kondisi ini menggambarkan apa yang harus terjadi agar aturan tersebut valid.
4. **Transfer Informasi:** jika fakta awal atau data masukan sesuai dengan kondisi aturan, informasi tersebut ditransfer ke aturan. Informasi ini dapat digunakan sebagai fakta baru dalam sistem pakar atau sebagai dasar untuk mengevaluasi aturan lainnya.
5. **Pengecekan dan Aktivasi Aturan:** jika kondisi aturan terpenuhi, aturan tersebut diaktifkan dan tindakan dijalankan. Tindakan ini dapat berupa solusi, saran, atau informasi tambahan.
6. **Pembaruan dan Iterasi:** Proses pengarsipan dan pengecekan aturan terus berlanjut hingga lebih banyak aturan yang diaktifkan atau tujuan akhir tercapai. Ketika sebuah aturan baru diaktifkan, informasi yang

dihasilkannya disebarkan ke aturan-aturan lain dalam basis pengetahuan.

6.2.2 Backward Chaining

Metode ini melibatkan penarikan kesimpulan dari tujuan yang ditentukan ke belakang melalui aturan-aturan yang ada dalam basis pengetahuan. Dengan kata lain, Pencocokan fakta atau pernyataan dimulai dari bagian sebelah kanan (THEN dulu). Penalaran dimulai dari hipotesis terlebih dahulu untuk menguji kebenaran hipotesis (Dalam menguji kebenaran hipotesis tersebut harus dicari fakta-fakta yang ada dalam basis pengetahuan). Berikut adalah konsep dan implementasi backward chaining:

Konsep Backward Chaining:

1. **Tujuan:** berhubungan dengan tujuan yang ingin dicapai oleh sistem pakar. Tujuannya mungkin untuk mendiagnosis, menentukan keputusan, atau membuat keputusan.
2. **Evaluasi Tujuan:** mengevaluasi tujuan yang ditetapkan oleh pengguna atau sistem. Pada awalnya, tujuan ini dianggap tidak dapat dicapai.
3. **Pengecekan Kondisi:** setelah mengevaluasi tujuan, prasyarat yang terkait dengan setiap aturan basis pengetahuan diperiksa selama pasca-pemrosesan (apakah prasyarat tujuan terpenuhi atau apakah diperlukan langkah tambahan).
4. **Pemanggilan Aturan:** jika kondisi terpenuhi, aturan yang terkait dengan kondisi tersebut diaktifkan. Jika kondisi terpenuhi, aturan diaktifkan. Aturan ini menjadi objek baru dalam proses backward chaining.
5. **Iterasi:** proses perulangan diulang sampai tujuan akhir tercapai. Dalam setiap iterasi, aturan baru dipicu tergantung pada kondisi yang terpenuhi.

6.2.3 Certainty Factor

Certainty Factor merupakan suatu metode yang dapat mendefinisikan ukuran terhadap aturan atau fakta pakar untuk menggambarkan keyakinan seseorang terhadap suatu masalah yang dihadapi.

Certainty factor juga metode untuk menunjukkan apakah suatu kejadian bersifat pasti atau tidak pasti. Alasan penggunaan metode ini karena dapat memberikan hasil yang akurat yang berasal dari perhitungan berdasarkan bobot gejala yang dipilih oleh pakar, mampu memberikan jawaban terhadap permasalahan yang tidak pasti dan dengan menggunakan metode ini, pakar mendeskripsikan keyakinan seorang pakar dalam memberikan pembobotan keyakinan sesuai dengan pengetahuan pakar. (Indriani & Rachmawati, 2018). Berikut adalah konsep dan implementasi Certainty Factor:

1. **Basis Pengetahuan:** basis pengetahuan sistem pakar terdiri dari aturan-aturan yang memiliki kontribusi dalam penalaran atau inferensi. Setiap aturan memiliki tingkat keyakinan yang berkaitan dengan fakta atau kondisi tertentu. Basis pengetahuan ini dapat diisi dengan informasi yang diperoleh dari berbagai sumber, seperti data empiris atau pengetahuan yang dimiliki oleh ahli/pakar di bidang tertentu. (Fahmy, 2023)
2. **Tingkat Keyakinan:** setiap aturan dalam basis pengetahuan memiliki tingkat keyakinan, yang mengindikasikan sejauh mana aturan tersebut mendukung atau bertentangan dengan kesimpulan tertentu. Tingkat keyakinan ini dapat berupa bilangan real antara -1 dan +1.
3. **Certainty Factor: Certainty Factor (CF)** adalah representasi numerik dari tingkat kepercayaan. CF menggambarkan seberapa kuat atau lemahnya hubungan antara aturan dan kesimpulan. Nilai CF positif menunjukkan tingkat keyakinan yang mendukung

kesimpulan, sedangkan nilai CF negatif menunjukkan tingkat keyakinan yang berlawanan dengan kesimpulan.

4. **Kombinasi Certainty Factor:** Ketika beberapa aturan aktif dan berkontribusi pada kesimpulan yang sama, CF dihitung dan digabungkan untuk mendapatkan CF keseluruhan.

6.2.4 Weighted Sum Model

Weighted Sum Model merupakan metode yang sangat umum, dan banyak diterapkan untuk membantu pengambil keputusan dalam mengambil suatu keputusan. (Sarika, 2012)

Weigh Sum Model erupakan bagian dalam metode MCDM(Multi-Criteria Decison Making) dalam mengevaluasi nilai pada setian alternatif.

$$A_i^{WSM-score} = \sum_{j=1}^n w_j x_{ij} , for i = 1,2,3,$$

(Kusrini, 2027)

Dimana :

n = jumlah kriteria

w_j = bobot dari setiap kriteria

x_{ij} = nilai matrik x

Nilai A_i yang paling besar merupakan alternatif yang terpilih

6.2.5 Perbandingan dan Pemilihan Metode Aktivasi yang Tepat

Dalam sistem pakar, metode aktivasi digunakan untuk menghubungkan aturan input atau kumpulan aturan dengan output yang dihasilkan oleh sistem pakar. Metode aktivasi menentukan bagaimana input diubah menjadi output berdasarkan logika atau aturan yang telah ditetapkan.

Berikut adalah beberapa metode aktivasi umum yang harus dipertimbangkan ketika memilih metode yang sesuai:

1. Metode Aktivasi Numerik: metode ini menghasilkan hasil dalam bentuk angka atau skor numerik. Metode ini cocok digunakan ketika aturan menghasilkan respons kuantitatif yang membutuhkan perhitungan atau perbandingan. Sebagai contoh, jika aturan membutuhkan penilaian pada skala 1 sampai 5 atau evaluasi prioritas, metode aktivasi numerik dapat digunakan.
2. Metode Aktivasi Fuzzy:
Metode *Fuzzy Logic* digunakan dalam sistem pakar untuk menangani ketidakpastian atau ambiguitas dalam data dan pengetahuan. Metode ini memungkinkan sistem pakar untuk memproses data yang tidak presisi atau berisi ketidakpastian dan memberikan hasil yang lebih akurat dan sesuai dengan kondisi sebenarnya. Misalnya, dalam sistem pakar diagnosa medis, metode aktivasi fuzzy dapat digunakan untuk menggambarkan tingkat keparahan gejala atau kemungkinan adanya penyakit. (Rony, 2023)
Metode ini menghasilkan output dalam bentuk derajat keanggotaan (*membership degree*) yang mencerminkan tingkat kelayakan atau kebenaran suatu pernyataan.
3. Metode Aktivasi Probabilistik: metode ini menghasilkan hasil dalam bentuk probabilitas atau peringkat. Metode ini tepat digunakan ketika aturan-aturan sistem pakar perlu menangani ketidakpastian berdasarkan informasi statistik atau data probabilistik. Sebagai contoh, jika aturan perlu memperkirakan probabilitas suatu kejadian berdasarkan data historis, maka metode aktivasi probabilistik dapat digunakan.

6.3 Aturan dalam Sistem Pakar

Aturan dalam sistem pakar merupakan pengetahuan yang disimpan dalam bentuk rule (aturan) yang digunakan untuk alur pemecahan suatu masalah. Aturan dalam sistem pakar

direpresentasikan antara IF dan THEN. IF sebagai antecedent dan THEN sebagai consequent.

Aturan dalam sistem pakar juga merupakan pernyataan logika yang digunakan untuk menggambarkan hubungan antara fakta-fakta dan keputusan dalam area tertentu.

Secara Umum sistem pakar (*expert system*) adalah sistem yang bertujuan untuk mengadaptasi pengetahuan atau kecerdasan manusia ke komputer sehingga komputer dapat menyelesaikan masalah dengan cara yang sama seperti manusia. Otak dari sistem pakar adalah mesin inferensi, yang juga dikenal sebagai penerjemah aturan. Komponen ini adalah program komputer yang menyediakan metode untuk penalaran dan perumusan kesimpulan. (Dasril, Ardi, Yeyi, Ilwan, Nursaka, 2020)

Defenisi lain menyatakan bahwa Sistem pakar adalah sistem berbasis komputer yang dirancang untuk meniru pengetahuan dan keterampilan manusia dalam bidang tertentu. Aturan-aturan sistem pakar membantu sistem untuk membuat keputusan dan memecahkan masalah. (Noviyanti, Fra Siskus, Listra, 2022)

Berikut adalah beberapa poin penting tentang aturan dalam sistem pakar:

1. **Hierarki aturan:** aturan-aturan sistem pakar dapat disusun dalam sebuah hirarki. Aturan umum yang berlaku untuk semua kasus ditempatkan dibagian atas dan aturan yang lebih spesifik ditempatkan dibagian bawah. Hal ini memungkinkan sistem pakar untuk mencari aturan yang paling tepat terlebih dahulu, baru kemudian aturan yang paling spesifik.
2. **Kombinasi aturan:** sistem pakar dapat menggunakan beberapa kombinasi aturan untuk menghasilkan kesimpulan. Kombinasi aturan ini dapat dibuat dengan menggunakan logika inferensi, seperti logika fuzzy, aturan produktif, atau logika probabilistik.

3. **Prioritas aturan:** aturan dalam sistem pakar biasanya memiliki prioritas tertentu. Prioritas ini menentukan urutan aturan yang dievaluasi ketika sistem pakar membuat keputusan. Aturan dengan prioritas yang lebih tinggi dievaluasi sebelum aturan dengan prioritas yang lebih rendah.
4. **Penambahan dan penghapusan aturan:** aturan sistem pakar dapat ditambahkan, diperbarui, atau dihapus seiring dengan berkembangnya pengetahuan di lapangan atau seiring dengan bertambahnya pengalaman yang dimiliki oleh sistem pakar. Beberapa sistem pakar memiliki mekanisme yang memungkinkan pengguna dan pakar untuk berkontribusi pada penambahan dan perbaikan aturan.

6.3.1 Pengenalan aturan dalam Sistem Pakar

Dalam sistem pakar, pengenalan aturan adalah proses di mana aturan yang digunakan untuk mendeskripsikan pengetahuan pakar dimasukkan ke dalam sistem pakar. Proses ini melibatkan identifikasi, pencatatan, dan formalisasi aturan yang digunakan untuk memodelkan pengetahuan pakar dalam domain tertentu.

Berikut adalah langkah-langkah umum dalam pengenalan aturan dalam sistem pakar:

1. **Identifikasi pengetahuan pakar:** langkah pertama dalam penemuan aturan adalah mengidentifikasi dan mendokumentasikan pengetahuan pakar. Hal ini melibatkan interaksi dengan para pakar atau meninjau dokumen-dokumen yang relevan untuk mendapatkan pengetahuan yang diperlukan.
2. **Formalisasi aturan:** Setelah aturan ditulis, aturan tersebut perlu diformalisasikan agar dapat dipahami oleh sistem pakar. Hal ini termasuk mengembangkan format yang sesuai untuk menggambarkan aturan, misalnya,

menggunakan sintaks bahasa pemrograman yang sesuai atau format khusus seperti aturan logika fuzzy.

3. **Integrasi aturan:** setelah aturan-aturan diformalisasikan, aturan-aturan tersebut perlu diintegrasikan ke dalam sistem pakar. Ini berarti mengimplementasikan aturan dalam bentuk yang dapat diproses oleh sistem pakar, misalnya, dengan menambahkan aturan ke dalam basis pengetahuan sistem atau membuat struktur data untuk menyimpan aturan.
4. **Verifikasi dan validasi:** setelah aturan-aturan diintegrasikan ke dalam sistem pakar, langkah selanjutnya adalah melakukan verifikasi dan validasi. Hal ini melibatkan pengujian aturan menggunakan data uji atau kasus yang diketahui untuk memastikan bahwa aturan tersebut memberikan hasil yang diharapkan dan konsisten dengan pengetahuan pakar.

6.3.2 Identifikasi Pengetahuan Pakar yang Relevan

Proses identifikasi pengetahuan pakar yang relevan dalam sistem pakar melibatkan langkah-langkah untuk mengumpulkan, memperoleh, dan mendokumentasikan pengetahuan yang dimiliki oleh pakar dalam domain tertentu. Berikut adalah beberapa langkah yang dapat dilakukan untuk mengidentifikasi pengetahuan pakar yang relevan:

1. Membuat Daftar Pertanyaan:
2. Berinteraksi dengan Pakar:
3. Studi Literatur dan Sumber Daya Lainnya:
4. Analisis Kasus dan Pengalaman
5. Menggunakan Metode Elicitation:

Metode elicitation adalah teknik yang digunakan untuk mendapatkan pengetahuan pakar dengan cara mengajukan pertanyaan spesifik atau menggunakan alat dan teknik tertentu. Contohnya yakni wawancara terstruktur,

observasi langsung, studi kasus, atau pengisian kuisioner. Metode ini membantu mengungkapkan pengetahuan pakar yang mungkin tidak terungkap melalui interaksi biasa.

6. Validasi dan Verifikasi Pengetahuan

6.4 Strategi Penulisan Aturan yang Efektif dalam Sistem Pakar

6.4.1 Format Penulisan Aturan yang Jelas dan Tepat

Penulisan aturan yang jelas dan tepat dalam sistem pakar sangat penting agar aturan dapat dipahami dengan baik oleh sistem dan pengguna. Berikut adalah beberapa prinsip dan format penulisan aturan yang direkomendasikan:

1. Format "Jika ... maka ..." (If-Then):

Format yang paling umum digunakan dalam penulisan aturan adalah format "Jika ... maka ...". Aturan ini menggambarkan kondisi yang harus terpenuhi (Jika) dan tindakan atau kesimpulan yang harus diambil (Maka). Format ini menyatakan bahwa jika kondisi tertentu terpenuhi, maka tindakan atau kesimpulan yang terkait dengan kondisi tersebut harus diambil.

Contoh:

Jika saya makan, maka saya akan kenyang

2. Klausa "Dan" (And) dan "Atau" (Or):

Dalam beberapa kasus, kondisi dalam aturan dapat lebih kompleks dan melibatkan beberapa kriteria. Untuk menggambarkan hubungan antara kondisi, klausa "Dan" (And) dan "Atau" (Or) dapat digunakan. Klausa "Dan" digunakan ketika semua kondisi harus terpenuhi, sementara klausa "Atau" digunakan ketika setidaknya satu kondisi harus terpenuhi.

Contoh:

Jika saya sudah mengikuti UTS dan UAS mata kuliah Kecerdasan Buatan, maka saya sudah menyelesaikan mata kuliah tersebut.

Jika saya sudah berlari setiap hari atau berenang setiap hari, maka tubuh saya pasti lebih sehat

3. Format Negasi:

Dalam beberapa kasus, aturan juga dapat melibatkan negasi kondisi. Untuk menggambarkan negasi, kata kunci "Tidak" atau menggunakan tanda negasi seperti tanda seru (!) dapat digunakan.

Contoh:

Jika tidak ada kondisi A, maka lakukan tindakan B.

Jika tidak ada kondisi A terpenuhi, maka lakukan tindakan B.

4. Gunakan Terminologi yang Konsisten:

Pastikan untuk menggunakan terminologi yang sesuai dengan bidangnya/yang sudah dipelajari. Gunakan istilah dan kata-kata yang familiar dan mudah dimengerti oleh pengguna sistem pakar. Hindari istilah-istilah yang ambigu atau kontradiktif yang dapat menyebabkan kebingungan. Juga hindari tautologi atau pengulangan informasi yang tidak perlu dalam aturan.

5. Berikan Nama yang Deskriptif:

Berikan nama yang deskriptif pada setiap aturan agar mudah dikenali dan dipahami. Nama aturan harus mencerminkan isi dan tujuan aturan tersebut.

6. Komentari dan Anotasi:

Tambahkan komentar dan catatan pada aturan, jika perlu, untuk memberikan penjelasan atau konteks lebih lanjut. Komentar dapat membantu untuk lebih memahami aturan dan memberikan wawasan bagi pengguna sistem pakar.

6.4.2 Menangani Ketidakpastian dalam Aturan

Ketidakpastian dalam aturan mengacu pada situasi di mana seseorang tidak memiliki informasi yang tepat atau terdapat ambiguitas dalam pengetahuan pakar yang dinyatakan dalam aturan. Dalam sistem pakar, terdapat beberapa teknik yang digunakan untuk menangani ketidakpastian ini:

1. **Nilai Probabilitas:** dalam beberapa kasus, probabilitas dapat diberikan pada kebenaran atau kepalsuan dari sebuah pernyataan aturan. Sebagai contoh, seseorang dapat mengatakan bahwa probabilitas bahwa suatu pernyataan benar adalah 80%. Probabilitas ini dapat digunakan untuk menggabungkan aturan dengan teknik berbasis probabilitas seperti teori keputusan.
2. **Logika Fuzzy:** Logika Fuzzy adalah peningkatan dari logika Boolean yang berhadapan dengan konsep kebenaran sebagian/ketidakpastian (ambiguitas). Logika Fuzzy digunakan untuk menangani ketidakpastian pada masalah-masalah dengan memiliki banyak alternative jawaban. Dimana logika klasik menyatakan bahwa segala hal dapat diekspresikan dalam istilah biner (benar/salah, hitam atau putih, ya atau tidak) dengan derajat keanggotaan pada suatu himpunan (0 atau 1). Ini memungkinkan representasi yang lebih fleksibel di mana suatu fakta dapat memiliki tingkat kebenaran yang beragam. (Tonggam, Vera, 2022)
3. **Penalaran Non-Monotonik:** penalaran non-monotonik adalah jenis penalaran yang memungkinkan kita untuk menarik kesimpulan yang bersifat tentatif atau sementara. Artinya, kesimpulan yang diambil dapat berubah ketika ada informasi baru yang bertentangan ditambahkan. Penalaran ini berguna ketika aturan-aturan dari sistem pakar mengandung informasi yang tidak lengkap atau tidak pasti.

4. **Teknik Penilaian:** teknik penilaian digunakan untuk menilai keandalan atau kualitas informasi yang terkandung dalam suatu aturan. Dengan memberikan bobot pada aturan atau sumber informasi, kita dapat mengukur tingkat kepercayaan akan kebenaran atau relevansi informasi tersebut.

DAFTAR PUSTAKA

- Wiji. 2015. *Konsep Sistem Pendukung Keputusan*, Malang: Yayasan Edelweis
- A. F. Indriani, E. Y. Rachmawati, and J. D. Fitriana. "Pemanfaatan Metode Certainty Factor dalam Sistem Pakar Diagnosa Penyakit pada Anak," *Techno.Com*, vol. 17, no. 1, pp. 12–22, 2018.
- Asrul, dkk. 2014. *Evaluasi Pembelajaran*. Medan: Cipta pustaka Media.
- Rizal Furwan Ramadhan, Wahyuddin S, Fahmay Rinanda Saputri, Jarwo, Johni S Pasaribu, Elisawati, Muhammad Faza Almaliki, Citra Nurina Prabiantissa, Melissa Indah Fianty, St. Hajrah Mansyur, Angga Aditya Permana, Ratna Dewi, Risanto Darmawan, Suwito Pomalingo, Moh Safii, 2023. *Kecerdasan Buatan Digital*, Padang: PT. Global Eksekutif Teknologi,
- S. Sarika, "Server Selection by using Weighted Sum and Revised Weighted Sum Decision Models," *Int. J. Inf. Commun. Technol. Res.*, vol. 2, no. 6, pp. 499–511, 2012.
- Kusrini, 2007. *Konsep dan Aplikasi Sistem Pendukung Keputusan*., Yogyakarta: Andi
- Rony. 2023. *Fenomena Artificial Intelligence*, Jambi: PT. Sonpedia Publishing Indonesia
- Dasril, Ardi, Yeyi, Ilwan, Nursaka, 2020. *Pengantar Teknologi Informasi*, Solok: ICM Publisher
- Noviyanti, Fra Siskus, Listra. 2022. *Teori dan Aplikasi Artificial Intelligence Studi Kasus Menggunakan Pemrograman Web*, Ponorogo: Uwais Inspirasi Indonesia
- Tonggam, Vera, 2022. *Buku Ajar Sistem Pakar*, Deli Serdang: Penerbit Cattleya Darmaya Fortuna

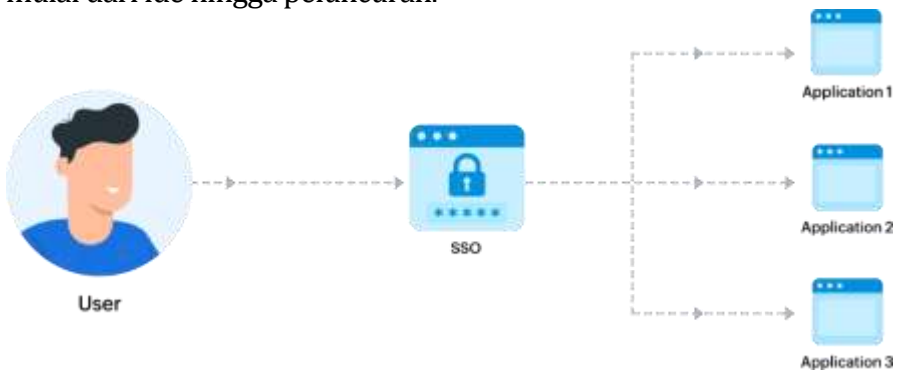
BAB 7

METODE PERANCANGAN APLIKASI & TOOLS SISTEM PAKAR

Oleh Aisyah Mutia Dawis

7.1 Perancangan Aplikasi

Perancangan aplikasi adalah proses yang kompleks dan terstruktur untuk merancang dan mengembangkan aplikasi komputer yang berfungsi untuk memenuhi kebutuhan pengguna. Ini melibatkan serangkaian langkah dan tahapan yang dirancang secara hati-hati untuk mencapai tujuan tertentu (Peranginain, 2022). Perancangan aplikasi adalah langkah awal yang sangat penting dalam pengembangan aplikasi komputer. Itu melibatkan serangkaian proses dan tahapan yang dirancang secara hati-hati untuk menciptakan pengalaman pengguna yang optimal. Dalam narasi ini, kita akan menjelajahi perjalanan perancangan aplikasi mulai dari ide hingga peluncuran.



Gambar 7.1. Perancangan aplikasi
(Sumber : www.manageengine.com)

Semuanya dimulai dengan ide. Di suatu tempat dalam pikiran seorang pengembang atau tim pengembang, muncul suatu konsep yang menarik untuk sebuah aplikasi. Ide ini mungkin timbul dari kebutuhan pengguna yang belum terpenuhi atau peluang di pasar yang belum tergali sepenuhnya. Dalam tahap ini, tim pengembang akan melakukan analisis dan penelitian untuk memastikan bahwa ide tersebut layak dan memungkinkan untuk diimplementasikan.



Gambar 7.2. Tim pengembang melakukan identifikasi
(Sumber : <https://appmaster.io>)

Setelah ide dikonfirmasi, tahap perancangan sebenarnya dimulai. Tim pengembang yang diilustrasikan pada gambar 7.2 akan mengidentifikasi persyaratan dan tujuan aplikasi, serta menganalisis kebutuhan pengguna potensial. Proses ini melibatkan komunikasi yang intens antara tim pengembang dan pemangku kepentingan, seperti manajer produk, desainer, dan pengguna akhir.

Berikutnya, tim pengembang akan merancang antarmuka pengguna. Desain ini haruslah intuitif, menarik, dan mudah

perangkat lunak untuk memastikan bahwa aplikasi berfungsi dengan baik dan tidak ada bug yang signifikan.

Selama proses pengembangan, kolaborasi dan komunikasi yang efektif di antara anggota tim pengembang sangat penting. Mereka harus terus berkomunikasi untuk memastikan bahwa tujuan dan visi yang sama tetap tercapai. Tim pengembang juga harus siap untuk melakukan perbaikan dan penyesuaian saat perubahan dan tantangan muncul.

Setelah tahap pengembangan selesai, tim pengembang akan melanjutkan dengan pengujian aplikasi. Pengujian ini melibatkan berbagai metode, seperti pengujian fungsionalitas, pengujian integrasi, pengujian kinerja, dan pengujian pengguna akhir (Susanto, E.S., Hamdani, F., Nuryansah, F. and Oper, N., 2022). Setiap masalah yang ditemukan selama pengujian harus diperbaiki.

7.2 Metode Perancangan Aplikasi

Metode merupakan langkah-langkah atau prosedur yang digunakan untuk mencapai tujuan tertentu. Berbagai bidang dalam kehidupan menggunakan metode yang berbeda, mulai dari ilmu pengetahuan, teknologi, bisnis, hingga seni. Metode membantu dalam mengatur pikiran, menyederhanakan proses, dan mencapai hasil yang diinginkan (Wijaya, M., 2022).

Dalam ilmu pengetahuan, metode ilmiah digunakan untuk menyelidiki fenomena alam. Metode ini melibatkan pengamatan, merumuskan hipotesis, mengumpulkan data, menganalisis data, dan menyimpulkan temuan. Metode ilmiah membantu menciptakan pengetahuan baru dan memvalidasi teori-teori yang ada (Mujtahidin, M. and Oktariato, M.L., 2022). Berikut adalah beberapa metode perancangan aplikasi yang umum digunakan:

1. Metode *User-Centered Design* (UCD): Metode ini menempatkan pengguna sebagai fokus utama dalam perancangan aplikasi. Melalui penelitian pengguna, observasi, dan pengumpulan umpan balik, UCD

memastikan bahwa kebutuhan dan preferensi pengguna diintegrasikan ke dalam setiap tahap perancangan (Kaligis, D.L. and Fatri, R.R., 2020).

2. Metode Agile: Seperti yang telah dijelaskan sebelumnya, metode Agile melibatkan iterasi pendek dan kolaborasi antara anggota tim untuk menghasilkan perangkat lunak yang responsif dan adaptif. Ini memungkinkan perubahan kebutuhan pengguna atau pasar dapat diakomodasi dengan lebih efisien (Sulistiani, H. and Darwis, D., 2020). Setiap metode memiliki pendekatan dan prinsip yang berbeda dalam perancangan aplikasi. Pilihan metode tergantung pada kebutuhan proyek, tim pengembang, dan karakteristik pengguna yang dituju.

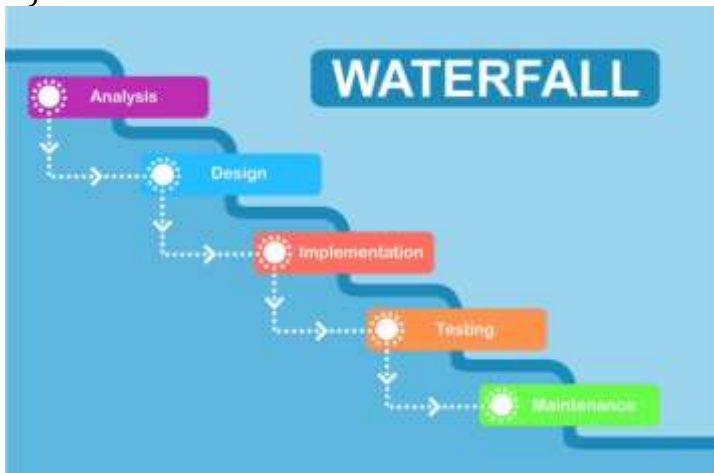


Gambar 7.4. Metode Agile
(Sumber : www.binaracademy.com)

Di bidang teknologi, metode pengembangan perangkat lunak seperti Agile yang tampak pada gambar 7.4 atau Waterfall yang ditunjukkan pada gambar 7.5 digunakan untuk mengatur proses pengembangan perangkat lunak. Metode Agile adalah pendekatan pengembangan perangkat

lunak yang fleksibel dan kolaboratif. Ini bertujuan untuk meningkatkan kecepatan, transparansi, dan adaptabilitas dalam pengembangan produk. Metode Agile didasarkan pada prinsip-prinsip Manifesto Agile yang menekankan individu dan interaksi, perangkat lunak yang berfungsi, kerjasama dengan pelanggan, serta respons terhadap perubahan.

Dalam metode Agile, proyek dibagi menjadi iterasi pendek yang disebut sprint. Setiap sprint berfokus pada tujuan yang jelas dan melibatkan kolaborasi antara anggota tim pengembang dan pemangku kepentingan. Tim bekerja dalam periode waktu yang singkat dan mengejar deliverables yang dapat digunakan atau diuji langsung. Metode Agile memecah proyek menjadi iterasi kecil dan fleksibel, memungkinkan tim untuk beradaptasi dengan perubahan kebutuhan. Sementara metode Waterfall melibatkan tahap-tahap linier yang harus diselesaikan sebelum melanjutkan ke tahap berikutnya (Andrian, D., 2021).



Gambar 7.5. Metode Waterfall
(Sumber : <https://osc.medcom.id/>)

Metode Waterfall adalah salah satu metode pengembangan perangkat lunak yang mengikuti pendekatan linier dan berurutan. Metode ini menguraikan proyek menjadi serangkaian tahap yang harus diselesaikan secara berurutan, mirip dengan aliran air yang mengalir dari atas ke bawah, sehingga mendapatkan nama "Waterfall" atau "Air Terjun". Tahapan dalam metode Waterfall meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan (Yusni, M.R. and Ahmad, L., 2022). Setiap tahap harus selesai sebelum melanjutkan ke tahap berikutnya. Setelah satu tahap selesai, tidak ada kembali ke tahap sebelumnya kecuali untuk perbaikan atau perubahan yang mungkin diperlukan.

Metode Waterfall menekankan pada perencanaan yang matang sebelum memulai pengembangan. Keuntungan dari metode ini adalah memungkinkan dokumentasi yang lengkap, jadwal yang jelas, dan pengendalian proyek yang lebih terstruktur (Anggreini, N.L. and Andiva, A.N., 2023). Metode Waterfall cocok untuk proyek dengan kebutuhan yang stabil dan jelas, di mana perubahan tidak diharapkan secara signifikan. Namun, kelemahan metode Waterfall adalah ketidakmampuannya menangani perubahan kebutuhan yang sering terjadi dalam pengembangan perangkat lunak. Jika perubahan dibutuhkan di tengah jalan, dapat menjadi sulit dan mahal untuk memodifikasi tahapan yang telah selesai. Selain itu, umpan balik dari pengguna atau pemangku kepentingan baru dapat diakomodasi pada tahap-tahap yang lebih lanjut dalam pengembangan.

Metode Waterfall masih digunakan dalam beberapa konteks pengembangan perangkat lunak, terutama di proyek yang memiliki kebutuhan yang stabil dan jelas, serta ketika perubahan dianggap minimal atau dapat dikelola dengan mudah.



Gambar 7.6. Metode Lean UX
(Sumber : <https://id.quora.com/>)

3. Metode Lean UX: Metode ini ditunjukkan pada gambar 7.6 bertujuan untuk mengurangi pemborosan dalam proses perancangan. Dengan fokus pada iterasi cepat dan umpan balik pengguna, Lean UX mempromosikan pengujian cepat dan pemahaman yang mendalam tentang kebutuhan pengguna sebelum perancangan aplikasi dimulai.
4. Metode Design Thinking: Pendekatan ini menekankan pada pemahaman mendalam terhadap pengguna dan tantangan yang dihadapi mereka. Melalui tahap-tahap seperti empati, definisi, ideasi, prototyping, dan pengujian, metode Design Thinking membantu tim merancang aplikasi yang inovatif dan relevan.
5. Metode *Rapid Application Development*: Metode ini merupakan sebuah metode pengembangan perangkat lunak yang bertujuan untuk menghasilkan aplikasi dengan cepat dan efisien. Metode ini menekankan pada kolaborasi antara pengembang, pemangku kepentingan, dan pengguna dalam setiap tahap pengembangan. Dalam RAD, proses pengembangan aplikasi dibagi menjadi serangkaian

iterasi pendek yang disebut "cycles". Setiap siklus melibatkan analisis, desain, konstruksi, dan pengujian komponen aplikasi. RAD menggabungkan penggunaan alat bantu (tools) pengembangan, komponen reusable, dan pendekatan yang terstruktur untuk mempercepat proses pengembangan.

Pendekatan RAD juga mendorong penggunaan prototyping yang cepat. Prototipe digunakan untuk mengumpulkan umpan balik dari pengguna sejak dini, memvalidasi kebutuhan, dan memastikan bahwa aplikasi yang dihasilkan sesuai dengan harapan pengguna. Keuntungan utama dari metode RAD adalah waktu pengembangan yang lebih singkat dan respons yang cepat terhadap perubahan. Dengan fokus pada pengembangan cepat, RAD memungkinkan tim untuk menghasilkan aplikasi yang dapat digunakan atau diuji lebih awal dalam proses pengembangan (Fauzi, S.M. and Wahyuddin, M.I., 2022). Hal ini memungkinkan pengguna untuk memberikan umpan balik yang berharga sehingga perbaikan dan perubahan dapat dilakukan dengan cepat. Namun, perlu diingat bahwa RAD mungkin tidak cocok untuk semua jenis proyek. Proyek-proyek dengan kebutuhan yang sangat kompleks atau yang memerlukan keamanan yang tinggi mungkin memerlukan pendekatan pengembangan yang lebih tradisional dan terstruktur.

6. Metode Gamification: Metode ini menggabungkan elemen-elemen permainan ke dalam perancangan aplikasi untuk meningkatkan keterlibatan pengguna. Dengan menggunakan elemen seperti poin, tantangan, penghargaan, dan kompetisi, metode gamification dapat meningkatkan motivasi dan pengalaman pengguna.

7.3 Tools Sistem Pakar

Dalam pembuatan aplikasi berbasis website, desktop, atau mobile, diperlukan penggunaan alat untuk memastikan bahwa aplikasi yang dibuat dapat beroperasi sesuai dengan harapan. Salah satu bidang dalam kecerdasan buatan (Artificial Intelligence) yang banyak digunakan adalah sistem pakar, yang membantu dalam menyelesaikan berbagai tugas dan masalah dalam kehidupan nyata (Nadeak, M., Ishak, I. and Setiawan, D., 2022). Beberapa contoh aplikasi sistem pakar yang terkenal termasuk Mycin, sebuah aplikasi sistem pakar dari Universitas Stanford yang digunakan untuk mendiagnosis bakteri penyakit tertentu, dan aplikasi sistem pakar dari General Electric yang digunakan untuk melakukan konsultasi kerusakan lokomotif. Berikut beberapa tools yang diperlukan dalam pembuatan aplikasi sistem pakar :

7.3.1 Bahasa Pemrograman

Bahasa pemrograman sistem pakar adalah bahasa yang digunakan untuk mengimplementasikan pengetahuan dan aturan dalam sistem pakar. Bahasa ini memberikan kemampuan kepada pengembang untuk mengekspresikan pengetahuan domain tertentu dan mengatur alur inferensi dalam sistem pakar. Ada beberapa bahasa pemrograman yang umum digunakan dalam pengembangan sistem pakar, di antaranya:

1. Java: Bahasa pemrograman Java sangat populer dalam pengembangan perangkat lunak, termasuk sistem pakar. Java menyediakan berbagai pustaka dan kerangka kerja yang dapat digunakan untuk mengimplementasikan pengetahuan dan algoritma inferensi dalam sistem pakar. Dalam Java, sistem pakar dapat dibangun sebagai kelas-kelas yang berinteraksi dengan pengguna melalui antarmuka pengguna.
2. Python: Python adalah bahasa pemrograman yang serbaguna dan populer, termasuk dalam pengembangan sistem pakar. Python menyediakan pustaka dan kerangka

- kerja seperti Pyke dan pyDatalog yang memudahkan implementasi sistem pakar. Python juga memiliki sintaksis yang mudah dibaca dan dipahami, sehingga cocok untuk pengembangan sistem pakar yang lebih sederhana.
3. Lisp: Lisp adalah keluarga bahasa pemrograman yang sering digunakan dalam kecerdasan buatan, termasuk sistem pakar. Bahasa pemrograman Lisp menggunakan sintaksis berbasis S-expression yang memudahkan penulisan aturan dan pengetahuan dalam sistem pakar. Dalam Lisp, aturan dan pengetahuan dapat diungkapkan secara ekspresif dan fleksibel.

Bahasa pemrograman sistem pakar dipilih berdasarkan kebutuhan dan preferensi pengembang, serta ketersediaan pustaka dan alat bantu yang mendukung. Beberapa bahasa pemrograman sistem pakar juga memiliki kerangka kerja khusus yang menyediakan fitur-fitur tambahan untuk mempercepat pengembangan sistem pakar.

7.3.2 Database

Database dalam sistem pakar berperan penting dalam menyimpan pengetahuan, aturan, dan fakta yang digunakan untuk proses inferensi. Struktur database yang terstruktur dan efisien memungkinkan sistem pakar memberikan solusi cerdas dan mengambil keputusan berdasarkan pengetahuan yang tersimpan (Oktari T, F.Z., 2022). Dengan menggunakan database sistem pakar yang baik, pengembangan, pemeliharaan, dan pembaruan pengetahuan menjadi lebih mudah, sehingga sistem pakar dapat terus berkembang dan memberikan solusi yang akurat dan bermanfaat bagi pengguna.

7.3.3 Web Server

Web server dalam sistem pakar berfungsi sebagai platform untuk menyediakan akses dan interaksi antara pengguna dengan sistem pakar melalui jaringan internet. Dengan menggunakan web server, sistem pakar dapat diakses secara online melalui browser web, memungkinkan pengguna untuk mengajukan pertanyaan, memasukkan data, dan menerima solusi yang dihasilkan oleh sistem (Kurniawan, D., 2022). Web server juga mendukung penyimpanan dan pengelolaan pengetahuan sistem pakar, serta menyediakan antarmuka yang user-friendly. Dengan adanya web server, sistem pakar dapat diakses secara mudah dan fleksibel, meningkatkan aksesibilitas dan pemanfaatan sistem oleh pengguna.

7.3.4 Browser

Browser dalam sistem pakar berfungsi sebagai antarmuka pengguna yang memungkinkan pengguna untuk berinteraksi dengan sistem pakar melalui tampilan grafis yang familiar. Dengan menggunakan browser, pengguna dapat mengakses aplikasi sistem pakar melalui internet atau intranet. Browser menyediakan fitur navigasi, input data, dan visualisasi output yang memudahkan pengguna dalam mengajukan pertanyaan, memasukkan data, dan menerima solusi dari sistem pakar (Hutasuhut, M.D.A., 2022). Dengan antarmuka yang intuitif dan user-friendly, browser membuat sistem pakar lebih mudah diakses dan digunakan oleh pengguna dari berbagai latar belakang, tanpa memerlukan pengetahuan teknis yang mendalam.

DAFTAR PUSTAKA

- Andrian, D., 2021. Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Pengawasan Proyek Berbasis Web. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 2(1), pp.85-93.
- Anggreini, N.L. and Andiva, A.N., 2023. PERANCANGAN DAN IMPLEMENTASI SISTEM INFORMASI PENJUALAN, PEMBELIAN DAN PERSEDIAAN BERBASIS WEB (STUDI KASUS ARI JAYA MOTOR CIPARAY). *Journal of Economics, Accounting, Tax, and Management (JECATAMA)*, 2(1), pp.1-10.
- Fauzi, S.M. and Wahyuddin, M.I., 2022. Penerapan Rapid Application Development (RAD) Dalam Pengembangan Aplikasi Penjualan Ikan Cupang Hias Berbasis Web. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 6(2), pp.751-759.
- Kaligis, D.L. and Fatri, R.R., 2020. Pengembangan Tampilan Antarmuka Aplikasi Survei Berbasis Web Dengan Metode User Centered Design. *JUST IT: Jurnal Sistem Informasi, Teknologi Informasi dan Komputer*, 10(2), pp.106-114.
- Hutasuhut, M.D.A., 2022. Perancangan User Interface (Ui) Dan User Experience (Ux) Situs Bank Masalah Untuk Memonitor Permasalahan Dan Ide Di Pt. Great Giant Pineapple Menggunakan Metode Design Thinking.
- Kurniawan, D., 2022. ANALISIS DAN PERANCANGAN SISTEM INFORMASI PENJUALAN BAHAN BANGUNAN BERBASIS WEB (E-COMMERCE) PADA TOKO BANGUNAN DI DAERAH TIGARAKSA MENGGUNAKAN METODE USER ACCEPTANCE TESTING (UAT) (Doctoral dissertation, KODEUNIVERSITAS041060# UniversitasBuddhiDharma).
- Mujtahidin, M. and Oktariato, M.L., 2022. Metode Penelitian Pendidikan Dasar: Kajian Perspektif Filsafat Ilmu. *Terampil: Jurnal Pendidikan dan Pembelajaran Dasar*, 9(1), pp.95-106.

- Nadeak, M., Ishak, I. and Setiawan, D., 2022. SISTEM PAKAR UNTUK MENDIAGNOSA PENYAKIT KANKER PAYUDARA (CARCINOMA MAMMAE) DENGAN MENGGUNAKAN METODE TEOREMA BAYES PADA RUMAH SAKIT VINA ESTETIKA MEDAN. *Jurnal Cyber Tech*, 1(5).
- Oktari T, F.Z., 2022. Sistem Pakar Diagnosa Tingkat Stres Siswa SMK Selama Pembelajaran Daring (online) Menggunakan Metode Certainty Factor (Studi Kasus: SMK Negeri 1 Pasir Penyus) (Doctoral dissertation, Universitas Islam Riau).
- Peranginain, kasiman. 2022. 'PERANCANGAN APLIKASI PEMBAYARAN BUKU PAKET DAN BUKU MODUL PADA SEKOLAH DASAR SWASTA HANG TUAH 3 BERBASIS JAVA Ita', *Informatika*, 03(03), pp. 477–483.
- Putra, Y.P. and Musril, H.A. 2022. 'Perancangan Aplikasi E-modul Pembelajaran Informatika di MTs Negeri 6 Agam', *Jurnal Edukasi Elektro*, 06(1), pp. 18–23.
- Solichuddin, R.B., 2021. Perancangan User Interface Dan User Experience Dengan Metode User Centered Design Pada Situs Web" Kalografi".
- Sulistiani, H. and Darwis, D., 2020. Penerapan Metode Agile untuk Pengembangan Online Analytical Processing (OLAP) pada Data Penjualan (Studi Kasus: CV Adilia Lestari). *J. Coreit*, 6(1), pp.50-56.
- Supiana, N., 2022. Pengembangan Aplikasi Geolocation Untuk Monitoring Lokasi Mahasiswa Selama Pandemi Berbasis Android Menggunakan Metode Waterfall (Studi Kasus: Stmik Insan Pembangunan. *Jurnal Khatulistiwa Informatika*, 10(1), pp.74-80.
- Susanto, E.S., Hamdani, F., Nuryansah, F. and Oper, N., 2022. Pengembangan Aplikasi Smart-Book Sebagai Media Pembelajaran Bahasa Inggris Anak Berbasis Ar (Augmented Reality). *Mnemonic: Jurnal Teknik Informatika*, 5(1), pp.64-71.

- Wijaya, M., 2022. Konseptual Perancangan Arsitektur Enterprise dengan Kerangka Kerja The Open Group Architecture Framework (TOGAF). *Media Informatika*, 21(3), pp.153-162.
- Yudistia, T.T. and Nisa, D.A., 2023. USER INTERFACE WEBSITE KULINER KHAS KOTA MADIUN MENGGUNAKAN DESIGN THINKING UNTUK PROMOSI UMKM LOKAL. *Jurnal Nawala Visual*, 5(1), pp.11-19.
- Yusni, M.R. and Ahmad, L., 2022. SISTEM INFORMASI PENDAFTARAN ORGANISASI MASYARAKAT PADA KESBANGPOL DAN HUMAS PROVINSI ACEH. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 3(1), pp.30-36.

BAB 8

PENERAPAN ALGORITMA K-NEAREST NEIGHBOUR DALAM SISTEM PAKAR

Oleh Andryanto. A

8.1 Pendahuluan

Sistem Pakar merupakan sistem komputer yang dirancang untuk meniru pengetahuan dan keterampilan seorang ahli manusia dalam suatu bidang tertentu (Medsker dan Bailey, 2020). Sistem Pakar dapat memberikan solusi dan rekomendasi yang akurat dalam situasi yang kompleks, di mana pengetahuan ahli manusia diperlukan (Varlamov, 2021). Sistem Pakar mampu memproses informasi yang rumit dan memecahkan masalah berdasarkan pengetahuan dan aturan yang telah diprogram di dalamnya (Chen, Jia dan Xiang, 2020).

Penggunaan Sistem Pakar sangat luas dan dapat diterapkan dalam berbagai bidang, seperti diagnosis medis, perencanaan bisnis, pengenalan pola, analisis keuangan, pengelolaan sumber daya, dan lain sebagainya (Mardani *et al.*, 2019; Pramanik *et al.*, 2020; Zhang dan Lu, 2021). Sehingga, pengguna dapat mengakses pengetahuan ahli dalam bidang tertentu secara efisien dan mendapatkan rekomendasi yang konsisten dan akurat.

Algoritma K-Nearest Neighbour (KNN) merupakan salah satu algoritma klasifikasi yang populer dalam *machine learning* dan *data mining* (Sarker *et al.*, 2020). KNN merupakan algoritma non-parametrik yang digunakan untuk mengklasifikasikan objek berdasarkan keterkaitan jarak antara objek yang ada di dataset (Deng *et al.*, 2023). Algoritma ini termasuk dalam kategori

“*unsupervised learning*” karena tidak memerlukan proses pembelajaran berlabel sebelumnya (Chen *et al.*, 2021).

8.2 Algoritma K-Nearest Neighbour (KNN)

Algoritma K-Nearest Neighbour (KNN) adalah salah satu algoritma yang digunakan dalam *machine learning* untuk tugas klasifikasi dan regresi (Kang, 2021; Bansal, Goyal dan Choudhary, 2022). Algoritma ini bekerja berdasarkan prinsip bahwa objek yang serupa cenderung berkumpul bersama dalam ruang fitur yang sama. KNN termasuk dalam kategori algoritma “*lazy learning*” atau “*instance-based learning*” karena tidak memerlukan proses pelatihan yang kompleks, melainkan menyimpan seluruh data pelatihan di memori (Sen, Hajra dan Ghosh, 2020; Dridi, 2021).

8.2.1 Cara kerja algoritma KNN dalam klasifikasi data

Cara kerja algoritma K-Nearest Neighbour (KNN) dalam klasifikasi data melibatkan beberapa langkah berikut:

1. Menyiapkan data pelatihan yang terdiri dari objek-objek yang telah diklasifikasikan sebelumnya. Setiap objek dalam data pelatihan memiliki atribut atau fitur yang menggambarkan karakteristik objek tersebut, serta label kelas yang sesuai. Data pelatihan ini akan digunakan untuk melatih model KNN.
2. Selanjutnya memilih nilai K yang tepat untuk algoritma KNN. Nilai K menentukan jumlah tetangga terdekat yang akan digunakan dalam proses klasifikasi. Pemilihan nilai K yang tepat penting untuk mencapai hasil yang optimal. Jika K terlalu kecil, model akan menjadi sensitif terhadap *noise* dan variasi data. Jika K terlalu besar, model dapat menghasilkan kebanyakan mayoritas kelas.
3. Setelah data pelatihan siap, algoritma KNN menghitung jarak antara objek baru yang akan diklasifikasikan dengan setiap objek dalam data pelatihan. Metrik jarak yang umum digunakan adalah jarak *Euclidean* atau jarak *Manhattan*.

Jarak ini menggambarkan seberapa dekat atau seberapa jauh objek baru dari objek dalam data pelatihan.

4. Setelah jarak dihitung, langkah berikutnya menentukan K tetangga terdekat dari objek baru berdasarkan nilai K yang telah ditentukan sebelumnya. Tetangga terdekat ini adalah objek-objek dalam data pelatihan yang memiliki jarak terdekat dengan objek baru.
5. Setelah tetangga terdekat ditentukan, algoritma KNN melakukan voting mayoritas untuk menentukan kelas objek baru. Dalam langkah ini, setiap tetangga terdekat memberikan suara berdasarkan label kelasnya. Kelas yang paling sering muncul di antara tetangga terdekat akan menjadi prediksi kelas objek baru. Misalnya, jika ada 5 tetangga terdekat dan 3 tetangga termasuk dalam kelas A, sementara 2 tetangga termasuk dalam kelas B, maka objek baru akan diklasifikasikan sebagai kelas A.
6. Setelah voting mayoritas dilakukan, algoritma KNN memberikan prediksi kelas untuk objek baru. Prediksi ini dapat digunakan untuk mengklasifikasikan objek baru ke dalam salah satu kelas yang ada dalam data pelatihan.
7. Langkah terakhir mengevaluasi kinerja model KNN. Hal ini dilakukan dengan membandingkan prediksi model dengan label kelas yang sebenarnya pada data pengujian yang terpisah. Metrik evaluasi yang umum digunakan meliputi akurasi, presisi, recall, dan F1-score.

8.2.2 Metrik jarak yang digunakan dalam algoritma KNN

Dalam algoritma *K-Nearest Neighbour* (KNN), metrik jarak digunakan untuk mengukur kedekatan atau kesamaan antara objek baru yang akan diklasifikasikan dengan objek yang ada dalam data pelatihan (Ali, Neagu dan Trundle, 2019; Putra, Novelan dan Rizki, 2022).

1. Jarak *Euclidean* merupakan metrik jarak yang paling umum digunakan dalam algoritma KNN. Metrik ini mengukur jarak lintas lurus antara dua titik dalam ruang fitur. Dalam ruang dua dimensi, jarak Euclidean antara dua titik (x_1, y_1) dan (x_2, y_2) dihitung dengan rumus:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Dalam ruang yang lebih tinggi, rumusnya menjadi:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + \dots + (n_2 - n_1)^2}$$

2. Jarak Manhattan, juga dikenal sebagai jarak kota blok, mengukur jarak antara dua titik sebagai jumlah perbedaan absolut dari koordinat atribut mereka. Metrik jarak Manhattan cocok digunakan ketika atribut memiliki skala yang berbeda atau ketika perbedaan dalam satu atribut lebih penting daripada yang lain. Dalam ruang dua dimensi, jarak Manhattan antara dua titik (x_1, y_1) dan (x_2, y_2) dihitung dengan rumus:

$$d = |x_2 - x_1| + |y_2 - y_1|$$

Dalam ruang yang lebih tinggi, rumusnya menjadi:

$$d = |x_2 - x_1| + |y_2 - y_1| + \dots + |n_2 - n_1|$$

3. Jarak Minkowski merupakan generalisasi dari jarak Euclidean dan jarak Manhattan. Metrik jarak Minkowski memungkinkan kita untuk mengontrol derajat kepekaan terhadap perbedaan dalam setiap atribut. Dalam jarak Minkowski, parameter p digunakan untuk mengontrol jenis metrik yang digunakan. Ketika $p=1$, jarak Minkowski menjadi jarak Manhattan, dan ketika $p=2$, jarak Minkowski menjadi jarak Euclidean. Rumus jarak Minkowski antara dua titik $(x_1, y_1, \dots, n_1)$ dan $(x_2, y_2, \dots, n_2)$ dalam ruang n -dimensi yaitu:

$$d = (|x_2 - x_1|^p + |y_2 - y_1|^p + \dots + |n_2 - n_1|^p)^{1/p}$$

Penggunaan parameter p yang berbeda dalam jarak Minkowski mempengaruhi sensitivitas terhadap perbedaan dalam atribut.

Pemilihan metrik jarak yang tepat tergantung pada karakteristik data dan domain masalah yang sedang dihadapi. Biasanya, jarak Euclidean menjadi pilihan default karena ketergantungannya yang sederhana dan kecocokannya dengan data numerik. Namun, terkadang jarak Euclidean tidak selalu menjadi pilihan yang optimal.

Pada pemilihan metrik jarak untuk algoritma KNN, ada beberapa pertimbangan yang perlu diperhatikan. Pertama, tipe data atribut harus dipertimbangkan. Jika data terdiri dari atribut kategorikal atau data biner, jarak Euclidean tidak cocok, dan metrik jarak seperti jarak Hamming atau jarak Jaccard dapat digunakan. Selanjutnya, skala atribut harus dipertimbangkan. Jika atribut memiliki skala yang berbeda, normalisasi atau menggunakan metrik jarak yang lebih tahan terhadap perbedaan skala seperti jarak Manhattan dapat diterapkan. Kepekaan terhadap outlier juga perlu diperhatikan, dan jika data mengandung outlier yang signifikan, metrik jarak seperti jarak Mahalanobis dapat digunakan. Akhirnya, pemilihan metrik jarak juga harus sesuai dengan domain masalah yang sedang dihadapi, dengan mempertimbangkan hubungan geometris atau kompleksitas hubungan atribut. Misalnya, dalam pengenalan wajah, metrik jarak seperti jarak cosine dapat lebih cocok untuk menangkap kemiripan sudut dan arah fitur wajah.

8.2.3 Cara kerja algoritma KNN dalam regresi

Algoritma *K-Nearest Neighbors* (KNN) juga dapat digunakan untuk memecahkan masalah regresi, di mana tujuan utamanya adalah memprediksi nilai kontinu berdasarkan atribut-atribut yang ada (Nabipour *et al.*, 2020; Cubillos, Wøhlk dan Wulff, 2022). Berikut langkah-langkah detail mengenai cara kerja algoritma KNN dalam regresi:

1. Seperti dalam algoritma KNN untuk klasifikasi, langkah pertama adalah memilih nilai K yang tepat. Nilai K

menentukan jumlah tetangga terdekat yang akan digunakan dalam proses regresi. Nilai K yang kecil akan menghasilkan prediksi yang lebih variatif, sedangkan nilai K yang besar akan menghasilkan prediksi yang lebih rata-rata.

2. Setelah memilih nilai K, langkah selanjutnya menghitung jarak antara objek baru yang akan diprediksi dengan setiap objek dalam data pelatihan. Metrik jarak yang umum digunakan adalah jarak Euclidean, tetapi metrik jarak lain seperti jarak Manhattan atau jarak Minkowski juga dapat digunakan tergantung pada karakteristik data.
3. Dalam algoritma KNN, kita memilih K objek dengan jarak terdekat dari objek baru sebagai tetangga terdekat. Ini dapat dilakukan dengan mengurutkan objek berdasarkan jarak dan memilih K objek pertama.
4. Setelah mengidentifikasi K tetangga terdekat, langkah berikutnya adalah melakukan regresi untuk memprediksi nilai kontinu untuk objek baru. Terdapat beberapa pendekatan yang dapat digunakan dalam regresi KNN. Pendekatan pertama yaitu regresi rata-rata, di mana nilai prediksi dihasilkan dengan mengambil rata-rata nilai target dari K tetangga terdekat. Misalnya, jika terdapat 5 tetangga dengan nilai target 4, 6, 7, 5, dan 8, maka nilai prediksi akan menjadi $(4 + 6 + 7 + 5 + 8) / 5 = 6$. Pendekatan kedua yaitu regresi berbasis bobot, di mana bobot yang berbeda diberikan kepada masing-masing tetangga terdekat berdasarkan jaraknya. Bobot yang lebih besar diberikan kepada tetangga yang lebih dekat, dan bobot yang lebih kecil diberikan kepada tetangga yang lebih jauh. Bobot ini dapat dihitung menggunakan fungsi kernel seperti fungsi Gaussian atau invers jarak, dan digunakan untuk memberikan kontribusi yang lebih besar dari tetangga yang lebih dekat dalam menghitung nilai prediksi. Pendekatan ketiga adalah regresi lokal, di mana diasumsikan bahwa

nilai target objek baru dapat diprediksi dengan menggunakan subkelompok tetangga terdekat yang memiliki karakteristik yang serupa dengan objek baru. Dalam hal ini, metode regresi seperti regresi linear dapat diterapkan pada subkelompok tetangga terdekat untuk memperoleh nilai prediksi.

5. Setelah melakukan regresi, langkah terakhir melakukan evaluasi model regresi KNN. Metrik evaluasi yang umum digunakan dalam regresi yaitu *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), dan koefisien determinasi (R-squared). Metrik-metrik ini membantu dalam mengukur sejauh mana model KNN mampu memprediksi nilai target secara akurat. Berikut penjelasan metrik evaluasi tersebut:

- a. MSE mengukur rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya. MSE dihitung dengan menjumlahkan selisih kuadrat antara setiap prediksi dan nilai sebenarnya, kemudian membaginya dengan jumlah total sampel. Semakin kecil nilai MSE, semakin baik kualitas prediksi model.

Rumus Mean Squared Error (MSE):

$$MSE = (1/n) * \sum (y_i - \hat{y}_i)^2$$

di mana:

n = jumlah sampel

y_i = nilai sebenarnya (target)

$\hat{y}_i$ = nilai prediksi dari model KNN

MSE mengukur rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya.

- b. MAE mengukur rata-rata dari selisih absolut antara nilai prediksi dan nilai sebenarnya. MAE dihitung dengan menjumlahkan selisih absolut antara setiap prediksi dan nilai sebenarnya, kemudian membaginya dengan jumlah total sampel. MAE memberikan gambaran tentang besarnya kesalahan prediksi dalam

satuan yang sama dengan variabel target. Semakin kecil nilai MAE, semakin baik kualitas prediksi model.

Rumus Mean Absolute Error (MAE):

$$MAE = (1/n) * \sum |y_i - \hat{y}_i|$$

di mana:

n = jumlah sampel

y_i = nilai sebenarnya (target)

$\hat{y}_i$ = nilai prediksi dari model KNN

MAE mengukur rata-rata dari selisih absolut antara nilai prediksi dan nilai sebenarnya.

- c. Koefisien determinasi, juga dikenal sebagai R-squared, mengukur sejauh mana variasi nilai target dapat dijelaskan oleh model. R-squared berkisar antara 0 hingga 1, di mana nilai 1 menunjukkan bahwa model secara sempurna menjelaskan variasi nilai target, sementara nilai 0 menunjukkan bahwa model tidak menjelaskan variasi sama sekali. R-squared dapat memberikan informasi tentang kualitas model dalam menjelaskan data. Semakin dekat nilai R-squared dengan 1, semakin baik kualitas prediksi model.

Rumus Koefisien Determinasi (R-squared):

$$R\text{-squared} = 1 - (SSR/SST)$$

di mana:

SSR (Sum of Squared Residuals) = jumlah kuadrat selisih antara nilai sebenarnya dan nilai prediksi ($\hat{y}_i - y_i$)²

SST (Total Sum of Squares) = jumlah kuadrat selisih antara nilai sebenarnya dan rata-rata nilai sebenarnya ($y_i - \bar{y}$)²

R-squared mengukur sejauh mana variasi nilai target dapat dijelaskan oleh model. Nilai 1 menunjukkan bahwa model secara sempurna menjelaskan variasi

nilai target, sedangkan nilai 0 menunjukkan bahwa model tidak menjelaskan variasi sama sekali.

83 Penerapan KNN Dalam Berbagai Bidang

Contoh penerapan KNN dalam berbagai bidang, seperti kesehatan, keuangan, atau teknologi. Berikut adalah contoh kasus penerapan KNN dalam masing-masing bidang:

8.3.1 Kesehatan

Dalam bidang kesehatan, KNN dapat digunakan dalam Sistem Pakar untuk mendiagnosis penyakit berdasarkan gejala yang dialami oleh pasien. Misalnya, dalam kasus pengenalan penyakit pada kulit, sistem dapat menggunakan data gejala pasien dan data gejala penyakit yang telah diklasifikasikan sebelumnya. KNN dapat digunakan untuk mencari kelompok gejala yang serupa dari pasien lain dalam dataset dan memberikan prediksi mengenai penyakit yang mungkin diderita oleh pasien berdasarkan gejala yang mirip dengan tetangga terdekatnya.

8.3.2 Keuangan

Dalam bidang keuangan, KNN dapat diterapkan dalam Sistem Pakar untuk analisis risiko dan prediksi harga saham. Misalnya, dengan menggunakan data historis harga saham dan faktor-faktor lain seperti volume perdagangan, tingkat suku bunga, dan indeks pasar, Sistem Pakar dapat melatih model KNN untuk memprediksi pergerakan harga saham di masa depan. Model KNN akan mencari tetangga terdekat dari data baru yang mencerminkan kondisi pasar saat ini dan memberikan prediksi harga saham berdasarkan harga tetangga terdekat tersebut.

8.3.3 Teknologi

Dalam bidang teknologi, KNN dapat digunakan dalam Sistem Pakar untuk klasifikasi objek berdasarkan fitur atau atributnya. Misalnya, dalam pengenalan pola suara atau

pengenalan tulisan tangan, Sistem Pakar dapat melatih model KNN dengan dataset yang berisi pola suara atau tulisan tangan yang sudah diklasifikasikan sebelumnya. Kemudian, model KNN dapat digunakan untuk mengklasifikasikan objek suara atau tulisan tangan yang baru berdasarkan kesamaan fitur dengan tetangga terdekat dalam dataset.

Pada ketiga contoh di atas, penerapan KNN dalam Sistem Pakar memungkinkan sistem untuk memanfaatkan pengetahuan dari data yang ada untuk membuat prediksi atau mengambil keputusan dalam bidang kesehatan, keuangan, atau teknologi.

DAFTAR PUSTAKA

- Ali, N., Neagu, D. dan Trundle, P. 2019. "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, 1, hal. 1–15.
- Bansal, M., Goyal, A. dan Choudhary, A. 2022. "A comparative analysis of K-Nearest Neighbour, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning," *Decision Analytics Journal*, hal. 100071.
- Chen, W. *et al.* 2021. "Diagnosis of wind turbine faults with transfer learning algorithms," *Renewable Energy*, 163, hal. 2053–2067.
- Chen, X., Jia, S. dan Xiang, Y. 2020. "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, 141, hal. 112948.
- Cubillos, M., Wøhlk, S. dan Wulff, J.N. 2022. "A bi-objective k-nearest-neighbors-based imputation method for multilevel data," *Expert Systems with Applications*, 204, hal. 117298.
- Deng, S. *et al.* 2023. "Non-parametric Nearest Neighbor Classification Based on Global Variance Difference," *International Journal of Computational Intelligence Systems*, 16(1), hal. 26.
- Dridi, S. 2021. "Supervised Learning-A Systematic Literature Review."
- Kang, S. 2021. "K-nearest neighbor learning with graph neural networks," *Mathematics*, 9(8), hal. 830.
- Mardani, A. *et al.* 2019. "Application of decision making and fuzzy sets theory to evaluate the healthcare and medical problems: a review of three decades of research with recent developments," *Expert Systems with Applications*, 137, hal. 202–231.

- Medsker, L.R. dan Bailey, D.L. 2020. "Models and guidelines for integrating expert systems and neural networks," in *Hybrid architectures for intelligent systems*. CRC Press, hal. 153–171.
- Nabipour, M. *et al.* 2020. "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, 8, hal. 150199–150212.
- Pramanik, M.I. *et al.* 2020. "Healthcare informatics and analytics in big data," *Expert Systems with Applications*, 152, hal. 113388.
- Putra, P.H., Novelan, M.S. dan Rizki, M. 2022. "Analysis K-Nearest Neighbor Method in Classification of Vegetable Quality Based on Color," *Journal of Applied Engineering and Technological Science (JAETS)*, 3(2), hal. 126–132.
- Sarker, I.H. *et al.* 2020. "K-nearest neighbor learning based diabetes mellitus prediction and analysis for eHealth services," *EAI Endorsed Transactions on Scalable Information Systems*, 7(26), hal. e4--e4.
- Sen, P.C., Hajra, M. dan Ghosh, M. 2020. "Supervised classification algorithms in machine learning: A survey and review," in *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, hal. 99–111.
- Varlamov, O. 2021. "'Brains' for Robots: Application of the Mivar Expert Systems for Implementation of Autonomous Intelligent Robots," *Big Data Research*, 25, hal. 100241.
- Zhang, C. dan Lu, Y. 2021. "Study on artificial intelligence: The state of the art and future prospects," *Journal of Industrial Information Integration*, 23, hal. 100224.

BAB 9

PENERAPAN ALGORITMA NAIVE BAYES DALAM SISTEM PAKAR

Oleh Edison Ulung Panggabean Simanihuruk

9.1 Pendahuluan

Sistem pakar adalah program komputer yang dibuat untuk meniru kemampuan penalaran manusia dalam memecahkan masalah atau memberikan solusi. Salah satu metode yang dapat digunakan dalam sistem pakar adalah algoritma Naive Bayes, yang merupakan algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan prediksi.

Buku ini akan membahas tentang penerapan algoritma Naive Bayes dalam sistem pakar. Pembahasan akan dimulai dengan pengenalan konsep dasar sistem pakar dan algoritma Naive Bayes. Kemudian akan dijelaskan langkah-langkah implementasi algoritma Naive Bayes dalam sistem pakar dan bagaimana algoritma ini dapat digunakan untuk memecahkan masalah pada beberapa bidang seperti kesehatan, keuangan, dan sebagainya.

Sistem Pakar terdiri dari beberapa komponen yang berfungsi untuk membantu sistem dalam menyelesaikan masalah atau memberikan solusi terhadap suatu permasalahan yang diberikan (Ahyuna & Aryasa, 2017). Berikut adalah beberapa komponen Sistem Pakar:

1. Basis pengetahuan (*Knowledge Base*)

Basis Pengetahuan adalah komponen utama dalam Sistem Pakar yang menyimpan informasi, pengetahuan, dan pengalaman dari para pakar dalam bidang tertentu. Basis Pengetahuan dapat

berupa database atau file teks yang berisi himpunan fakta, aturan, prosedur, dan kebijakan yang digunakan oleh Sistem Pakar dalam melakukan penalaran dan memberikan solusi.

2. Mesin inferensi (*Inference Engine*)

Mesin Inferensi adalah komponen dalam Sistem Pakar yang bertugas melakukan penalaran atau proses berpikir secara logika dengan menggunakan aturan-aturan yang telah disimpan dalam Basis Pengetahuan. Mesin Inferensi dapat mengevaluasi fakta-fakta yang diberikan dan menghasilkan suatu kesimpulan atau rekomendasi berdasarkan pengetahuan yang tersimpan dalam Basis Pengetahuan.

3. Antarmuka pengguna (*User Interface*)

Antarmuka Pengguna adalah komponen dalam Sistem Pakar yang berfungsi sebagai jembatan antara pengguna dengan sistem. Antarmuka Pengguna dapat berupa tampilan grafis atau bahasa alami yang memungkinkan pengguna untuk berinteraksi dengan Sistem Pakar dengan mudah dan intuitif.

4. Basis kasus (*Case Base*)

Basis Kasus adalah komponen dalam Sistem Pakar yang digunakan untuk menyimpan informasi tentang kasus-kasus yang telah terjadi sebelumnya. Basis Kasus dapat digunakan sebagai sumber informasi untuk membantu Sistem Pakar dalam menyelesaikan masalah yang serupa.

5. Sistem pelaporan (*Explanation Facility*)

Sistem Pelaporan adalah komponen dalam Sistem Pakar yang memberikan penjelasan atau justifikasi atas keputusan atau rekomendasi yang dihasilkan oleh sistem. Sistem Pelaporan dapat membantu pengguna dalam memahami proses berpikir yang dilakukan oleh Sistem Pakar dalam mengambil keputusan atau memberikan solusi.

Keseluruhan komponen Sistem Pakar saling terintegrasi untuk membentuk sebuah sistem yang dapat memberikan solusi

atau rekomendasi dalam situasi yang memerlukan pengetahuan khusus. Dengan memanfaatkan pengetahuan dan pengalaman para ahli, Sistem Pakar dapat membantu organisasi atau perusahaan untuk meningkatkan efisiensi dan kualitas proses bisnis, meningkatkan kecepatan pengambilan keputusan, dan meminimalkan risiko kesalahan dalam pengambilan keputusan (Hakam Febtadianrano Putro, Retno Tri Vlandari, 2020).

9.2 Pengenalan Sistem Pakar

Sistem Pakar adalah program komputer yang dirancang untuk meniru kemampuan penalaran dan keahlian dalam memecahkan masalah yang biasanya dimiliki oleh seorang pakar manusia di bidang tertentu. Sistem Pakar dapat diartikan sebagai sistem yang memanfaatkan pengetahuan dan pengalaman ahli dalam bidang tertentu untuk mengambil keputusan atau memberikan saran dalam situasi yang memerlukan pengetahuan khusus (Wulandari, Novi Yohanes, 2018).

Tujuan utama dari Sistem Pakar adalah memberikan solusi atau jawaban yang cepat dan akurat dalam bidang tertentu. Sistem Pakar dapat digunakan dalam berbagai bidang seperti kesehatan, keuangan, manufaktur, dan banyak lagi.

1. Defenisi algoritma Neive Bayes

Algoritma Naive Bayes adalah salah satu algoritma pembelajaran mesin yang digunakan dalam klasifikasi dan prediksi. Algoritma ini didasarkan pada teorema Bayes dan diasumsikan bahwa setiap *fitur* dalam data independen satu sama lain. Meskipun asumsi ini jarang terpenuhi dalam dunia nyata, algoritma Naive Bayes masih sering digunakan karena kecepatan komputasi yang tinggi dan kinerja yang baik dalam banyak aplikasi.

2. Konsep dasar

a. Teorema Bayes

Algoritma Naive Bayes menggunakan teorema Bayes untuk menghitung probabilitas suatu kelas berdasarkan probabilitas *fitur-fitur* yang terkait. Teorema Bayes menyatakan bahwa probabilitas suatu hipotesis (kelas) berdasarkan bukti (*fitur-fitur*) dapat dihitung dengan memperhitungkan probabilitas awal hipotesis dan kemungkinan bukti yang diamati.

b. Asumsi Naive Bayes

Algoritma Naive Bayes mengasumsikan bahwa setiap *fitur* dalam data independen secara kondisional terhadap kelas tertentu. Dengan kata lain, *fitur-fitur* dianggap saling bebas jika diketahui kelasnya. Meskipun asumsi ini jarang benar dalam dunia nyata, algoritma ini tetap dapat memberikan hasil yang baik dalam banyak kasus.

c. Probabilitas dan klasifikasi

Algoritma Naive Bayes menggunakan probabilitas untuk melakukan klasifikasi. Diberikan *fitur-fitur input*, algoritma menghitung probabilitas setiap kelas untuk menentukan kelas yang paling mungkin. Probabilitas dihitung berdasarkan probabilitas kelas awal dan probabilitas kondisional *fitur-fitur* yang diamati.

d. Estimasi parameter

Untuk menggunakan algoritma Naive Bayes, parameter-parameter probabilitas perlu diestimasi dari data pelatihan. Estimasi ini melibatkan menghitung probabilitas awal kelas dan probabilitas kondisional *fitur-fitur* yang terkait dengan kelas.

e. Smoothing

Dalam kasus di mana tidak ada data pelatihan yang mengandung suatu kombinasi *fitur* dan kelas tertentu, algoritma Naive Bayes dapat mengalami masalah nol

probabilitas. Untuk mengatasi masalah ini, teknik smoothing dapat digunakan untuk memberikan probabilitas non-nol dengan melakukan penyesuaian pada perkiraan probabilitas.

Algoritma Naive Bayes dapat diterapkan dalam berbagai bidang seperti klasifikasi teks, analisis sentimen, deteksi *spam*, diagnosis medis, dan banyak lagi. Meskipun asumsi naifnya, algoritma ini sering kali memberikan hasil yang baik dan menjadi salah satu algoritma klasifikasi yang populer dalam pembelajaran mesin (Rizqiyan et al., 2017).

3. Kelebihan dan Kekurangan Algoritma Naive Bayes

a. Kelebihan dan Kekurangan Algoritma Naive Bayes

Sederhana dan Mudah diimplementasikan : Algoritma Naive Bayes relatif sederhana dan mudah dipahami. Itu dapat dengan cepat diimplementasikan dengan *dataset* yang besar.

Efisiensi Komputasi : Algoritma Naive Bayes memiliki kecepatan komputasi yang tinggi. Karena asumsi independensi variabel, perhitungan probabilitas dapat dilakukan secara paralel, sehingga algoritma ini cocok untuk *dataset* dengan *fitur* yang besar.

Skalabilitas yang baik : Algoritma Naive Bayes cenderung memiliki kinerja yang baik pada *dataset* yang besar dan dapat dengan mudah mengatasi dimensi tinggi karena perhitungan probabilitas dilakukan secara independen pada setiap *fitur*.

Efektif pada Data yang Kurang : Algoritma Naive Bayes dapat memberikan hasil yang baik bahkan ketika jumlah data pelatihan terbatas. Algoritma ini masih dapat memberikan estimasi probabilitas dengan baik meskipun ada beberapa data yang hilang atau tidak lengkap.

Tidak sensitif terhadap data yang irrelevant : Algoritma Naive Bayes tidak terlalu dipengaruhi oleh adanya *fitur-fitur* yang tidak relevan atau *redundant* dalam *dataset*. Karena itu,

algoritma ini dapat memberikan hasil yang baik bahkan jika ada beberapa *fitur* yang tidak memberikan kontribusi signifikan pada klasifikasi.

b. Kekurangan algoritma Naive Bayes

Asumsi independensi yang kuat : Asumsi independensi variabel pada Naive Bayes sering kali tidak terpenuhi dalam dunia nyata. *Fitur-fitur* dalam data sering kali saling terkait, dan asumsi ini dapat mengurangi kinerja algoritma.

Ketidakmampuan menangani hubungan *Non linear* : Algoritma Naive Bayes tidak dapat menangani hubungan nonlinear antara *fitur-fitur* dalam data. Jika ada hubungan nonlinier yang signifikan, algoritma ini mungkin menghasilkan kinerja yang kurang akurat.

Sensitif terhadap data pemisahan kelas yang buruk: Jika data pemisahan kelas tidak optimal, yaitu jika terdapat tumpang tindih antara kelas-kelas, maka algoritma Naive Bayes dapat menghasilkan kinerja yang buruk.

Kurangnya pengelompokan informasi : Algoritma Naive Bayes cenderung mengasumsikan bahwa setiap *fitur* memiliki kontribusi yang sama terhadap klasifikasi. Oleh karena itu, algoritma ini tidak efektif dalam mengelompokkan *fitur-fitur* yang saling bergantung secara kualitatif (Chatrina, Siregar et al., 2020).

Ketidakmampuan menangani data kontinu : Naive Bayes secara alami lebih cocok untuk data diskrit. Ketika dihadapkan pada data kontinu, diperlukan asumsi tambahan seperti menggunakan distribusi probabilitas atau metode diskritisasi, yang dapat mempengaruhi kinerja algoritma.

9.3 Penerapan Algoritma Neive Bayes dalam Sistem Pakar.

Algoritma Naive Bayes adalah algoritma klasifikasi yang berdasarkan pada Teorema Bayes dengan asumsi bahwa *fitur-fitur* yang digunakan dalam klasifikasi adalah saling independen. Algoritma ini sering digunakan dalam sistem pakar untuk mengklasifikasikan data ke dalam kategori atau kelas yang sesuai. Berikut adalah langkah-langkah umum untuk menerapkan algoritma Naive Bayes dalam sistem pakar (Chatrina, Siregar et al., 2020):

1. Pilih dan siapkan *dataset* : Pertama, Anda perlu memilih dan menyiapkan *dataset* yang akan digunakan dalam sistem pakar. *Dataset* ini harus berisi contoh-contoh yang telah diklasifikasikan ke dalam kategori yang sesuai.
2. Ekstraksi *fitur* : Set(Ginting & Trinanda, 2014)elah *dataset* dipersiapkan, langkah berikutnya adalah mengekstraksi *fitur-fitur* yang relevan dari setiap contoh dalam . *Fitur-fitur* ini dapat berupa atribut-atribut yang berhubungan dengan masalah yang ingin Anda pecahkan.
3. Pelatihan model : Selanjutnya, Anda perlu melatih model Naive Bayes menggunakan *dataset* yang telah dipersiapkan. Pada tahap ini, model akan menghitung probabilitas kelas dan probabilitas *fitur-fitur* yang terkait dengan setiap kelas.
4. Pengujian model : Setelah model dilatih, Anda dapat menguji model dengan memberikan contoh-contoh baru yang belum diklasifikasikan. Model akan menggunakan probabilitas yang telah dihitung sebelumnya untuk mengklasifikasikan contoh-contoh baru ke dalam kategori yang sesuai.
5. Evaluasi performa : Tahap terakhir adalah mengevaluasi performa model Naive Bayes dalam sistem pakar Anda. Anda dapat menggunakan metrik evaluasi seperti akurasi, presisi, *recall*, atau *F1-score* untuk menilai sejauh mana model dapat melakukan klasifikasi dengan benar.

Penerapan algoritma Naive Bayes dalam sistem pakar dapat membantu dalam memecahkan masalah klasifikasi, seperti dalam pengenalan pola, deteksi *spam*, analisis sentimen, atau diagnosis medis. Namun, perlu diingat bahwa asumsi independensi pada algoritma Naive Bayes tidak selalu terpenuhi dalam setiap kasus, sehingga kemungkinan terdapat penyesuaian yang perlu dilakukan tergantung pada masalah yang sedang Anda hadapi (Hakam Febtadianrano Putro, Retno Tri Vulandari, 2020).

1. Langkah-langkah implementasi

Berikut adalah langkah-langkah implementasi yang lebih rinci untuk menerapkan algoritma Naive Bayes dalam sistem pakar:

- a. Siapkan lingkungan pengembangan: Pertama, pastikan Anda memiliki lingkungan pengembangan yang sesuai, seperti Python dengan library SciKit-Learn atau R dengan library e1071. Instal library yang diperlukan jika belum terpasang.
- b. Pilih dan siapkan *dataset* : Pilih *dataset* yang relevan dengan masalah yang ingin Anda pecahkan. Pastikan *dataset* tersebut memiliki atribut-atribut yang dapat digunakan sebagai *fitur* dalam klasifikasi. Bagi *dataset* menjadi dua *subset* : *subset* pelatihan (*training set*) dan *subset* pengujian (*testing set*).
- c. *Preprocessing* data : Lakukan langkah-langkah *preprocessing* data jika diperlukan. Ini dapat mencakup menghapus data yang tidak lengkap atau tidak relevan, melakukan normalisasi atau penskalaan pada atribut numerik, atau mengubah atribut kategorikal menjadi representasi numerik yang sesuai.
- d. Ekstraksi *fitur* : Identifikasi dan ekstraksi *fitur-fitur* yang relevan dari *dataset*. Misalnya, jika Anda ingin mengklasifikasikan email sebagai *spam* atau bukan *spam*, *fitur-fitur* dapat mencakup kata-kata kunci, panjang email, atau keberadaan tautan.

- e. Pelatihan model Naive Bayes : Gunakan *subset* pelatihan (*training set*) untuk melatih model Naive Bayes. Sesuaikan *dataset* ke dalam model dengan menghitung probabilitas kelas dan probabilitas *fitur-fitur* yang terkait dengan setiap kelas. Terdapat beberapa variasi Naive Bayes, seperti Naive Bayes Gaussian untuk atribut numerik dan Naive Bayes Multinomial atau Bernoulli untuk atribut kategorikal.
- f. Pengujian model: Gunakan subset pengujian (*testing set*) untuk menguji performa model yang telah dilatih. Berikan contoh-contoh baru kepada model dan amati klasifikasinya. Hitung metrik evaluasi, seperti akurasi, presisi, recall, atau F1-score, untuk mengevaluasi performa model.
- g. Evaluasi dan penyesuaian : Evaluasi hasil pengujian dan analisis performa model Naive Bayes. Jika hasilnya belum memuaskan, Anda dapat mencoba melakukan penyesuaian, seperti mencoba *fitur-fitur* yang berbeda atau menggunakan teknik seleksi *fitur* untuk meningkatkan performa model.
- h. Implementasi dalam sistem pakar : Setelah model Naive Bayes menghasilkan hasil yang baik, sehingga anda dapat mengimplementasikannya dalam sistem pakar. Pastikan algoritma dan model yang telah dilatih terintegrasi dengan baik dalam sistem, sehingga dapat digunakan untuk memberikan klasifikasi yang akurat kepada pengguna.
- i. Pemeliharaan dan pengoptimalan : Tetap lakukan pemeliharaan dan pengoptimalan sistem pakar dengan Naive Bayes. Perbarui model dan *dataset* secara berkala untuk menjaga performa sistem dan memperbaiki keakuratan klasifikasi seiring waktu.

j. Ingatlah bahwa langkah-langkah ini hanya memberikan panduan umum, dan implementasi yang sebenarnya akan bervariasi tergantung pada bahasa pemrograman, library yang digunakan, dan masalah spesifik yang sedang Anda hadapi. Seperti contoh gambar Dibawah untuk langkah-langkah sistem pakar dengan metode Naive Bayes. Perhitungan Neive Bayes yang digunakan :

Menghitung $P(a_i|v_j)$ dengan rumus :

$$P(a_i|v_j) = \frac{n_{c_i+m.p}}{n+m} \dots\dots\dots(1)$$

Dimana :

N_c = nilai data *record* pada data training yag $v = v_j$

$a = a_i$

p = 1/banyaknya jenis kelas/penyakit

m = jumlah parameter/gejala

n = nilai data *record* pada data training yang $v = v_j$ /tiap kelas

Persamaan diselesaikan melalui perhitungan sebagai berikut :

1. Menentukan nilai n_c untuk setiap class/kelas
2. Menghitung nilai $P(a_i|v_j)$ dan menghitung nilai $P(v_j)$,
dimana : $P(a_i|v_j) = \frac{n_{c_i+m.p}}{n+m}$ dan nilai $P(v_j) = P(v_j) = \frac{n}{m}$
3. Menghitung $P(a_i|v_j) \cdot P(v_j)$ untuk tiap v
4. Menentukan hasil klassifikasi yaitu v yang memiliki hasil perkalian terbesar.

Langkah-Langkah sistem pakar metode Naïve Bayes

2. Preprocessing data

Preprocessing data merupakan langkah penting sebelum melatih model Naive Bayes dalam sistem pakar. Tujuan dari *preprocessing* data adalah untuk mempersiapkan data agar sesuai dengan kebutuhan model dan memaksimalkan performa

klasifikasi. Berikut adalah beberapa langkah umum dalam *preprocessing* data:

- a. Menghapus data yang tidak lengkap atau tidak relevan: Identifikasi dan hapus data yang tidak lengkap, seperti entri yang memiliki nilai kosong atau hilang. Juga, jika terdapat atribut yang tidak relevan atau tidak memberikan informasi yang berarti dalam klasifikasi, Anda dapat menghapusnya.
- b. Normalisasi atau penskalaan data: Jika terdapat atribut numerik dengan skala yang berbeda, normalisasi atau penskalaan dapat membantu agar atribut-atribut tersebut memiliki rentang nilai yang serupa. Misalnya, Anda dapat menggunakan metode normalisasi min-max untuk mengubah nilai-nilai atribut ke dalam rentang antara 0 dan 1.
- c. Mengatasi *outlier*: *Outlier* adalah nilai yang jauh berbeda dari nilai-nilai lain dalam atribut. *Outlier* dapat mempengaruhi model secara negatif. Jika terdapat *outlier* dalam atribut numerik, Anda dapat memilih untuk menghapusnya atau menggunakan metode penggantian nilai *outlier* dengan nilai tengah (median) atau nilai rata-rata.
- d. Mengubah atribut kategorikal menjadi representasi numerik: Jika terdapat atribut kategorikal, seperti jenis kelamin atau status pernikahan, Anda perlu mengubahnya menjadi representasi numerik agar dapat dimanfaatkan oleh model Naive Bayes. Misalnya, Anda dapat menggunakan metode "*one-hot encoding*" untuk mengubah setiap nilai kategorikal menjadi atribut biner baru.
- e. Menangani data yang tidak seimbang: Jika terdapat ketidakseimbangan dalam jumlah sampel antara kelas-kelas yang ingin Anda klasifikasikan, seperti kelas positif

dan kelas negatif, Anda perlu menangani hal ini agar model tidak condong pada satu kelas yang dominan. Metode *oversampling* (menambahkan sampel ke kelas minoritas) atau *undersampling* (mengurangi sampel dari kelas mayoritas) dapat digunakan untuk mengatasi masalah ini.

- f. Penggunaan teknik seleksi *fitur* : Jika *dataset* memiliki banyak *fitur*, teknik seleksi *fitur* dapat membantu memilih *fitur-fitur* yang paling relevan atau memberikan kontribusi terbesar dalam klasifikasi. Hal ini dapat mengurangi dimensi data dan meningkatkan efisiensi serta akurasi model.
- g. Pembagian *dataset* menjadi *subset* pelatihan dan pengujian: Setelah melakukan *preprocessing* data, bagi *dataset* menjadi *subset* pelatihan (*training set*) dan subset pengujian (*testing set*). *Training set* digunakan untuk melatih model Naive Bayes, sedangkan *testing set* digunakan untuk menguji performa model.

3. Pembuatan model Naive Bayes

Untuk membuat model Naive Bayes dalam sistem pakar, Anda dapat mengikuti langkah-langkah berikut:

- a. Pilih tipe Naive Bayes yang sesuai : Terdapat beberapa variasi Naive Bayes yang dapat Anda pilih, tergantung pada jenis atribut yang anda miliki dalam *dataset* anda. Beberapa tipe Naive Bayes umum meliputi:
 - Naive Bayes Gaussian : Digunakan untuk atribut numerik yang mengikuti distribusi Gaussian (normal).
 - Naive Bayes Multinomial: Digunakan untuk atribut diskrit dengan frekuensi hitungan, seperti jumlah kata dalam dokumen.
 - Naive Bayes Bernoulli : Digunakan untuk atribut biner atau atribut diskrit yang menggambarkan keberadaan

atau ketidakhadiran suatu *fitur*, seperti kategori ya/tidak.

b. Pelatihan model Naive Bayes :

- Naive Bayes Gaussian : Untuk setiap kelas, hitung rata-rata dan simpangan baku atribut numerik. Ini melibatkan menghitung rata-rata dan simpangan baku untuk setiap atribut pada setiap kelas.
- Naive Bayes Multinomial : Untuk setiap kelas, hitung frekuensi kemunculan setiap atribut diskrit dan frekuensi total atribut dalam setiap kelas.
- Naive Bayes Bernoulli: Untuk setiap kelas, hitung frekuensi keberadaan dan ketidakhadiran setiap atribut biner.

c. Menghitung probabilitas prior: Hitung probabilitas prior untuk setiap kelas dengan membagi jumlah sampel dalam kelas tersebut dengan jumlah total sampel.

d. Menghitung probabilitas *fitur* :

- Naive Bayes Gaussian : Untuk setiap atribut numerik dalam setiap kelas, hitung probabilitas distribusi Gaussian menggunakan rata-rata dan simpangan baku yang telah dihitung sebelumnya.
- Naive Bayes Multinomial : Untuk setiap atribut diskrit dalam setiap kelas, hitung probabilitas distribusi multinomial dengan membagi frekuensi kemunculan atribut dengan frekuensi total atribut dalam kelas tersebut.
- Naive Bayes Bernoulli : Untuk setiap atribut biner dalam setiap kelas, hitung probabilitas distribusi Bernoulli dengan membagi frekuensi keberadaan atribut dengan frekuensi total atribut dalam kelas tersebut.

e. Mengklasifikasikan data baru :

- Terapkan rumus teorema Bayes untuk menghitung probabilitas kelas untuk data baru berdasarkan probabilitas prior dan probabilitas *fitur* yang telah dihitung sebelumnya.
- Pilih kelas dengan probabilitas tertinggi sebagai prediksi kelas untuk data baru.

4. Evaluasi model Naive Bayes

Evaluasi model Naive Bayes dilakukan untuk mengukur sejauh mana model dapat melakukan klasifikasi dengan benar. Berikut adalah beberapa metrik evaluasi yang umum digunakan untuk mengevaluasi model Naive Bayes:

- a. Akurasi (*Accuracy*): Akurasi mengukur sejauh mana model dapat mengklasifikasikan data dengan benar. Akurasi dihitung dengan membagi jumlah prediksi yang benar dengan total jumlah sampel.
- b. Presisi (*Precision*): Presisi mengukur sejauh mana prediksi positif model benar. Presisi dihitung dengan membagi jumlah prediksi positif yang benar dengan total jumlah prediksi positif.
- c. Recall (*Sensitivitas* atau *True Positive Rate*): *Recall* mengukur sejauh mana model dapat mengidentifikasi secara benar contoh-contoh positif. *Recall* dihitung dengan membagi jumlah prediksi positif yang benar dengan total jumlah contoh positif dalam *dataset*.
- d. F1-score: F1-score adalah ukuran yang menggabungkan presisi dan *recall*. F1-score dihitung dengan mengambil rata-rata harmonik antara presisi dan recall. Ini memberikan nilai yang seimbang antara presisi dan recall.
- e. *Confusion Matrix*: *Confusion matrix* adalah tabel yang menunjukkan jumlah prediksi yang benar dan salah dalam setiap kelas. *Confusion matrix* dapat digunakan

untuk menghitung metrik evaluasi seperti akurasi, presisi, dan *recall*.

- f. *Cross Validation* : *Cross validation* adalah teknik untuk menguji dan mengevaluasi model dengan membagi *dataset* menjadi *subset* pelatihan dan pengujian secara berulang-ulang. Ini membantu dalam memperoleh estimasi yang lebih stabil tentang performa model.

9.4 Studi Kasus

Berikut adalah studi kasus Sistem Pakar (Chatrina, Siregar et al., 2020).

1. Study kasus Sistem pakar yang akan dibangun bertujuan untuk mendeteksi apakah sebuah email termasuk dalam kategori *spam* atau bukan *spam* berdasarkan kontennya.

Langkah-langkah:

- a. Persiapan *Dataset*: Kumpulkan *dataset* email yang sudah diklasifikasikan sebagai *spam* atau bukan *spam*. *Dataset* ini harus berisi atribut-atribut yang relevan, seperti isi email, pengirim, subjek, dll. Pastikan *dataset* terdiri dari sampel-sampel yang sudah diberi label kelas (*spam* atau bukan *spam*).
- b. *Preprocessing* Data: Lakukan preprocessing data pada *dataset* untuk membersihkan dan mempersiapkannya sebelum digunakan. Ini mungkin melibatkan langkah-langkah seperti menghapus karakter khusus, mengubah teks menjadi huruf kecil, menghapus *stopword*, dan melakukan *stemming* atau lemmatisasi pada kata-kata.
- c. Ekstraksi *Fitur* : Ekstraksi *fitur* adalah langkah penting dalam mempersiapkan data untuk model Naive Bayes. Dalam kasus ini, *fitur-fitur* yang relevan dapat mencakup kata-kata kunci dalam email, jumlah kata, keberadaan

tautan, dan atribut-atribut lain yang dapat membantu membedakan *spam* dan bukan *spam*.

- d. Pembagian *dataset* : Bagi *dataset* menjadi *subset* pelatihan dan pengujian. *Subset* pelatihan akan digunakan untuk melatih model Naive Bayes, sedangkan *subset* pengujian akan digunakan untuk menguji performa model.
- e. Pelatihan model Naive Bayes: Gunakan subset pelatihan untuk melatih model Naive Bayes. Dalam kasus ini, Anda akan menggunakan Naive Bayes Multinomial karena *fitur-fitur* yang diekstraksi adalah atribut diskrit. Hitung probabilitas prior untuk setiap kelas dan probabilitas *fitur-fitur* yang terkait dengan setiap kelas.
- f. Evaluasi model : Gunakan subset pengujian untuk menguji performa model yang telah dilatih. Hitung metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1-score* untuk mengevaluasi sejauh mana model dapat membedakan antara *spam* dan bukan *spam*. Analisis *confusion matrix* juga berguna untuk memahami prediksi model.
- g. Penyesuaian dan pemeliharaan : Jika performa model belum memuaskan, Anda dapat melakukan penyesuaian, seperti mencoba *fitur-fitur* yang berbeda, meningkatkan preprocessing data, atau menggunakan teknik seleksi *fitur*. Selain itu, penting untuk melakukan pemeliharaan dan pembaruan model secara berkala seiring dengan munculnya tren *spam* baru atau perubahan dalam *dataset*.
- h. Implementasi dalam sistem pakar : Setelah model Naive Bayes memberikan hasil yang memuaskan, integrasikan model ke dalam sistem pakar. Pastikan algoritma dan model terintegrasi dengan baik sehingga dapat memberikan klasifikasi yang akurat kepada pengguna.

2. Study kasus sistem pakar yang akan dibangun bertujuan untuk membantu dalam diagnosa penyakit berdasarkan gejala yang dialami oleh pasien (Ilia, 2015).

Langkah-langkah:

- a. Identifikasi penyakit dan gejala : Identifikasi penyakit yang ingin didiagnosis dan daftar gejala yang berhubungan dengan penyakit tersebut. Misalnya, penyakit diabetes dengan gejala seperti sering merasa haus, sering buang air kecil, dan penurunan berat badan.
- b. Kumpulkan basis pengetahuan : Kumpulkan basis pengetahuan yang terdiri dari aturan-aturan yang menjelaskan hubungan antara gejala-gejala dengan penyakit-penyakit yang mungkin. Misalnya, jika gejala A, B, dan C ada, maka kemungkinan terkena penyakit X tinggi.
- c. *Preprocessing* data: Lakukan *preprocessing* data pada basis pengetahuan jika diperlukan. Ini mungkin melibatkan penghapusan redundansi, normalisasi, atau pengubahan format data agar sesuai dengan format yang diperlukan oleh sistem pakar.
- d. Desain Arsitektur Sistem Pakar: Tentukan arsitektur sistem pakar yang akan digunakan. Ini dapat meliputi penggunaan aturan produksi (*rule-based*), jaringan saraf tiruan (*neural network*), atau algoritma Naive Bayes.
- e. Implementasi sistem pakar : Implementasikan basis pengetahuan dan algoritma di dalam sistem pakar. Pastikan sistem pakar dapat menerima masukan gejala dari pengguna dan menghasilkan diagnosis penyakit yang mungkin.
- f. Evaluasi dan validasi : Lakukan evaluasi dan validasi sistem pakar dengan menggunakan *dataset* pengujian yang sudah diketahui diagnosanya. Periksa apakah sistem

pakar memberikan hasil yang sesuai dengan yang diharapkan.

- g. Pemeliharaan dan peningkatan : Pemeliharaan dan pembaruan sistem pakar sangat penting untuk menjaga keakuratan dan relevansi sistem. Perbarui basis pengetahuan dengan penelitian terbaru atau perkembangan dalam bidang medis dan pastikan sistem tetap diperbarui secara berkala.
- h. Integrasi dengan sistem medis : Integrasikan sistem pakar ke dalam sistem medis yang ada. Sistem pakar dapat diintegrasikan dengan rekam medis elektronik (*Electronic Medical Records/EMR*) atau digunakan sebagai alat bantu bagi dokter dalam memberikan diagnosis dan rekomendasi pengobatan.

3. Tantangan dan Peluang dalam Penerapan Algoritma Naive Bayes dalam Sistem Pakar

Penerapan algoritma Naive Bayes dalam sistem pakar memiliki tantangan dan peluang tertentu. Berikut adalah beberapa tantangan dan peluang yang mungkin Anda temui (Ilia, 2015) :

a. Tantangan :

Ketergantungan *Fitur* : Algoritma Naive Bayes mengasumsikan independensi kondisional antara *fitur-fitur yang ada*. Namun, dalam beberapa kasus, *fitur-fitur tersebut mungkin* saling tergantung atau memiliki keterkaitan yang kompleks. Hal ini dapat mengurangi akurasi model Naive Bayes.

Data yang Tidak Seimbang : Jika *dataset* yang digunakan memiliki distribusi kelas yang tidak seimbang, di mana jumlah sampel di satu kelas jauh lebih banyak daripada kelas lainnya, ini dapat menyebabkan bias dalam model Naive Bayes. Model cenderung lebih condong ke kelas mayoritas

dan kurang mampu mengenali sampel-sampel dari kelas minoritas.

Data Kontinu : Algoritma Naive Bayes umumnya bekerja dengan baik pada atribut diskrit atau biner. Namun, jika *dataset* Anda memiliki atribut kontinu, seperti atribut numerik, Naive Bayes Gaussian dapat digunakan. Namun, dengan data yang sangat kontinu atau memiliki distribusi yang tidak normal, hasilnya mungkin tidak akurat.\

Asumsi Independensi : Algoritma Naive Bayes mengasumsikan independensi antara *fitur-fitur* yang ada. Ini dapat menjadi tantangan jika ada keterkaitan yang signifikan antara *fitur-fitur* dalam *dataset*. Jika asumsi ini tidak terpenuhi, hasil prediksi model Naive Bayes dapat menjadi tidak akurat.

b. Peluang :

Kecepatan dan Efisiensi: Algoritma Naive Bayes memiliki sifat komputasi yang cepat dan efisien. Ini membuatnya cocok untuk sistem pakar yang membutuhkan *respons* cepat dan dapat digunakan dalam waktu nyata.

Interpretabilitas : Naive Bayes adalah algoritma yang relatif sederhana dan mudah diinterpretasikan. Ini memungkinkan pengguna atau profesional medis untuk memahami dan menganalisis faktor-faktor yang berkontribusi pada prediksi dan keputusan yang dibuat oleh sistem pakar.

Skalabilitas : Naive Bayes dapat dengan mudah diterapkan pada *dataset* dengan jumlah *fitur* yang besar. Algoritma ini dapat bekerja dengan baik bahkan dengan *dataset* yang memiliki dimensi tinggi.

Akurasi yang Baik dalam Kasus yang Sesuai : Meskipun Naive Bayes memiliki asumsi yang sederhana, ia dapat memberikan hasil yang baik dalam kasus-kasus di mana asumsi independensi *fitur* dapat dipenuhi. Jika *fitur-fitur* dalam *dataset* benar-benar independen atau memiliki

keterkaitan yang lemah, Naive Bayes dapat menghasilkan prediksi yang akurat.

Dalam penerapan Naive Bayes pada sistem pakar, penting untuk mempertimbangkan tantangan yang ada dan memilih model yang sesuai dengan karakteristik data Anda. Jika tantangan tersebut dapat diatasi dan peluangnya dimanfaatkan, Naive Bayes dapat menjadi algoritma yang efektif untuk mendukung keputusan dalam sistem pakar.

9.5 Tantangan dalam Implementasi Algoritma

Naive Bayes Dalam Sistem Pakar

Implementasi algoritma Naive Bayes dalam sistem pakar dapat menghadapi beberapa tantangan. Berikut adalah beberapa tantangan yang mungkin dihadapi dalam implementasi tersebut :

1. Data yang tidak lengkap atau tidak konsisten : Sistem pakar membutuhkan data yang lengkap dan konsisten untuk melakukan pelatihan dan pengujian model Naive Bayes. Jika data yang digunakan tidak lengkap atau ada ketidaksesuaian dalam data yang ada, ini dapat mempengaruhi akurasi dan kualitas prediksi sistem pakar.
2. *Overfitting* atau *underfitting* : Naive Bayes dapat menghadapi masalah *overfitting* atau *underfitting* jika tidak dilakukan dengan baik. *Overfitting* terjadi ketika model terlalu cocok dengan data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Sedangkan *underfitting* terjadi ketika model terlalu sederhana dan tidak dapat menangkap pola yang kompleks dalam data. Dalam kedua kasus ini, performa model akan menurun.
3. Asumsi independensi yang Tidak Terpenuhi : Algoritma Naive Bayes mengasumsikan independensi antara *fitur-fitur* yang ada. Namun, dalam banyak kasus, *fitur-fitur* tersebut

mungkin saling terkait atau ada ketergantungan di antara mereka. Jika asumsi independensi tidak terpenuhi, performa model Naive Bayes dapat menurun.

4. Penanganan data kontinu : Naive Bayes umumnya cocok untuk data yang bersifat diskrit atau kategorikal. Jika *dataset* Anda memiliki atribut yang kontinu atau numerik, seperti usia atau suhu, diperlukan penanganan khusus. Anda dapat menggunakan teknik seperti binning atau menggunakan varian dari Naive Bayes seperti Naive Bayes Gaussian.
5. Tidak sesuai untuk kasus khusus : Algoritma Naive Bayes cenderung memberikan performa yang baik dalam kasus-kasus di mana independensi *fitur* dapat dianggap. Namun, dalam beberapa kasus kompleks, seperti klasifikasi teks dengan banyak *fitur* terkait atau dalam kasus dengan hubungan *non-linear* antara *fitur* dan kelas, Naive Bayes mungkin tidak memberikan performa yang optimal.
6. Kekurangan informasi kontekstual : Naive Bayes hanya mempertimbangkan frekuensi atau probabilitas setiap *fitur* secara terpisah. Ini berarti informasi kontekstual yang mungkin ada di antara *fitur-fitur* tersebut dapat terlewatkan. Hal ini dapat mempengaruhi kemampuan model Naive Bayes untuk menghasilkan prediksi yang tepat.
7. Menghadapi tantangan ini, penting untuk memahami batasan dan asumsi dari algoritma Naive Bayes dan melakukannya dengan hati-hati dalam implementasi sistem pakar. Selain itu, eksplorasi dan penggunaan teknik lain, seperti ensemble learning atau algoritma yang lebih kompleks, juga dapat membantu mengatasi beberapa tantangan tersebut dalam sistem pakar.

9.6 Peluang dan Potensi Pengembangan Algoritma Naive Bayes Dalam Sistem Pakar

Kesimpulan, penerapan algoritma Naive Bayes dalam sistem pakar memiliki tantangan dan peluang tertentu. Tantangan yang mungkin dihadapi meliputi ketergantungan *fitur*, data yang tidak seimbang, data kontinu, asumsi independensi yang tidak terpenuhi, dan kekurangan informasi kontekstual. Namun, ada juga peluang seperti kecepatan dan efisiensi, interpretabilitas, skalabilitas, dan akurasi yang baik dalam kasus yang sesuai.

Dalam mengatasi tantangan tersebut, penting untuk mempertimbangkan kualitas data, penanganan khusus untuk data kontinu, pemahaman terhadap asumsi dan batasan Naive Bayes, serta eksplorasi teknik lain jika diperlukan. Menggabungkan pengetahuan domain dengan pemilihan model yang tepat dan penyesuaian yang sesuai dapat meningkatkan kinerja sistem pakar yang menggunakan algoritma Naive Bayes (Rizqiyani et al., 2017). Namun, perlu diingat bahwa kesesuaian penggunaan algoritma Naive Bayes dalam sistem pakar tergantung pada kasus-kasus spesifik dan karakteristik data yang ada. Oleh karena itu, evaluasi yang cermat dan pemilihan algoritma yang sesuai dengan kebutuhan dan tujuan sistem pakar sangat penting untuk memastikan hasil yang akurat dan dapat diandalkan.

1. Ringkasan Pembahasan

Dalam pembahasan sebelumnya, telah dijelaskan tentang penerapan algoritma Naive Bayes dalam sistem pakar, langkah-langkah implementasi, preprocessing data, pembuatan model Naive Bayes, evaluasi model, dan studi kasus dalam bidang kesehatan.

Algoritma Naive Bayes dapat digunakan dalam sistem pakar untuk membantu dalam diagnosa penyakit berdasarkan gejala yang dialami oleh pasien. Langkah-langkah implementasi mencakup identifikasi penyakit dan gejala, pengumpulan basis pengetahuan, preprocessing data, desain arsitektur sistem

pakar, implementasi sistem pakar, evaluasi dan validasi, serta pemeliharaan dan peningkatan sistem.

Tantangan dalam implementasi algoritma Naive Bayes dalam sistem pakar meliputi ketergantungan *fitur*, data yang tidak seimbang, data kontinu, asumsi independensi yang tidak terpenuhi, dan kekurangan informasi kontekstual. Namun, ada juga peluang seperti kecepatan dan efisiensi, interpretabilitas, skalabilitas, dan akurasi yang baik dalam kasus yang sesuai.

Dalam menghadapi tantangan tersebut, penting untuk mempertimbangkan kualitas data, penanganan khusus untuk data kontinu, pemahaman terhadap asumsi dan batasan Naive Bayes, serta eksplorasi teknik lain jika diperlukan. Evaluasi yang cermat dan pemilihan algoritma yang sesuai dengan kebutuhan sistem pakar juga penting untuk mencapai hasil yang akurat dan dapat diandalkan.

Dengan pemahaman yang baik tentang penerapan algoritma Naive Bayes dalam sistem pakar dan kesesuaian dengan kasus yang ada, sistem pakar dapat menjadi alat yang berguna dalam bidang kesehatan untuk memberikan diagnosa penyakit dan rekomendasi pengobatan kepada pengguna.

2. Saran untuk Pengembangan Selanjutnya

Berikut adalah beberapa saran untuk pengembangan selanjutnya dalam penerapan algoritma Naive Bayes dalam sistem pakar:

- a. **Kualitas Data:** Perhatikan kualitas data yang digunakan dalam sistem pakar. Pastikan data lengkap, akurat, dan terstruktur dengan baik. Lakukan validasi dan pembersihan data untuk menghindari masalah seperti data yang hilang, outliers, atau inkonsistensi.
- b. **Perluas Basis Pengetahuan:** Tingkatkan basis pengetahuan sistem pakar dengan mengumpulkan lebih banyak data dan informasi terkait penyakit dan gejala yang ingin didiagnosis. Dengan memiliki basis

pengetahuan yang lebih luas, sistem pakar dapat memberikan diagnosis yang lebih akurat dan dapat diandalkan.

- c. Integrasi Teknik Data Mining: Selain menggunakan algoritma Naive Bayes, pertimbangkan untuk mengintegrasikan teknik data mining lainnya, seperti pohon keputusan (decision tree), jaringan saraf tiruan (neural networks), atau ensemble learning. Menggabungkan beberapa algoritma dapat meningkatkan performa sistem pakar dan memberikan hasil yang lebih baik.
- d. Penanganan Data Kontinu yang Lebih Lanjut: Jika *dataset* Anda memiliki atribut yang kontinu atau numerik, eksplorasi lebih lanjut tentang teknik yang tepat untuk menangani data tersebut. Selain Naive Bayes Gaussian, Anda juga dapat mempertimbangkan menggunakan metode seperti Naive Bayes berdasarkan kernel atau melakukan discretization pada data tersebut.
- e. Integrasi Teknologi NLP: Jika sistem pakar Anda berfokus pada diagnosa penyakit berdasarkan teks atau data medis, pertimbangkan untuk mengintegrasikan teknologi pemrosesan bahasa alami (Natural Language Processing/NLP). Dengan menggunakan teknik NLP, Anda dapat melakukan analisis teks lebih lanjut, seperti ekstraksi *fitur*, pemahaman bahasa, atau klasifikasi teks yang lebih canggih.
- f. Validasi dan Evaluasi yang Lebih Komprehensif: Lakukan validasi dan evaluasi yang lebih komprehensif terhadap sistem pakar yang dikembangkan. Gunakan berbagai metrik evaluasi, seperti akurasi, presisi, recall, dan F1-score, serta lakukan pengujian silang (*cross-validation*) untuk memastikan performa yang konsisten dan

generalisasi yang baik pada data yang belum pernah dilihat sebelumnya.

- g. Kolaborasi dengan Profesional Medis: Melibatkan profesional medis dalam pengembangan sistem pakar sangat penting. Dengan bekerja sama dengan dokter atau pakar medis, Anda dapat memastikan bahwa sistem pakar mencerminkan pengetahuan medis terkini dan dapat memberikan rekomendasi yang tepat dalam praktik klinis.

Monitoring dan Pemeliharaan Sistem: Setelah sistem pakar diperkenalkan, penting untuk melakukan monitoring dan pemeliharaan yang berkala. Perbarui basis pengetahuan dengan penelitian dan perkembangan terbaru dalam bidang kesehatan. Juga, perhatikan umpan balik dari pengguna dan pakar medis untuk meningkatkan sistem pakar secara keseluruhan.

Dengan mengikuti saran-saran ini, Anda dapat meningkatkan kualitas dan performa sistem pakar yang menggunakan AI (*Artificial intelligence*) (Roy Mubarak, S.T., 2020).

DAFTAR PUSTAKA

- Ahyuna, A., & Aryasa, K. 2017. Sistem Pakar Diagnosa Dan Tatalaksana Penyakit Demam Berdarah Dengue Menggunakan Jaringan Saraf Tiruan Backpropagation. *E-JURNAL JUSITI: Jurnal Sistem Informasi Dan Teknologi Informasi*, 6(1), 1–11. <http://ejurnal.dipaneegara.ac.id/index.php/jusiti/article/view/40>
- Chatrina, Siregar, N., Ruli, A, Siregar, R., & Yoga, Distra, Sudirman, M. 2020. Implementasi Metode Naive Bayes Classifier (NBC) Pada Komentar Warga Sekolah Mengenai Pelaksanaan Pembelajaran Jarak Jauh (PJJ). *Jurnal Teknologia*, 34(1), 102–110. <https://aperti.e-journal.id/teknologia/article/view/67>
- Ginting, S. L. B., & Trinanda, R. P. 2014. Penggunaan Metode Naïve Bayes Classifier Pada Aplikasi Perpustakaan. *JBP UNIKOM*, 5(1), 1–7. https://elib.unikom.ac.id/files/disk1/632/jbptunikompp-gdl-reggypasya-31586-11-unikom_r-l.pdf
- Hakam Febtadianrano Putro, Retno Tri Vlandari, W. L. Y. S. 2020. Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan. *Jurnal Tikomsin*, 8(2), 1. <https://p3m.sinus.ac.id/jurnal/index.php/TIKomSiN/article/download/500/415>
- Ilia, P. T. I. 2015. Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size. *CATENA*, 145(1), 164–179. <https://doi.org/https://doi.org/10.1016/j.catena.2016.06.004>
- Rizqiyani, V., Mulwinda, A., & Putri, R. D. M. 2017. Klasifikasi Judul Buku dengan Algoritma Naive Bayes dan Pencarian Buku pada Perpustakaan Jurusan Teknik Elektro. *Jurnal Teknik Elektro*, 9(2), 60–65. <https://journal.unnes.ac.id/nju/index.php/jte/article/view/11658/7332>

- Roy Mubarak, S.T., M. K. 2020. Implementasi Artificial Intelligence Dalam Proses Industri Manufaktur Otomotif. *Jurnal Ilmu Komputer J*, 3(10-15), 10-15. <http://jurnal-pranataindonesia.ac.id/index.php/jik/article/viewFile/32003/28>
- Wulandari, Novi Yohanes, S. A. 2018. Sistem Pakar Untuk Mendiagnosa dan Klasifikasi Anak Berkebutuhan Khusus Dengan Backward Dan Forward Chaining Pada Slb Cahaya Pertiwi di Bekasi. In A. Y. Siregar (Ed.), *Jurnal Ilmu Komputer* (Jurnal Ilm, Vol. 1, Issue 1). Jurnal Ilmu Komputer-JIK.

BAB 10

PENERAPAN ALGORITMA MARKOV DALAM SISTEM PAKAR

Oleh Reagan Surbakti Saragih

10.1 Pendahuluan

Sistem pakar telah menjadi alat yang berharga dalam berbagai bidang, termasuk di bidang kesehatan, keuangan, industri, dan lain sebagainya. Mereka membantu dalam mengambil keputusan yang kompleks dengan menganalisis data, menerapkan pengetahuan domain, dan menyediakan rekomendasi yang tepat.

Salah satu pendekatan yang dapat digunakan dalam pengembangan sistem pakar adalah Algoritma Markov. Algoritma Markov, atau Model Markov, adalah model probabilistik yang dapat memodelkan perubahan keadaan dari waktu ke waktu berdasarkan urutan peristiwa yang terjadi. Dalam konteks sistem pakar, algoritma ini dapat memberikan wawasan tentang perubahan keadaan sistem, membuat prediksi, dan membantu dalam pengambilan keputusan.

Pada buku ini, kami akan menjelajahi penerapan Algoritma Markov dalam sistem pakar. Kami akan membahas konsep dasar Algoritma Markov, termasuk Model Markov, rantai Markov, dan Model Markov Tersembunyi (*Hidden Markov Model*). Kami juga akan membahas langkah-langkah dalam pembuatan model Markov, estimasi parameter, prediksi keadaan, dan pengambilan keputusan.

Buku ini juga akan menyoroti tantangan yang mungkin dihadapi dalam penerapan algoritma Markov dalam sistem pakar, seperti keterbatasan data, kompleksitas model, dan asumsi yang

terkait. Selain itu, kita juga akan membahas peluang dan manfaat yang ditawarkan oleh algoritma Markov dalam sistem pakar, seperti kemampuan untuk memodelkan dinamika sistem dan membuat prediksi yang relevan.

Dengan pemahaman yang mendalam tentang penerapan Algoritma Markov dalam sistem pakar, pembaca akan dapat mengaplikasikan pengetahuan ini dalam berbagai konteks dan membuat sistem pakar yang lebih cerdas dan efektif. Buku ini diharapkan dapat memberikan wawasan yang berharga bagi pembaca yang tertarik dalam pengembangan sistem pakar yang menggunakan Algoritma Markov.

Selanjutnya, kita akan menjelajahi dasar-dasar Algoritma Markov dan langkah-langkah dalam penerapannya dalam sistem pakar.

10.2 Dasar-dasar Algoritma Markov

Algoritma Markov, juga dikenal sebagai Model Markov, adalah model probabilistik yang digunakan untuk memodelkan perubahan keadaan dari waktu ke waktu berdasarkan urutan peristiwa yang terjadi. Model Markov sangat berguna dalam sistem pakar karena dapat menggambarkan dependensi waktu dan memprediksi keadaan masa depan berdasarkan keadaan saat ini. Berikut adalah beberapa konsep dasar yang perlu dipahami dalam Algoritma Markov:

1. Model Markov
 - a. Model Markov adalah model matematis yang menggambarkan perubahan keadaan sistem secara probabilistik.
 - b. Model ini didasarkan pada asumsi bahwa keadaan sistem pada waktu tertentu hanya bergantung pada keadaan sebelumnya dan tidak bergantung pada keadaan sebelumnya sebelumnya (tidak ada memori).

2. Rantai Markov
 - a. Rantai Markov adalah jenis Model Markov yang paling umum.
 - b. Rantai Markov terdiri dari sejumlah keadaan diskret dan transisi probabilistik antara keadaan-keadaan tersebut.
 - c. Probabilitas transisi menunjukkan peluang perubahan dari satu keadaan ke keadaan lain.
3. Model Markov Tersembunyi (*Hidden Markov Model/HMM*)
 - a. Model Markov Tersembunyi adalah model yang digunakan ketika tidak semua keadaan dapat diamati langsung.
 - b. Model ini terdiri dari keadaan tersembunyi (yang tidak diamati) dan peristiwa yang teramati yang dipancarkan oleh keadaan tersembunyi.
 - c. Probabilitas emisi menggambarkan peluang pengamatan peristiwa tertentu berdasarkan keadaan tersembunyi.
4. *State Space*
 - a. State space adalah himpunan semua keadaan yang mungkin dalam Model Markov.
 - b. Keadaan bisa bersifat diskret atau kontinu, tergantung pada konteks masalah yang sedang dihadapi.
5. *Probabilitas Transisi*
 - a. Probabilitas transisi adalah probabilitas perubahan dari satu keadaan ke keadaan lain dalam Model Markov.
 - b. Probabilitas ini digunakan untuk menggambarkan bagaimana sistem bergerak dari satu keadaan ke keadaan lain seiring waktu.

6. Probabilitas Emisi

- a. Probabilitas emisi adalah probabilitas peristiwa yang teramati muncul berdasarkan keadaan tersembunyi dalam Model Markov Tersembunyi.
- b. Probabilitas ini menjelaskan bagaimana pengamatan peristiwa terjadi berdasarkan keadaan yang mendasarin.

Dengan pemahaman tentang dasar-dasar ini, Anda dapat memahami konsep utama dalam Algoritma Markov. Lanjutkan ke langkah-langkah berikutnya dalam penerapannya dalam sistem pakar untuk memperdalam pengetahuan Anda.

10.3 Konsep Model Markov

Model Markov adalah model probabilistik yang digunakan untuk memodelkan perubahan keadaan dari waktu ke waktu berdasarkan urutan peristiwa yang terjadi. Model ini sangat berguna dalam banyak bidang, termasuk sistem pakar, karena mampu memprediksi keadaan masa depan berdasarkan keadaan saat ini.

1. Pengenalan Model Markov dan Prinsip Dasarnya

Model Markov adalah model matematis yang digunakan untuk memodelkan perubahan keadaan dari waktu ke waktu berdasarkan urutan peristiwa yang terjadi. Model ini didasarkan pada asumsi bahwa keadaan sistem pada waktu tertentu hanya bergantung pada keadaan sebelumnya dan tidak bergantung pada keadaan sebelumnya sebelumnya (tidak ada memori). Prinsip dasar Model Markov adalah:

a. Markov Property:

Prinsip Markov menyatakan bahwa keadaan saat ini adalah cukup untuk memprediksi keadaan masa depan dan tidak memerlukan pengetahuan tentang sejarah lengkap sistem.

Dalam kata lain, masa depan hanya bergantung pada keadaan saat ini.

b. Independensi:

Model Markov mengasumsikan independensi waktu, yaitu keadaan masa depan tidak dipengaruhi oleh urutan peristiwa sebelumnya, selain keadaan saat ini. Ini dikenal sebagai sifat tanpa memori. Oleh karena itu, Model Markov sering digambarkan sebagai proses tanpa memori.

c. Transisi Probabilistik:

Model Markov menggunakan transisi probabilistik untuk menggambarkan perubahan dari satu keadaan ke keadaan lain. Probabilitas transisi menunjukkan peluang perubahan keadaan berdasarkan keadaan saat ini. Probabilitas ini dapat ditentukan berdasarkan data historis atau pengetahuan domain yang relevan.

d. Rantai Markov:

Rantai Markov adalah bentuk paling umum dari Model Markov. Ini terdiri dari sejumlah keadaan diskret dan transisi probabilistik antara keadaan-keadaan tersebut. Dalam rantai Markov, probabilitas transisi tetap konstan seiring waktu.

Contoh sederhana dari Model Markov adalah "Cuaca". Misalnya, kita ingin memprediksi cuaca besok (keadaan masa depan) berdasarkan cuaca hari ini (keadaan saat ini). Jika kita menggunakan Model Markov, kita akan mengasumsikan bahwa cuaca besok hanya bergantung pada cuaca hari ini, dan tidak dipengaruhi oleh cuaca sebelumnya. Misalnya, jika cuaca hari ini adalah cerah, maka probabilitas cuaca besok menjadi cerah atau berawan dapat ditentukan berdasarkan data historis atau pengetahuan domain.

Dalam Model Markov, penting untuk menentukan state space (himpunan keadaan yang mungkin), matriks transisi

(probabilitas perubahan keadaan), dan distribusi peluang awal (probabilitas awal keadaan). Dengan menggunakan prinsip dasar ini, Model Markov dapat diterapkan dalam berbagai konteks, termasuk dalam sistem pakar, untuk memprediksi dan mengambil keputusan berdasarkan perubahan keadaan sistem dari waktu ke waktu.

10.4 Representasi Model Markov

Representasi Model Markov dapat dilakukan dengan menggunakan beberapa elemen penting, yaitu state space, matriks transisi, dan distribusi peluang awal. Berikut adalah penjelasan lebih lanjut tentang setiap elemen tersebut:

1. State Space:

State space adalah himpunan semua keadaan yang mungkin dalam Model Markov. Ini mencakup semua keadaan yang dapat diambil oleh sistem. State space bisa bersifat diskret atau kontinu, tergantung pada jenis masalah yang sedang dihadapi. Misalnya, jika kita ingin memodelkan cuaca sebagai Model Markov, state space dapat terdiri dari keadaan-keadaan seperti "cerah", "berawan", "hujan", dan sebagainya.

2. Matriks Transisi:

Matriks transisi adalah representasi matematis dari probabilitas transisi antara keadaan-keadaan dalam Model Markov. Matriks ini berukuran $N \times N$, di mana N adalah jumlah keadaan dalam state space. Setiap elemen matriks menunjukkan probabilitas perubahan dari satu keadaan ke keadaan lain. Misalnya, jika kita memiliki state space dengan tiga keadaan (A, B, C), maka matriks transisi akan memiliki bentuk berikut:

$$T = \begin{bmatrix} P(A \rightarrow A) & P(A \rightarrow B) & P(A \rightarrow C) \\ P(B \rightarrow A) & P(B \rightarrow B) & P(B \rightarrow C) \\ P(C \rightarrow A) & P(C \rightarrow B) & P(C \rightarrow C) \end{bmatrix}$$

Di sini, setiap elemen $T[i][j]$ menunjukkan probabilitas perubahan dari keadaan i ke keadaan j .

3. Distribusi Peluang Awal:

Distribusi peluang awal adalah distribusi probabilitas yang menunjukkan probabilitas awal atau peluang setiap keadaan dalam Model Markov pada waktu awal. Ini adalah vektor baris dengan ukuran N , di mana N adalah jumlah keadaan dalam state space. Setiap elemen vektor menunjukkan probabilitas awal setiap keadaan. Misalnya, jika kita memiliki state space dengan tiga keadaan (A, B, C), maka distribusi peluang awal dapat dinyatakan sebagai vektor berikut:

$$P = [P(A), P(B), P(C)]$$

Di sini, $P[i]$ adalah probabilitas awal untuk keadaan i .

Dengan menggabungkan state space, matriks transisi, dan distribusi peluang awal, Model Markov dapat direpresentasikan secara matematis. Representasi ini memungkinkan kita untuk melakukan analisis, perhitungan, dan prediksi berdasarkan probabilitas perubahan keadaan dari waktu ke waktu.

Pada implementasi Model Markov dalam sistem pakar, perhatikan bahwa matriks transisi dan distribusi peluang awal harus diisi dengan probabilitas yang relevan berdasarkan data historis atau pengetahuan domain yang tersedia. Dengan memperbarui matriks transisi dan distribusi peluang awal sesuai dengan perubahan dalam sistem, Model Markov dapat memberikan prediksi yang lebih akurat dan membantu dalam pengambilan keputusan.

Estimasi parameter pada Model Markov melibatkan penentuan nilai probabilitas transisi dan distribusi peluang awal berdasarkan data yang tersedia. Terdapat beberapa metode yang umum digunakan untuk melakukan estimasi parameter ini, yaitu:

1. Metode *Maximum Likelihood Estimation* (MLE)

Metode MLE digunakan untuk memperkirakan parameter Model Markov berdasarkan data observasi yang ada. Pendekatan ini mencari nilai probabilitas transisi dan distribusi peluang awal yang memberikan kemungkinan tertinggi untuk urutan peristiwa yang diamati. Langkah-langkah umum dalam metode MLE adalah sebagai berikut:

- a. Menghitung frekuensi observasi dari urutan peristiwa yang diamati.
- b. Menghitung probabilitas relatif untuk setiap transisi dan keadaan awal berdasarkan frekuensi observasi.
- c. Normalisasi probabilitas relatif menjadi probabilitas sesuai dengan aturan bahwa total probabilitas dalam satu keadaan harus sama dengan 1.

Metode MLE dapat digunakan untuk memperkirakan parameter Model Markov dengan memaksimalkan likelihood fungsi yang sesuai dengan data observasi.

2. *Expectation-Maximization* (EM)

Metode ini sering digunakan dalam kasus Model Markov Tersembunyi (*Hidden Markov Model*). EM adalah algoritma iteratif yang menggabungkan langkah ekspektasi (*expectation*) dan langkah pemaksimalan (*maximization*). Pada langkah ekspektasi, diperkirakan probabilitas tersembunyi (seperti status keadaan tersembunyi dalam Model Markov Tersembunyi), sedangkan pada langkah pemaksimalan, dilakukan pembaruan parameter untuk memaksimalkan likelihood function.

3. Metode Bayesian

Metode Bayesian menggunakan pendekatan probabilitas untuk melakukan estimasi parameter. Dalam metode ini, diadopsi pemodelan probabilistik yang melibatkan pengetahuan awal tentang parameter (prior) dan data observasi. Estimasi parameter diperoleh dengan

menggabungkan prior dengan likelihood function untuk mendapatkan posterior distribution. Metode Bayesian sering digunakan ketika terdapat sedikit data observasi, dan pendekatan ini memungkinkan penggunaan pengetahuan awal yang relevan.

4. Metode Bootstrap:

Metode Bootstrap adalah teknik yang digunakan untuk melakukan estimasi parameter dengan menghasilkan banyak sampel acak dari data observasi yang ada. Dalam konteks Algoritma Markov, metode ini dapat digunakan untuk memperkirakan parameter dengan menghasilkan banyak urutan peristiwa berdasarkan data observasi. Estimasi parameter kemudian diperoleh dengan menganalisis distribusi hasil *bootstrap*.

10.5 Pengambilan Keputusan

Pengambilan keputusan berdasarkan probabilitas keadaan yang dihasilkan oleh model Markov melibatkan analisis probabilitas untuk menentukan tindakan yang optimal. Setelah model Markov menghasilkan prediksi tentang keadaan yang mungkin terjadi, probabilitas keadaan tersebut dapat digunakan sebagai dasar untuk pengambilan keputusan. Berikut adalah beberapa pendekatan umum untuk pengambilan keputusan dalam konteks model Markov:

1. Pemilihan Keadaan Terbaik (*Best State Selection*):

Metode ini melibatkan pemilihan keadaan dengan probabilitas tertinggi sebagai keadaan yang paling mungkin terjadi. Misalnya, jika model Markov menghasilkan probabilitas tertinggi untuk keadaan "A" dibandingkan dengan keadaan lainnya, maka keputusan yang diambil adalah berdasarkan asumsi bahwa keadaan "A" adalah yang paling mungkin terjadi.

2. Tindakan Terbaik (*Best Action Selection*):

Jika model Markov juga menyediakan informasi tentang tindakan yang dapat diambil dalam setiap keadaan, pengambilan keputusan dapat didasarkan pada tindakan dengan nilai probabilitas terbaik. Ini melibatkan memilih tindakan yang memiliki probabilitas keberhasilan tertinggi atau nilai utilitas terbesar berdasarkan model Markov. Misalnya, jika model Markov memberikan probabilitas tertinggi untuk tindakan "X" dalam keadaan "A", maka tindakan "X" dipilih sebagai tindakan yang optimal.

3. Pengambilan Keputusan Stokastik:

Pendekatan ini melibatkan pengambilan keputusan berdasarkan distribusi probabilitas dari beberapa keadaan atau tindakan yang mungkin. Misalnya, jika terdapat beberapa keadaan dengan probabilitas yang signifikan, maka keputusan dapat diambil secara acak berdasarkan probabilitas masing-masing keadaan. Pendekatan ini sering digunakan ketika keputusan harus diambil dalam situasi yang tidak pasti atau ketika variasi hasil diinginkan.

Pada akhirnya, metode pengambilan keputusan yang digunakan akan bergantung pada konteks aplikasi dan preferensi pengguna. Selain itu, pertimbangan lain seperti risiko, tujuan, dan konsekuensi dari setiap tindakan juga dapat mempengaruhi proses pengambilan keputusan berdasarkan probabilitas keadaan yang dihasilkan oleh model Markov.

10.6 Evaluasi dan Pemeliharaan

Setelah mengimplementasikan Algoritma Markov dalam sistem pakar, penting untuk melakukan evaluasi dan pemeliharaan terhadap model Markov yang digunakan. Evaluasi dan pemeliharaan ini bertujuan untuk memastikan

bahwa model Markov tetap akurat, efektif, dan menghasilkan prediksi yang dapat diandalkan. Berikut ini adalah beberapa aspek yang perlu diperhatikan dalam evaluasi dan pemeliharaan model Markov:

1. Evaluasi Performa:

Untuk mengukur performa model Markov, perlu digunakan metrik evaluasi yang relevan. Metrik ini dapat mencakup akurasi prediksi, presisi, *recall*, *F1-score*, atau metrik lainnya yang sesuai dengan tujuan sistem pakar yang dikembangkan. Dengan melakukan evaluasi performa secara berkala, Anda dapat mengetahui sejauh mana model Markov mampu memprediksi keadaan dengan benar dan mengidentifikasi area yang perlu diperbaiki.

2. Perbarui Data

Model Markov membutuhkan data yang relevan dan mutakhir untuk memberikan prediksi yang akurat. Oleh karena itu, penting untuk memperbarui data yang digunakan dalam model secara berkala. Jika data yang digunakan sudah tidak mewakili kondisi atau perubahan yang terjadi dalam sistem, kemungkinan prediksi model Markov menjadi tidak akurat. Pembaruan data dapat melibatkan pengumpulan data baru, penghapusan data yang tidak relevan, atau penggunaan teknik pembaruan data yang sesuai dengan kebutuhan.

3. Optimalisasi Performa

Pemeliharaan model Markov juga melibatkan optimalisasi performa. Anda dapat melakukan analisis terhadap waktu komputasi yang dibutuhkan oleh model Markov, kebutuhan memori, atau penggunaan sumber daya lainnya. Jika model terlalu lambat atau memerlukan sumber daya yang besar, mungkin perlu dilakukan langkah-langkah untuk meningkatkan efisiensi, seperti memperbaiki algoritma

yang digunakan, melakukan kompresi data, atau menggunakan teknik lain yang relevan.

4. Revalidasi Model

Model Markov juga perlu diuji ulang secara berkala untuk memastikan keandalan dan konsistensinya. Revalidasi model dapat melibatkan pengujian ulang dengan data yang independen atau menggunakan teknik *cross-validation* untuk memastikan bahwa model tetap bekerja dengan baik pada data yang tidak digunakan dalam pelatihan.

5. Perbaikan dan Peningkatan

Selama evaluasi dan pemeliharaan, mungkin ditemukan kelemahan atau area yang perlu ditingkatkan dalam model Markov. Berdasarkan hasil evaluasi, Anda dapat melakukan perbaikan atau penyesuaian pada model, seperti memperbaiki estimasi parameter, menambahkan atau mengubah fitur yang digunakan, atau mengubah struktur model jika diperlukan.

10.7 Metrik Evaluasi performa model Markov dalam sistem pakar

Dalam evaluasi performa model Markov dalam sistem pakar, terdapat beberapa metrik yang dapat digunakan untuk mengukur kualitas prediksi dan efektivitas model. Beberapa metrik evaluasi yang umum digunakan adalah sebagai berikut:

1. Akurasi (*Accuracy*)

Akurasi adalah metrik yang mengukur sejauh mana model Markov mampu memprediksi keadaan dengan benar. Metrik ini menghitung persentase prediksi yang sesuai dengan keadaan sebenarnya. Akurasi diperoleh dengan membagi jumlah prediksi benar dengan total jumlah prediksi.

2. Presisi (*Precision*)

Presisi mengukur seberapa baik model Markov dalam mengidentifikasi keadaan tertentu secara tepat. Presisi dihitung

dengan membagi jumlah prediksi benar untuk suatu keadaan dengan total prediksi yang dilakukan untuk keadaan tersebut. Presisi memberikan informasi tentang seberapa andal model dalam mengklasifikasikan keadaan tertentu.

3. *Recall*

Recall (juga dikenal sebagai sensitivitas) mengukur sejauh mana model Markov dapat mendeteksi keadaan yang sebenarnya. *Recall* dihitung dengan membagi jumlah prediksi benar untuk suatu keadaan dengan total jumlah keadaan sebenarnya yang seharusnya terdeteksi. *Recall* memberikan informasi tentang seberapa baik model dalam mengenali semua keadaan yang relevan.

4. *F1-Score*

F1-score adalah *harmonic mean* antara presisi dan *recall*. Metrik ini memberikan gambaran keseluruhan tentang keseimbangan antara presisi dan *recall*. F1-score dapat digunakan sebagai metrik evaluasi yang baik ketika jumlah keadaan yang tidak seimbang (*imbalance*) dalam data.

5. *Log-Likelihood*

Log-likelihood digunakan terutama dalam Model Markov Tersembunyi (*Hidden Markov Model*). Metrik ini mengukur seberapa baik model Markov mampu menjelaskan data observasi. Semakin tinggi nilai *log-likelihood*, semakin baik model cocok dengan data.

6. *Confusion Matrix*

Confusion matrix adalah tabel yang memperlihatkan jumlah prediksi “benar” dan “salah” untuk setiap keadaan yang diprediksi. Dari confusion matrix, dapat dihitung berbagai metrik evaluasi seperti akurasi, presisi, *recall*, dan lainnya.

Pilihan metrik evaluasi yang digunakan tergantung pada jenis aplikasi dan kebutuhan sistem pakar yang dikembangkan.

Penting untuk memilih metrik evaluasi yang sesuai dengan konteks dan tujuan penggunaan model Markov dalam sistem pakar.

10.8 Pemeliharaan Model Markov

Dalam penerapan algoritma Markov dalam sistem pakar, pemeliharaan model Markov adalah langkah penting untuk memastikan kinerja yang optimal dan keakuratan prediksi. Dua aspek utama dalam pemeliharaan model Markov adalah pembaruan data dan optimalisasi performa.

Pembaruan data melibatkan memastikan bahwa data yang digunakan dalam model Markov tetap relevan dan mutakhir. Dalam beberapa kasus, perubahan dalam sistem atau lingkungan dapat mempengaruhi distribusi keadaan atau peristiwa, sehingga data perlu diperbarui secara teratur. Hal ini dapat melibatkan pengumpulan data baru, penghapusan data yang tidak relevan, atau teknik pembaruan data lainnya yang sesuai dengan kebutuhan. Dengan memperbarui data, model Markov dapat tetap mencerminkan situasi yang terbaru dan menghasilkan prediksi yang lebih akurat.

Selain itu, optimalisasi performa juga penting dalam pemeliharaan model Markov. Ini melibatkan analisis terhadap kinerja model, seperti waktu komputasi yang dibutuhkan, penggunaan memori, atau penggunaan sumber daya lainnya. Jika model Markov terlalu lambat atau memerlukan sumber daya yang besar, langkah-langkah optimalisasi dapat diambil, seperti memperbaiki algoritma yang digunakan, melakukan kompresi data, atau menerapkan teknik lain yang relevan. Optimalisasi performa membantu memastikan bahwa model Markov dapat berjalan dengan efisien dan memberikan prediksi dalam waktu yang sesuai.

Dengan memperbarui data secara teratur dan melakukan optimalisasi performa, pemeliharaan model Markov dapat menjaga kualitas dan konsistensi prediksi. Ini memungkinkan sistem pakar

untuk memberikan rekomendasi yang lebih akurat dan berguna kepada pengguna. Penting untuk diingat bahwa pemeliharaan model Markov adalah proses berkelanjutan, karena perubahan dalam sistem dan lingkungan dapat terus terjadi. Dengan pemeliharaan yang tepat, model Markov dapat tetap relevan dan efektif dalam mendukung pengambilan keputusan dalam sistem pakar.

DAFTAR PUSTAKA

- Russell, S. J., & Norvig, P. 2016. Artificial Intelligence: A Modern Approach. Pearson.
Buku ini menyajikan pemahaman yang komprehensif tentang kecerdasan buatan dan mencakup topik algoritma Markov serta aplikasinya dalam sistem pakar.
- Rabiner, L. R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 77(2), 257-286.
Makalah ini memberikan tutorial rinci tentang Model Markov Tersembunyi (Hidden Markov Model) dan aplikasinya dalam pengenalan ucapan. Konsep dan metode dalam makalah ini dapat diterapkan dalam sistem pakar.
- Barber, D. 2012. Bayesian Reasoning and Machine Learning. Cambridge University Press.
Buku ini menggabungkan konsep algoritma Markov dan pendekatan bayesian dalam pembelajaran mesin. Berisi penjelasan teoritis dan studi kasus yang relevan dengan penerapan algoritma Markov dalam sistem pakar.
- Murphy, K. P. 2012. Machine Learning: A Probabilistic Perspective. MIT Press.
Buku ini memberikan pemahaman yang mendalam tentang pendekatan probabilistik dalam pembelajaran mesin. Termasuk di dalamnya adalah pembahasan mengenai algoritma Markov dan aplikasinya dalam berbagai bidang, termasuk sistem pakar.
- Koller, D., & Friedman, N. 2009. Probabilistic Graphical Models: Principles and Techniques. MIT Press.
Buku ini menyajikan prinsip-prinsip dan teknik-teknik yang terkait dengan model grafis probabilistik. Model Markov adalah salah satu topik utama dalam buku ini dan dapat diterapkan dalam sistem pakar.

Sutton, R. S., & Barto, A. G. 2018. Reinforcement Learning: An Introduction. MIT Press.

BAB 11

PENERAPAN ALGORITMA CERTAINTY FACTOR DALAM SISTEM PAKAR

Oleh Herlina Latipa Sari

11.1 Ketidakpastian (*Uncertainty*)

Ketidakpastian dapat diartikan sebagai kurangnya informasi untuk mengambil keputusan, hal ini menjadi masalah tersendiri karena dengan ketidakpastian keputusan yang diambil menjadi tidak tepat. Ketidakpastian juga hadir di dalam sistem pakar, sumber dari ketidakpastian berasal dari validitas kaidah berbasis pengetahuan dan validitas yang berasal dari respon pengguna sistem pakar terhadap query yang diminta oleh sistem pakar. Ketidakpastian yang terjadi pada satu kaidah disebabkan oleh 3 hal yaitu aturan tunggal, ketidaksesuaian antar kaidah dan resolusi konflik (Hartati dan Iswanti, 2008)

Uncertainty atau ketidakpastian dalam *Artificial Intelligence* (AI) disajikan dalam tiga langkah yaitu:

1. Seorang pakar menyediakan pengetahuan tidak pasti (*inexact*), yang berupa term atau aturan dengan nilai berupa kemungkinan. Bisa berupa numeric (contoh nilai probabilitas), grafik, atau simbolik.
2. Pengetahuan yang tidak pasti (*inexact*) dari himpunan dasar suatu kejadian dapat secara langsung digunakan untuk menarik inferensi (kesimpulan) dalam kasus yang sederhana.

Dalam langkah ketidakpastian pada banyak kasus, kejadian-kejadian yang bervariasi saling berhubungan. Sehingga perlu untuk mengkombinasikan informasi-informasi yang tersedia dalam representasi ketidakpastian dari himpunan dasar kejadian ke dalam *goal value* (nilai tujuan) system. Beberapa metode dapat digunakan secara terintegrasi. Metode utamanya adalah Probabilitas Bayesian, teori tentang kejadian, factor kepastian, dan himpunan fuzzy (kabur).

Terdapat 4 Sumber ketidakpastian adalah sebagai berikut :

1. Data
Kehilangan data, data tidak dapat diandalkan, data yang mendua, penyajian data tidak tepat, data tidak konsisten, data subjektif, data diperoleh dari kelalaian.
2. Pengetahuan pakar
 - a. Ketidakkonsistenan antara pakar yang berbeda
 - b. Kemasuk-akalan (best guess dari pakar)
 - c. Kualitas : pemahaman yang dalam pada pengetahuan causal (sebab akibat) dan kualitas secara statistik (pengamatan)
 - d. Cakupan (hanya domain)
3. Representasi pengetahuan
Representasi pengetahuan adalah proses dimana sebuah objek ditangkap oleh indera seseorang, lalu masuk ke akal untuk diproses yang hasilnya adalah sebuah konsep/ide yang dengan bahasa akan disampaikan/diungkapkan kembali. Dan setiap orang beda-beda dalam menangkap semua objek menjadi ide sehingga menjadi tidak pasti.
 - a. Keterbatasan model pada system riil
 - b. Keterbatasan pengungkapan dari mekanisme representasi.
4. Proses inferensi
Inferensi adalah tindakan atau proses yang berasal kesimpulan

- a. Deduktif-hasil yang diperoleh secara formal benar, tetapi salah pada system riil
- b. Induktif-konklusi baru tidak ditemukan dengan baik metode penalaran tidak valid.

11.2 Algoritma Certainty Factor (CF)

Faktor kepastian (*Certainty Factor*) menyatakan dalam sebuah kejadian (fakta atau hipotesis) berdasarkan bukti atau penilaian pakar (Kusumadewi, 2003). Metode Certainty Factor digunakan ketika menghadapi suatu masalah yang jawabannya tidak pasti. Ketidakpastian ini bisa merupakan probabilitas, Algoritma ini diperkenalkan oleh Shortlife Buchanan pada tahun 1970-an.

Salah satu contoh aplikasi sistem pakar yang menggunakan metode *Certainty Factor (CF)* untuk menangani ketidakpastian adalah MYCIN, sistem pakar untuk mendiagnosa infeksi bakteri pada darah. *Certainty Factor* didefinisikan sebagai berikut : (Giarattano dan Riley, 2005).

$$CF(H,E) = MB(H,E) - MD(H,E)$$

Dimana :

$CF(H,E)$: *Certainty factor* hipotesis H yang dipengaruhi oleh *evidence* (gejala) E

$MB(H,E)$: Ukuran kepercayaan (*measure of increased belief*) terhadap hipotesis H yang dipengaruhi oleh *evidence* E

$MD(H,E)$: Ukuran ketidakpercayaan (*measure of increased disbelief*) terhadap hipotesis H yang dipengaruhi oleh *evidence* E.

Algoritma *Certainty Factor (CF)* mirip dengan *fuzzy logic*, karena ketidakpastian direpresentasikan dengan derajat kepercayaan sedangkan perbedaannya adalah pada *fuzzy logic* saat perhitungan untuk rule yang premisnya lebih dari satu , fuzzy

logic tidak memiliki nilai keyakinan untuk rule tersebut sehingga perhitungannya hanya melihat nilai terkecil untuk operator AND atau nilai terbesar untuk operator OR dari setiap premis yang pada rule tersebut berbeda dengan *certainty factor* yaitu setiap rule memiliki nilai keyakinan sendiri tidak hanya premis-premisnya saja yang memiliki nilai keyakinan (Harmon dkk, 1985).

Sistem pakar perlu dibekali dengan konsep faktor kepastian agar dapat menangani masalah yang berhubungan dengan ketidakpastian. Biasanya ketidakpastian ini disebabkan karena informasi atau data yang diperoleh tidak lengkap, tidak dapat dipercaya sepenuhnya, berasal dari berbagai sumber dan saling bertolak belakang, bersifat tipikal atau mirip dan Bahasa penyajiannya kurang tepat. Seorang pakar misalnya dokter, sering kali menganalisis informasi yang ada dengan ungkapan seperti “mungkin”, “kemungkinan besar”, “hampir pasti”. Untuk mengakomodasi hal ini kita menggunakan *Certainty Factor* (CF) guna menggambarkan tingkat keyakinan pakar terhadap masalah yang sedang dihadapi (Sutojo, Mulyanto dan Suhartono, 2010).

Nilai *Certainty Factor* ada 2 yaitu :

1. Nilai *Certainty Factor* kaidah yang nilainya melekat pada suatu kaidah/rule tertentu dan besarnya nilai diberikan oleh pakar.
2. Nilai *Certainty Factor* yang diberikan oleh pengguna untuk mewakili derajat kepastian / keyakinan atas premis (misalnya gejala, kondisi, ciri) yang dialami pengguna.

Pada implementasi sistem pakar akan terlihat bahwa nilai CF yang diberikan oleh pengguna ada pada sesi konsultasi, sehingga sistem pakar selain meminta jawaban pengguna atas pertanyaan yang diajukan juga meminta masukan nilai *certainty factor* atas jawaban tersebut ; sedangkan nilai CF kaidah dapat pada suatu *rule* / kaidah yang biasanya diberikan oleh pakar berdasarkan pengalamannya; sehingga sangat bersifat subyektif.

Pemberian nilai *certainty factor* kaidah bisa sangat bervariasi dalam arti nilai yang diberikan oleh pakar bisa berbeda-beda untuk suatu kaidah yang sama. Kelebihan Algoritma *Certainty Factor* adalah :

1. Metode ini cocok dipakai dalam sistem pakar yang mengandung ketidakpastian.
2. Dalam sekali proses perhitungan dapat mengolah 2 data saja sehingga keakuratan data dapat terjaga.

Kekurangan Algoritma *Certainty Factor* adalah :

1. Pemodelan ketidakpastian yang menggunakan perhitungan algoritma *Certainty Factor* biasanya masih diperdebatkan.
2. Untuk data lebih dari 2 buah, harus dilakukan beberapa kali pengolahan data.

11.3 Penerapan Algoritma Certainty Factor (CF) dalam Sistem Pakar

Dalam Penerapan Algoritma *Certainty Factor* (CF) dalam Sistem Pakar ini pada studi kasus identifikasi penyakit Skizofrenia dalam mendiagnosa gangguan jiwa dengan metode *Certainty Factor* di Rumah Sakit Jiwa Ketergantungan Obat Soeprapto (RSJKO) Daerah Bengkulu, berdasarkan dengan observasi pengumpulan data yang menyebutkan bahwa penyakit Skizofrenia merupakan peringkat pertama dari 10 besar penyakit rawat jalan di rumah sakit tersebut.

Penerapan Algoritma *Certainty Factor* (CF) atau faktor kepastian yang diharapkan dapat memberikan jawaban kepada pengguna terhadap sesuatu yang tidak pasti. Tingkat keyakinan *Certainty Factor* (CF) menggunakan nilai untuk mengasumsikan derajat keyakinan seorang pakar terhadap suatu data dan

Tabel 11.1. Tingkat Keyakinan Certainty Factor (CF)

<i>Uncertain Term</i>	<i>Certainty Factor</i>
<i>Definitely Not</i> (Pasti Tidak)	-1.0
<i>Almost Certainly Not</i> (Hampir Pasti Tidak)	-0.8
<i>Probably Not</i> (Kemungkinan Besar Tidak)	-0.6
<i>Maybe Not</i> (Mungkin Tidak)	-0.4
<i>Unknow</i> (Tidak Tahu)	-0.2 to 0.2
<i>Maybe</i> (Mungkin)	0.4
<i>Probably</i> (Kemungkinan Besar)	0.6
<i>Almost Certainly</i> (Hampir Pasti)	0.8
<i>Definitely</i> (Pasti)	1.0

Tabel 11.2. Nilai CF gejala terhadap Penyakit

No	Nama Penyakit	Gejala	Nilai CF
1	Skizofrenia Paramoid	Mendengar suara-suara gaib yang menyuruh untuk melakukan sesuatu	1.0
		Mencium berbauan yang sebenarnya tidak ada	0.6
		Melihat sesuatu yang sebenarnya tidak ada	0.6
		Merasa ada orang yang ingin membunuhnya	1.0
		Mempunyai keyakinan bahwa dirinya memiliki kekuatan	0.6
		Menarik diri dari pergaulan luar atau masyarakat	0.2
2	Skizofrenia Terdisorganisasi	Berperilaku seperti anak kecil	0.7
		Bicara tidak teratur	1.0
		Berfikir tidak logis	0.8
		Ekspresi wajah kosong	0.8

No	Nama Penyakit	Gejala	Nilai CF
		Bergerak dan berperilaku tanpa arah	0.5
		Menarik diri dari pergaulan luar atau masyarakat	0.4
		Merasa canggung untuk berinteraksi dengan lingkungan sekitar	0.2
3	Skizofrenia Katatonik	Perangai yang tidak biasa	0.4
		Merasa canggung untuk berinteraksi dengan lingkungan sekitar	0.2
		Melakukan gerakan yang berulang-ulang	1.0
		Mengulangi kata atau kalimat berulang-ulang	1.0
		Menarik diri dari pergaulan luar atau masyarakat	0.4
4	Skizofrenia tidak tergolongkan	Mendengar suara-suara gaib yang menyuruh untuk melakukan sesuatu	0.6
		Melihat sesuatu yang sebenarnya tidak ada	0.5
		Merasa sedang dimata-matai	0.8
		Menarik diri dari pergaulan luar atau masyarakat	0.4
		Berperilaku gaduh dan gelisah	1.0
5	Skizofrenia Residul	Ketidak mampuan untuk mengalami kesenangan	0.5
		Tidak ada kesenangan untuk membentuk hubungan atau	0.8

No	Nama Penyakit	Gejala	Nilai CF
		berinteraksi	
		Ketidak mampuan melakukan aktivitas rutin	0.4
		Tidak ada ekspresi emosi yang dapat dilihat	1.0

Ada beberapa aturan untuk perhitungan *certainty factor* (CF), tetapi aturan yang digunakan pada sistem adalah seperti dibawah ini :

Rule kombinasi

IF evidence 1 (E_1) $\rightarrow$ (E_1) THEN Hipotesis (H)

IF evidence 2 (E_2) $\rightarrow$ (E_2) THEN Hipotesis (H)

$CF(CF_1, CF_2) = CF_1 + CF_2 * (1 - CF_1)$ Nilai CF_1 dan $CF_2 > 0$

$CF(CF_1, CF_2) = CF_1 + CF_2 * (1 + CF_1)$ Nilai CF_1 dan $CF_2 < 0$

$CF(CF_1, CF_2) = (CF_1 + CF_2) / (1 - \min\{|CF_1|, |CF_2|\})$ Nilai CF_1 dan CF_2 berlawanan tanda

Kasus dalam penelitian ini pada saat konsultasi, keluarga pasien atau pasien memilih gejala :

1. Mendengar suara-suara gaib yang menyuruh untuk melakukan sesuatu
2. Mencium berbau yang sebenarnya tidak ada
3. Melihat sesuatu yang sebenarnya tidak ada
4. Merasa ada orang yang ingin membunuhnya
5. Menarik diri dari pergaulan luar atau masyarakat.

Dengan masing-masing nilai CF pengguna terhadap gejala yang dipilih keluarga pasien atau pasien pada saat konsultasi yaitu:

1. Mungkin dengan nilai CF yang telah ditentukan yaitu 0.4
2. Tidak tahu dengan nilai CF yang telah ditentukan yaitu 0.2

3. Hampir pasti dengan nilai CF yang telah ditentukan yaitu 0.8
4. Mungkin dengan nilai CF yang telah ditentukan yaitu 0.4
5. Mungkin dengan nilai CF yang telah ditentukan yaitu 0.4

Berdasarkan gejala dan nilai CF pengguna yang telah dipilih oleh pengguna maka dapat dihitung nilai CF masing-masing penyakit dengan melihat tabel 13.2 nilai CF gejala terhadap penyakit untuk menentukan penyakit apa saja yang mempunyai gejala-gejala yang telah dipilih oleh keluarga pasien ataupun pasien.

1. Gejala yang terpilih untuk *skizofronia* Paranoid yaitu :
 - a. Mendengar suara-suara gaib yang menyuruh untuk melakukan sesuatu dengan nilai CF pakar 1.0
 - b. Mencium bebauan yang sebenarnya tidak ada dengan nilai CF pakar 0.6
 - c. Melihat sesuatu yang sebenarnya tidak ada dengan nilai CF pakar 0.6
 - d. Merasa ada orang yang ingin membunuhnya dengan nilai CF pakar 1.0
 - e. Menarik diri dari pergaulan luar atau masyarakat dengan nilai CF pakar 0.2

Jadi dapat di hitung nilai CF untuk penyakit *Skizofrenia* paranoid sebagai berikut :

$$CF_{\text{gejala}} = CF [\text{pengguna}] * CF [\text{pakar}]$$

$$CF_{\text{gejala1}} = 0.4 * 1.0 = 0.4$$

$$CF_{\text{gejala2}} = 0.2 * 0.6 = 0.12$$

$$CF_{\text{gejala3}} = 0.8 * 0.6 = 0.48$$

$$CF_{\text{gejala4}} = 0.4 * 1.0 = 0.4$$

$$CF_{\text{gejala5}} = 0.4 * 0.2 = 0.08$$

Setelah didapat nilai masing-masing CF_{gejala} , maka hitung nilai CF_{combine}

$$\begin{aligned}
CF_{\text{combine}} &= CF_{\text{old}} + CF_{\text{gejala}} * (1 - CF_{\text{old}}) \\
CF_{\text{combine}} (CF_{\text{gejala1}}, CF_{\text{gejala2}}) &= CF_{\text{gejala1}} + CF_{\text{gejala2}} * (1 - CF_{\text{gejala1}}) \\
&= 0.4 + 0.12 * (1 - 0.4) \\
&= 0.472_{\text{old1}} \\
CF_{\text{combine}} (CF_{\text{old1}}, CF_{\text{gejala3}}) &= CF_{\text{old1}} + CF_{\text{gejala3}} * (1 - CF_{\text{old1}}) \\
&= 0.472 + 0.48 * (1 - 0.472) \\
&= 0.72544_{\text{old2}} \\
CF_{\text{combine}} (CF_{\text{old2}}, CF_{\text{gejala4}}) &= CF_{\text{old2}} + CF_{\text{gejala4}} * (1 - CF_{\text{old2}}) \\
&= 0.725 + 0.4 * (1 - 0.725) \\
&= 0.835264_{\text{old3}} \\
CF_{\text{combine}} (CF_{\text{old3}}, CF_{\text{gejala5}}) &= CF_{\text{old3}} + CF_{\text{gejala5}} * (1 - CF_{\text{old3}}) \\
&= 0.835 + 0.4 * (1 - 0.835) \\
&= 0.8484_{\text{old4}}
\end{aligned}$$

$CF_{\text{persentase}} = CF_{\text{combine}} * 100\% = 0.8484 * 100\% = 84.84\%$
 Jadi persentase gejala yang dipilih untuk penyakit *Skizofrenia* Paranoid adalah 84.84% yang berarti hampir pasti.

2. *Skizofrenia* Terdisorganisasi hanya terpilih satu gejala yaitu : menarik diri dari pergaulan luar atau masyarakat dengan nilai CF pakar 0.4.

$$\begin{aligned}
CF_{\text{gejala}} &= CF [\text{pengguna}] * CF [\text{pakar}] \\
&= 0.2 * 0.4 = 0.08
\end{aligned}$$

$$\begin{aligned}
\text{Maka } CF_{\text{combine}} &= 0 + 0.08 * (1 - 0.08) \\
&= 0.16_{\text{old1}}
\end{aligned}$$

$CF_{\text{persentase}} = CF_{\text{combine, old1}} * 100\% = 0.16 * 100\% = 16\%$
 Jadi persentase gejala yang dipilih untuk penyakit *Skizofrenia* terdisorganisasi adalah 16% yang berarti tidak yakin teken *Skizofrenia* terdisorganisasi.

3. *Skizefronia* Katatonik hanya terpilih satu gejala yaitu menarik diri dari pergaulan luar atau masyarakat dengan nilai CF pakar 0.4 yang berarti persentase gejala yang dipilih untuk penyakit *Skizofrenia* Katatonik sama dengan penyakit *Skizofrenia* Terdisorganisasi yaitu 16% yang berarti tidak yakin.
4. Gejala yang terpilih untuk *Skizofrenia* tidak tergolongkan, yaitu :
 - a. Mendengar suara-suara gaib yang menyuruh untuk melakukan sesuatu dengan nilai CF pakar 0.6
 - b. Melihat sesuatu yang sebenarnya tidak ada dengan nilai CF pakar 0.5
 - c. Menarik diri dari pergaulan luar atau masyarakat dengan nilai CF pakar 0.4

Jadi dapat dihitung nilai CF untuk penyakit *Skizofrenia* tidak tergolongkan sebagai berikut :

$$CF_{\text{gejala}} = CF [\text{pengguna}] * CF [\text{pakar}]$$

$$CF_{\text{gejala1}} = 0.4 * 0.6 = 0.24$$

$$CF_{\text{gejala2}} = 0.8 * 0.5 = 0.4$$

$$CF_{\text{gejala3}} = 0.4 * 0.4 = 0.16$$

Setelah didapat nilai masing-masing CF_{gejala} , maka hitung nilai CF_{combine}

$$CF_{\text{combine}} = CF_{\text{old}} + CF_{\text{gejala}} * (1 - CF_{\text{old}})$$

$$CF_{\text{combine}} (CF_{\text{gejala1}}, CF_{\text{gejala2}}) = CF_{\text{gejala1}} + CF_{\text{gejala2}} * (1 - CF_{\text{gejala1}})$$

$$= 0.24 + 0.4 * (1 - 0.24)$$

$$= 0.544_{\text{old1}}$$

$$CF_{\text{combine}} (CF_{\text{old1}}, CF_{\text{gejala3}}) = CF_{\text{old1}} + CF_{\text{gejala3}} * (1 - CF_{\text{old1}})$$

$$= 0.544 + 0.16 * (1 - 0.544)$$

$$= 0.61696_{\text{old2}} \text{ dibulatkan } 0.6170$$

$$CF_{\text{persentase}} = CF_{\text{combine}} * 100\% = 0.6170 * 100\% = 61.70 \%$$

Jadi persentase gejala yang dipilih untuk penyakit *Skizofrenia* tidak tergolongkan adalah 61.70% yang berarti kemungkinan besar.

Dari perhitungan masing-masing gejala dari tiap penyakit, maka diperoleh jumlah persentasenya yaitu :

1. *Skizofrenia* Paranoid 84.84%
2. *Skizofrenia* Terdisorganisasi 16%
3. *Skizofrenia* Katatonik 16%
4. *Skizofrenia* Tidak Tergolongkan 61.70%

Dengan cara yang sama, maka semua penyakit dihitung nilai CF nya berdasarkan gejala yang dipilih dan nilai CF pengguna yang dipilih. Perhitungan dari nilai CF untuk semua penyakit diperoleh penyakit *Skizofrenia* Paranoid yang mempunyai nilai CF terbesar yaitu 0.8484 yang dipresentasikan menjadi 84.84%, maka yang dipilih sebagai kesimpulan hasil diagnosis menurut gejala dan CF pengguna yang dipilih yaitu penyakit *Skizofrenia* Paranoid.

DAFTAR PUSTAKA

- Giarratano, Joseph, Riley, Gary., 2005, *Expert System Principles and Programming*, PWS Publishing Company, Boston.
- Harmon, Paul., King, David, 1985, *Expert System, Artificial Intelligence in Business*, John Wiley and Sons, Inc., New York.
- Hartati. S, Iswanti. S, 2008, *Sistem Pakar & Pengembangannya*. Yogyakarta : Graha Ilmu
- Kusumadewi. S. 2003, *Artificial Intelligence : Teknik dan Aplikasinya*, Graha Ilmu, Yogyakarta.
- Sutojo. T, Mulyanto. F, Suhartono. V, 2010, *Kecerdasan Buatan*, Penerbit Andi, Yogyakarta.

BAB 12

PENERAPAN ALGORITMA BACKWARD CHAINING DALAM SISTEM PAKAR

Oleh Andri Sukmaindrayana

12.1 Sistem Pakar Berbasis Kaidah

Sistem pakar adalah suatu program komputer yang memperlihatkan derajat keahlian dalam pemecahan masalah di bidang tertentu sebanding dengan seorang pakar. Keahlian sistem pakar dalam memecahkan suatu masalah diperoleh dengan cara merepresentasikan pengetahuan seorang atau beberapa orang pakar dalam format tertentu dan menyimpannya dalam basis pengetahuan. Sistem pakar berbasis kaidah (*rule-based expert system*) adalah sistem pakar yang menggunakan kaidah (*rules*) untuk merepresentasikan pengetahuan di dalam basis pengetahuannya.

Mesin inferensi (*inference engine*) merupakan bagian yang bertindak sebagai pencari solusi dari suatu permasalahan berdasar pada kaidah-kaidah yang ada dalam basis pengetahuan sistem pakar. Selama proses inferensi, mesin inferensi memeriksa status dari basis pengetahuan dan memori kerja (*working memory*) untuk menentukan fakta apa saja yang diketahui dan untuk menambahkan fakta baru yang dihasilkan ke dalam memori kerja tersebut. Fakta-fakta yang merupakan hasil dari proses inferensi disimpan dalam memori kerja.

12.2 Strategi Pencarian Pada Mesin Inferensi

Metode inferensi adalah mekanisme berfikir dan pola-pola penalaran yang digunakan oleh sistem untuk mencapai suatu kesimpulan. Metode ini akan menganalisa masalah tertentu dan selanjutnya akan mencari jawaban atau kesimpulan yang terbaik. Penalaran dimulai dengan mencocokkan kaidah-kaidah dalam basis pengetahuan dengan fakta-fakta yang ada dalam basis data.

Ada dua strategi pencarian dasar yang bisa digunakan oleh mesin inferensi dalam mencari kesimpulan untuk mendapatkan solusi bagi permasalahan yang dihadapi sistem pakar, yaitu runut maju (*forward chaining*) dan runut balik (*backward chaining*). Dalam kesempatan ini, penulis membahas mengenai strategi pencarian runut balik (*backward chaining*) :

12.2.1 Runut Balik (*backward chaining*)

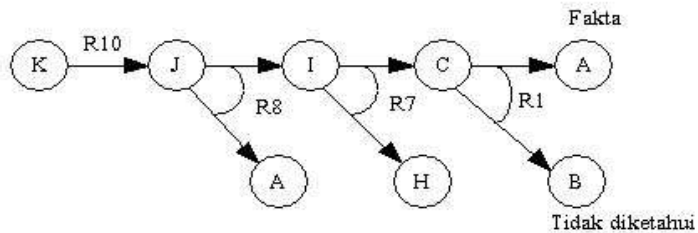
Runut balik merupakan strategi pencarian yang arahnya kebalikan dari runut maju. Proses pencarian dimulai dari tujuan, yaitu kesimpulan yang menjadi solusi permasalahan yang dihadapi. Mesin inferensi mencari kaidah-kaidah dalam basis pengetahuan yang kesimpulannya merupakan solusi yang ingin dicapai, kemudian dari kaidah-kaidah yang diperoleh, masing-masing kesimpulan dirunut balik jalur yang mengarah ke kesimpulan tersebut.

Backward Chaining menggunakan pendekatan *goal-driven*, dimulai dari harapan apa yang akan terjadi (hipotesis) dan kemudian mencari bukti yang mendukung (atau berlawanan) dengan harapan kita. Sering hal ini memerlukan perumusan dan pengujian hipotesis sementara. Jika suatu aplikasi menghasilkan *tree* yang sempit dan cukup dalam, maka gunakan *backward chaining*.

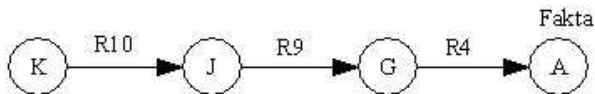
Contoh:

Seperti pada contoh *forward chaining*, terdapat 10 aturan yang sama pada basis pengetahuan dan fakta awal yang diberikan

hanya A dan E. ingin membuktikan apakah K bernilai benar. Proses penalaran *backward chaining* terlihat pada gambar berikut :



Contoh Backward Chaining gagal



Contoh Backward Chaining berhasil

Gambar 12.1. Contoh Backward Chaining

12.2.2 Algoritma Runut Balik

Penggunaan strategi pencarian runut balik untuk membangun mesin inferensi memerlukan suatu algoritma tertentu sehingga bisa diimplementasikan menggunakan bahasa pemrograman. Salah satu algoritma runut balik yang diambil berdasarkan referensi yang penyusun gunakan adalah sebagai berikut :

1. Inisialisasi

a) Buat 3 tabel yaitu :

- 1) Tabel Working Memory, untuk menyimpan pasangan atribut-nilai hasil dari proses inferensi.
- 2) Tabel Goal, untuk menyimpan atribut yang nilainya sedang dicari.
- 3) Tabel Rule/Premise Status, untuk menyimpan nomor kaidah, status dari kaidah yang bersesuaian,

pasangan atribut-nilai dari klausa premis kaidah-kaidah dalam basis pengetahuan, nomor klausa premis dalam kaidah yang bersesuaian, dan status dari klausa premis tersebut.

- b) Semua klausa premis dalam tabel Rule/Premise Status diberi status free.
- c) Semua kaidah dalam tabel Rule/Premise Status diberi status active.

2. Mulai Inferensi

- a) Sebutkan kesimpulan akhir (solusi dari permasalahan) yang ingin dicapai.
- b) Letakkan atribut dari klausa kesimpulan yang merupakan solusi dari permasalahan pada puncak tabel Goal.

3. Pengecekan kaidah

- a) Jika tabel Goal kosong maka STOP.
- b) Jika tabel Goal tidak kosong maka cari kaidah-kaidah berstatus *active* yang atribut dari klausa kesimpulannya bersesuaian dengan atribut yang berada pada puncak tabel Goal.
 - 1) Jika hanya 1 kaidah yang ditemukan, lakukan langkah 6. Jika ada beberapa kaidah yang ditemukan, cari yang berstatus *triggered*, lakukan langkah 6. Jika tidak ada kaidah yang berstatus *triggered*, pilih salah satu kaidah lalu proses dengan langkah 6.
 - 2) Jika tidak ada kaidah yang ditemukan, lakukan langkah 4.

4. Query

Ambil salah satu data dari sekumpulan data yang diberikan oleh pemakai yang bersesuaian dengan atribut yang berada pada puncak tabel Goal.

- a) Jika tidak ada maka STOP.
- b) Jika ada maka pindahkan atribut yang berada pada puncak tabel Goal lalu tempatkan pada tabel Working Memory beserta nilainya, yaitu data yang diambil tadi.

5. Pembaharuan Status Kaidah/Premis

- a) Gunakan isi dari tabel Working Memory untuk membaharui tabel Rule/Premise Status.
- b) Jika ada klausa premis yang berstatus *false* pada suatu kaidah maka beri status *discard* pada kaidah tersebut. Tetapi jika seluruh klausa premis pada suatu kaidah semuanya berstatus *true* maka beri status *triggered* pada kaidah tersebut.
- c) Kembali ke langkah 3.

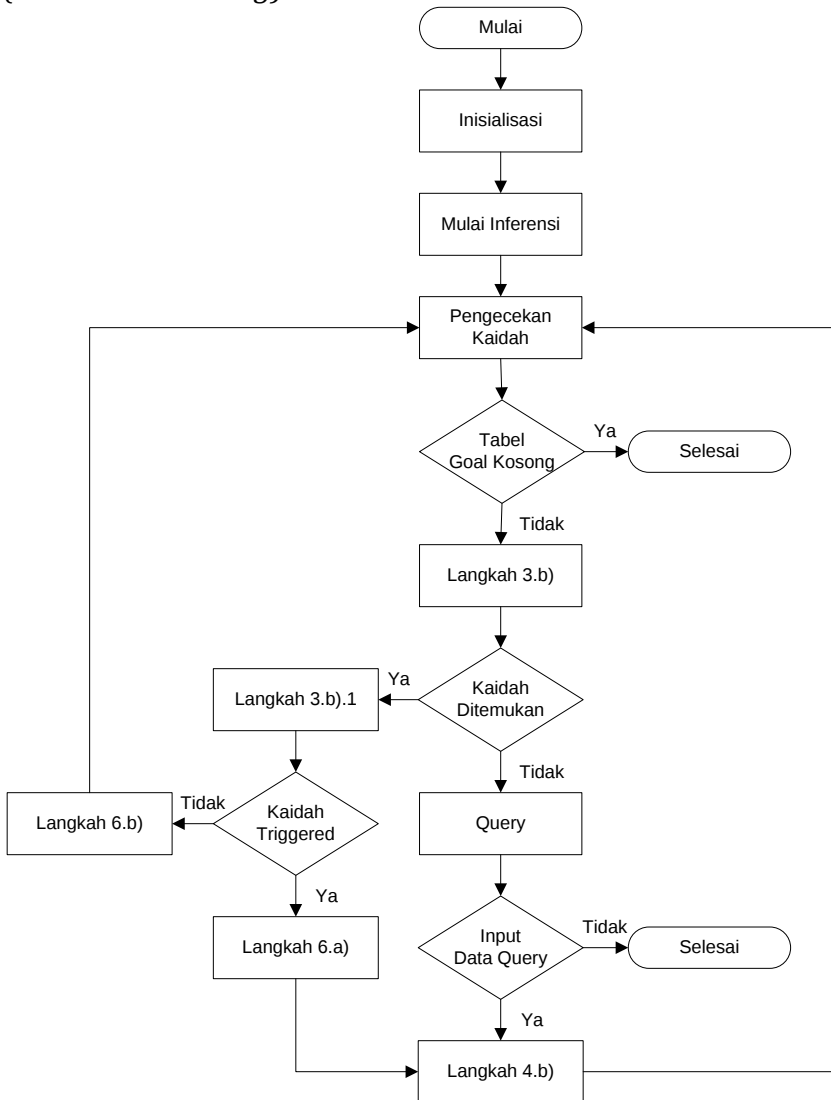
6. Evaluasi Kaidah

- a) Jika kaidah berstatus *triggered* maka pindahkan atribut yang berada pada puncak tabel Goal ke tabel Working Memory beserta nilainya yang terdapat dalam klausa kesimpulan pada kaidah tersebut. Ubah status kaidah tersebut menjadi *fired*. Kembali ke langkah 5.
- b) Jika kaidah tidak berstatus *triggered* maka pilih atribut dari klausa premis pertama yang berstatus *free* lalu tempatkan pada puncak tabel Goal. Kembali ke langkah 3.

Algoritma runut balik di atas mempunyai 2 keadaan yang menyebabkan proses inferensi berhenti, yaitu :

1. Pada saat tabel Goal kosong, berarti kesimpulan yang merupakan solusi dari permasalahan sudah diperoleh.
2. Pada saat data dari pemakai yang diminta oleh sistem pakar untuk memenuhi prompt kaidah tidak ada, berarti kesimpulan yang merupakan solusi dari permasalahan tidak ditemukan.

Berikut ini adalah diagram alir algoritma runut balik (*backward chaining*) :



Gambar 12.2. Diagram Alir Algoritma Runut Balik (*Backward Chaining*)

12.3 Analisis dan Perancangan Penerapan Algoritma Runut Balik (*Backward Chaining*)

Penerapan algoritma Runut Balik (*Backward Chaining*) yang disusun oleh penulis adalah mengenai Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala. Adapun tahapan-tahapan analisis dan perancangan sistem ini adalah sebagai berikut :

12.3.1 Analisis Sistem

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining adalah sistem yang bisa mendeteksi kemungkinan penyakit apa yang diderita pasien sesuai dengan pertanyaan-pertanyaan gejala penyakit yang dialami pasien tersebut. Bagi si pengguna sistem dalam melakukan pendagnosaan kemungkinan penyakit hanya menjawab beberapa pertanyaan gejala dengan jawaban 'Ya' atau 'Tidak' saja yang selanjutnya akan dicocokkan dengan gejala yang memenuhi beberapa diagnosa dari kemungkinan penyakit yang dialami oleh pasien tersebut.

Data-data masukan yang akan dianalisis tersebut adalah sebagai berikut :

1. Data Pasien

Data pasien digunakan untuk menampilkan identitas yang melakukan pendagnosaan dengan data-data sebagai berikut :

Tabel 12.1. Data Pasien

No	Nama Kolom Data
1.	No. Registrasi
2.	Nama Pasien
3.	Usia Pasien
4.	Agama Pasien
5.	Pekerjaan Pasien

No	Nama Kolom Data
6.	Nama Suami
7.	Usia Suami
8.	Agama Suami
9.	Pekerjaan Suami
10.	Tanggal Periksa
11.	Anamnesa
12.	Diagnosa
13.	Terapi

2. Data Penyakit

Data penyakit digunakan untuk menampung kemungkinan-kemungkinan penyakit-penyakit yang akan didiagnosa, meskipun dalam batasan permasalahan ini hanya ada tiga jenis penyakit yang akan didiagnosa, yaitu kanker serviks, kista dan mioma.

3. Data Gejala

Data gejala, dalam hal ini adalah gejala penyakit merupakan proses pendataan berupa tanda-tanda atau gejala yang mungkin menjadi penyebab dari suatu penyakit tertentu. Penulis mendapatkan data-data gejala penyakit ini bersumber dari pakar dan data lainnya diambil dari referensi lain.

4. Data Pengetahuan

Analisis Data Pengetahuan adalah data yang menampung pengetahuan mengenai kemungkinan gejala-gejala apa saja yang mendukung pada suatu penyakit tertentu.

Berikut ini adalah daftar acuan untuk data pengetahuan dalam sistem yang penyusun buat :

Tabel 12.2. Data Gejala Penyakit

No	Gejala	Penyakit		
		Serviks	Kista	Miom a
1.	Alami anemia			*
2.	Alami perdarahan didalam jumlah banyak baik pada waktu haid atau di luar periode haid			*
3.	Alami rasa nyeri layaknya berlangsung infeksi pada rahim			*
4.	Alami tekanan dibagian panggul			*
5.	Basah atau keluar darah pada vagina yang kental dan berbau	*		
6.	Berat badan tidak stabil	*		
7.	Berat badan yang menurun secara drastis		*	
8.	Bila diraba, sisi bawah perut di dekat rahim dapat merasa kenyal			*
9.	Cembelit			*
10.	Depresi		*	
11.	Kerontokan rambut dan rasa lemas di seluruh bagian tubuh		*	
12.	Kesehatan dan nafsu makan menurun		*	
13.	Mengalami keputihan yang tidak normal disertai dengan perdarahan dan jumlahnya berlebih	*		
14.	Mengalami ketidak suburan		*	
15.	Mengalami perdarahan spontan	*		
16.	Mengalami rasa sakit pada bagian paha atau salah satu paha mengalami bengkak	*		

No	Gejala	Penyakit		
		Serviks	Kista	Miom a
17.	Mengalami sakit saat buang air kecil	*		
18.	Menstruasi yang tidak teratur		*	
19.	Merasakan sakit atau merasakan ada tekanan ketika buang air kecil dan buang air besar		*	
20.	Merasakan sakit di panggul dalam jangka waktu yang panjang selama periode menstruasi		*	
21.	Merasakan sakit di panggul setelah olahraga berat ataupun setelah berhubungan seksual		*	
22.	Merasakan sakit ketika berhubungan seksual		*	*
23.	Merasakan sakit pada area vagina dan terjadi bercak-bercak pendarahan di vagina		*	
24.	Merasakan seperti ada tekanan di perut bagian bawah atau di sekitar panggul yang semakin lama semakin terasa		*	
25.	Mual-mual dan muntah		*	
26.	Nafsu makan menjadi sangat berkurang	*		
27.	Nyeri punggung atau kaki			*
28.	Pada saat menstruasi, darah yang keluar dalam jumlah banyak dan berlebih	*		
29.	Pendarahan		*	

No	Gejala	Penyakit		
		Serviks	Kista	Miom a
30.	Perdarahan menstruasi yang hebat			*
31.	Periode menstruasi yang berkepanjangan			*
32.	Rasakan nyeri di pinggul serta perut	*		*
33.	Saat berhubungan intim selaku merasakan sakit, bahkan sering diikuti oleh adanya perdarahan	*		
34.	Sakit di perut bagian bawah atau di bagian panggul, terkadang terasa sangat menyakitkan secara tiba-tiba		*	
35.	Sakit pada pinggul atau nyeri ketika berhubungan	*		
36.	Sering buang air kecil			*
37.	Sisi perut merasa penuh			*
38.	Sulit mengosongkan kandung kemih			*
39.	Susah untuk buang air kecil	*		

12.3.2 Cara Representasi Pengetahuan

Pada sistem yang menyusun buat, cara yang digunakan dalam merepresentasikan pengetahuan adalah dengan menggunakan aturan sebagai berikut :

Aturan 1:

1.	IF (Gejala = Basah atau keluar darah pada vagina yang kental dan berbau) AND
2.	(Gejala = Berat badan tidak stabil) AND
3.	(Gejala = Mengalami keputihan yang tidak normal disertai dengan perdarahan dan jumlahnya berlebih) AND
4.	(Gejala = Mengalami perdarahan spontan) AND
5.	(Gejala = Mengalami rasa sakit pada bagian paha atau salah satu paha mengalami bengkak) AND
6.	(Gejala = Mengalami sakit saat buang air kecil) AND
7.	(Gejala = Nafsu makan menjadi sangat berkurang) AND
8.	(Gejala = Pada saat menstruasi, darah yang keluar dalam jumlah banyak dan berlebih) AND
9.	(Gejala = Rasakan nyeri di pinggul serta perut) AND
10.	(Gejala = Saat berhubungan intim selaku merasakan sakit, bahkan sering diikuti oleh adanya perdarahan) AND
11.	(Gejala = Sakit pada pinggul atau nyeri ketika berhubungan) AND
12.	(Gejala = Susah untuk buang air kecil)
	THEN
	Serviks

Aturan 2:

1.	IF (Gejala = Berat badan yang menurun secara drastis) AND
2.	(Gejala = Depresi) AND
3.	(Gejala = Kerontokan rambut dan rasa lemas di seluruh bagian tubuh) AND
4.	(Gejala = Kesehatan dan nafsu makan menurun) AND

5.	(Gejala = Mengalami ketidak suburan) AND
6.	(Gejala = Menstruasi yang tidak teratur) AND
7.	(Gejala = Merasakan sakit atau merasakan ada tekanan ketika buang air kecil dan buang air besar) AND
8.	(Gejala = Merasakan sakit di panggul dalam jangka waktu yang panjang selama periode menstruasi) AND
9.	(Gejala = Merasakan sakit di panggul setelah olahraga berat ataupun setelah berhubungan seksual) AND
10.	(Gejala = Merasakan sakit ketika berhubungan seksual) AND
11.	(Gejala = Merasakan sakit pada area vagina dan terjadi bercak-bercak pendarahan di vagina) AND
12.	(Gejala = Merasakan seperti ada tekanan di perut bagian bawah atau di sekitar panggul yang semakin lama semakin terasa) AND
13.	(Gejala = Mual-mual dan muntah) AND
14.	(Gejala = Pendarahan) AND
15.	(Gejala = Sakit di perut bagian bawah atau di bagian panggul, terkadang terasa sangat menyakitkan secara tiba-tiba)
	THEN
	Kista

Aturan 3:

1.	IF (Gejala = Alami anemia) AND
2.	(Gejala = Alami perdarahan didalam jumlah banyak baik pada waktu haid atau di luar periode haid) AND
3.	(Gejala = Alami rasa nyeri layaknya berlangsung infeksi

	<i>pada rahim) AND</i>
4.	<i>(Gejala = Alami tekanan dibagian panggul) AND</i>
5.	<i>(Gejala = Bila diraba, sisi bawah perut di dekat rahim dapat merasa kenyal) AND</i>
6.	<i>(Gejala = Cembelit) AND</i>
7.	<i>(Gejala = Merasakan sakit ketika berhubungan seksual) AND</i>
8.	<i>(Gejala = Perdarahan menstruasi yang hebat) AND</i>
9.	<i>(Gejala = Periode menstruasi yang berkepanjangan) AND</i>
10.	<i>(Gejala = Rasakan nyeri di pinggul serta perut) AND</i>
11.	<i>(Gejala = Sering buang air kecil) AND</i>
12.	<i>(Gejala = Sisi perut merasa penuh) AND</i>
13.	<i>(Gejala = Sulit mengosongkan kandung kemih)</i>
	THEN
	<i>Mioma</i>

Setelah basis pengetahuan dimasukan ke dalam basis data, maka selanjutnya menyusun proses diagnosa yang akan menganalisa masalah sesuai dengan basis pengetahuan yang dimasukan dan selanjutnya mencari hasil identifikasi jenis penyakit atau merupakan kesimpulannya. Pada proses diagnosa ini akan memulai melakukan pelacakan pencocokan antara aturan-aturan dalam basis data dengan data yang dimasukkan dalam proses diagnosa tersebut.

Berdasarkan metode pencarian pada proses diagnosa yang digunakan pada sistem ini yaitu dengan cara alur mundur atau *Backward Chaining*, maka kinerja dalam proses diagnosa untuk mencapai ke kesimpulan didasarkan pada langkah-langkah sebagai berikut :

1. Memulai penalaran dari sekumpulan hipotesa atau prediksi awal untuk menentukan jenis penyakit yang akan didiagnosa.

2. Menemukan atau memasukan data-data gejala pada penyakit yang didiagnosa.
3. Demikian selanjutnya sampai ditampilkan data hasil dari proses diagnosa tersebut.

Sebagai contoh dilakukan proses diagnosa dengan menggunakan metode *backward chaining*. Dengan prediksi atau hipotesa awal adalah jenis penyakit *serviks*.

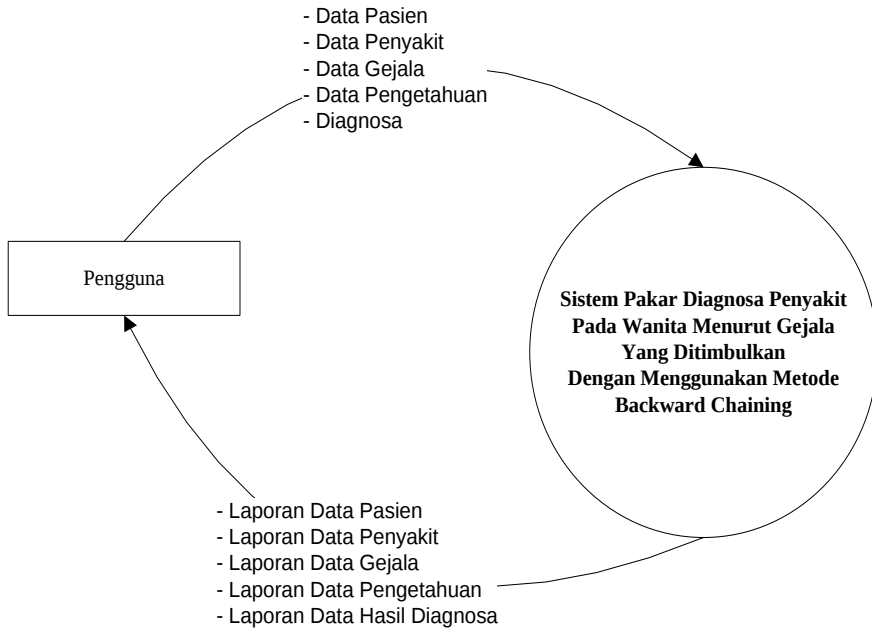
- Langkah 1 *Goal Serviks*
- Langkah 2 *Goal Diketahui*
- Langkah 3 *Temukan aturan dengan goal tersebut, dimulai dari aturan 1*
- Langkah 4 *Lihat jika aturan 1 dengan kondisi 1 "Gejala = Basah atau keluar darah pada vagina yang kental dan berbau" → Ya*
- Langkah 5 *Lihat aturan 1, kondisi 2 "Gejala = Berat badan tidak stabil" → Ya*
- Langkah 6 *Lihat aturan 1, kondisi 3 "Gejala = Mengalami keputihan yang tidak normal disertai dengan perdarahan dan jumlahnya berlebih" → Tidak*
- Langkah 7 *Lompat ke aturan 2, kondisi 1 "Gejala = Berat badan yang menurun secara drastis" → Ya*
- Langkah 8 *Lihat aturan 2, kondisi 2 "Gejala = Depresi" → Tidak*
- Langkah 9 *Lompat ke aturan 3, kondisi 1 "Gejala = Alami anemia" → Tidak*
- Langkah 10 *Lompat ke aturan 1, kondisi 4 "Gejala = Mengalami perdarahan spontan" → Ya*
- Langkah 11 *Lihat aturan 1, kondisi 5 "Gejala = Mengalami rasa sakit pada bagian paha atau salah satu paha mengalami bengkak" → Ya*
- Langkah 12 *Lihat aturan 1, kondisi 6 "Gejala = Mengalami sakit saat buang air kecil" → Ya*

- Langkah 13 Lihat aturan 1, kondisi 7 "Gejala = Nafsu makan menjadi sangat berkurang" → Ya*
- Langkah 14 Lihat aturan 1, kondisi 8 "Gejala = Pada saat menstruasi, darah yang keluar dalam jumlah banyak dan berlebih" → Ya*
- Langkah 15 Lihat aturan 1, kondisi 9 "Gejala = Rasakan nyeri di pinggul serta perut" → Ya*
- Langkah 16 Lihat aturan 1, kondisi 10 "Gejala = Saat berhubungan intim selaku merasakan sakit, bahkan sering diikuti oleh adanya perdarahan" → Ya*
- Langkah 17 Lihat aturan 1, kondisi 11 "Gejala = Sakit pada pinggul atau nyeri ketika berhubungan" → Ya*
- Langkah 18 Lihat aturan 1, kondisi 12 "Gejala = Susah untuk buang air kecil" → Ya*

Kesimpulan

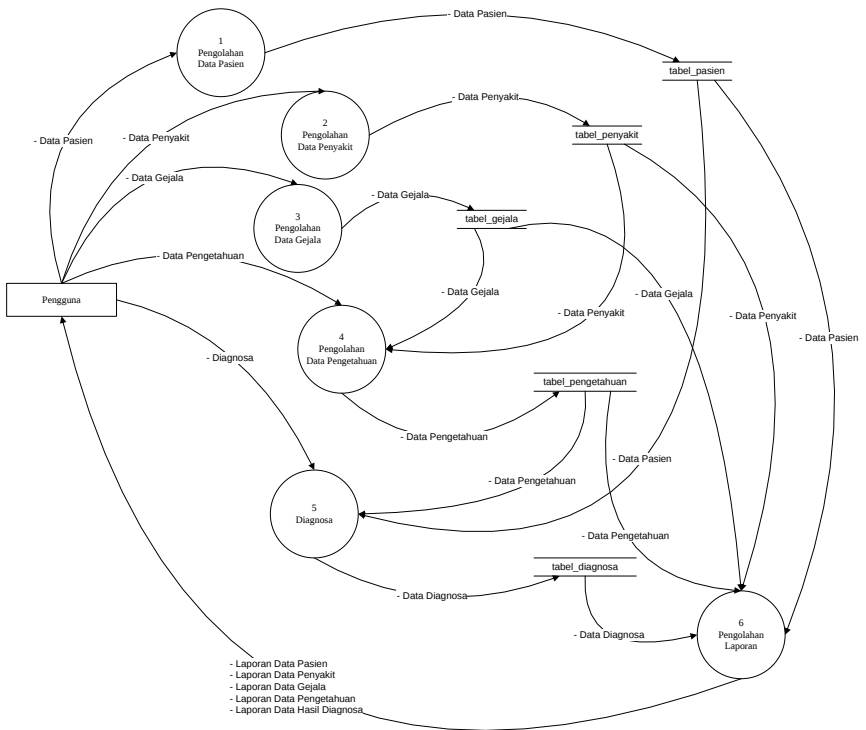
Dari proses diagnosa dengan menggunakan metode *Backward Chaining* seperti disebutkan dalam langkah-langkah diatas maka dapat ditampilkan bahwa nilai presentase dari hipotesa bahwa pasien tersebut memiliki penyakit serviks adalah sebesar 91%. Perhitungan tersebut diambil dari jumlah gejala yang dijawab "Ya" sebanyak 11 gejala dari 12 gejala yang ditimbulkan dari penyakit serviks tersebut.

12.4 Diagram Konteks dari Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining



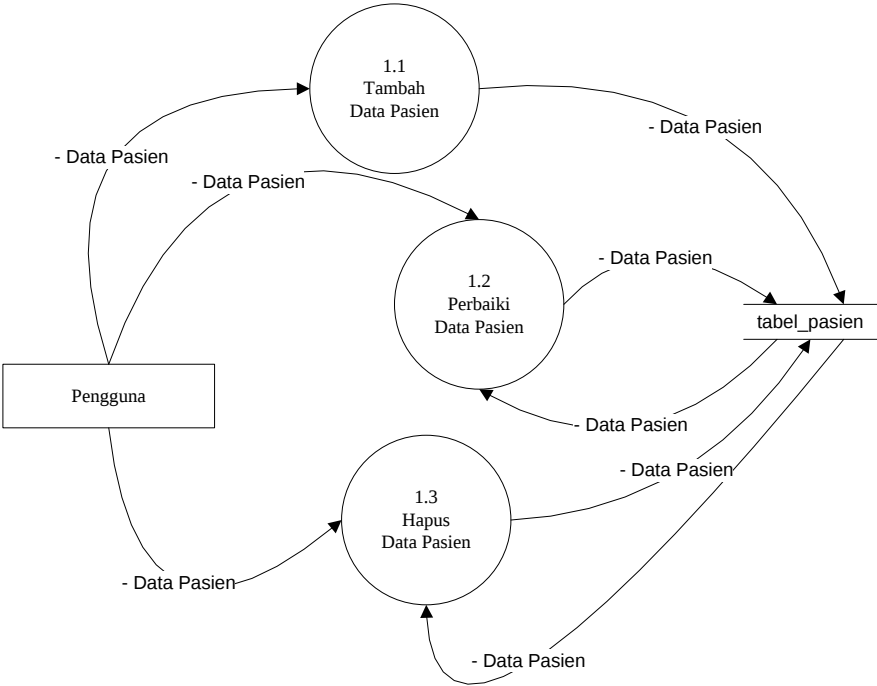
Gambar 12.3. Diagram Konteks dari Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining

12.4.1 DFD Level 1 dari Diagram Konteks Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining



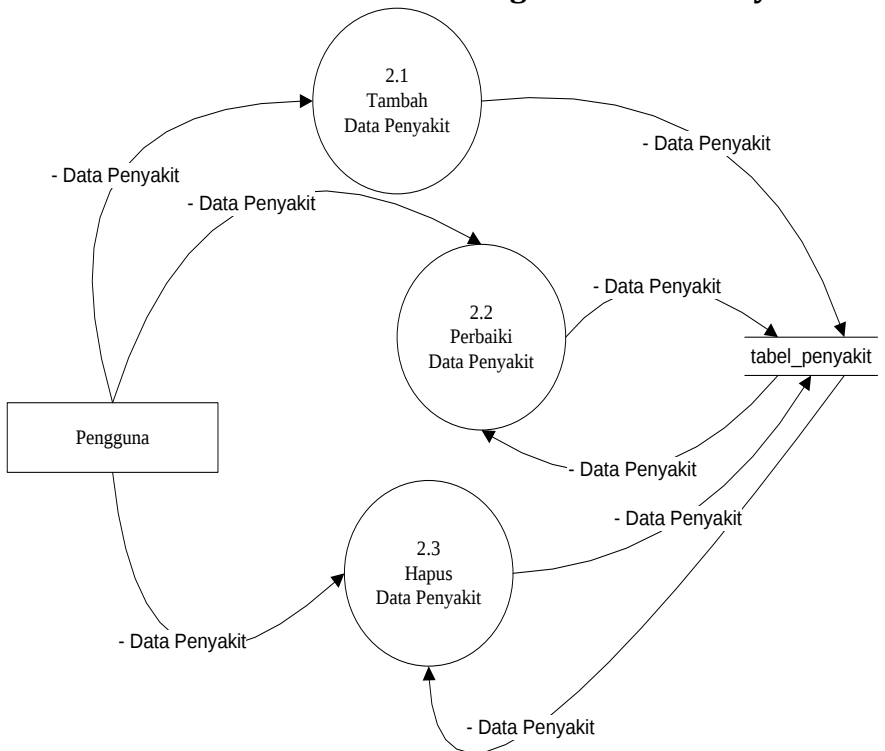
Gambar 12.4. DFD Level 1 dari Diagram Konteks dari Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining

12.4.2 DFD Level 1 dari Proses 1 Pengolahan Data Pasien



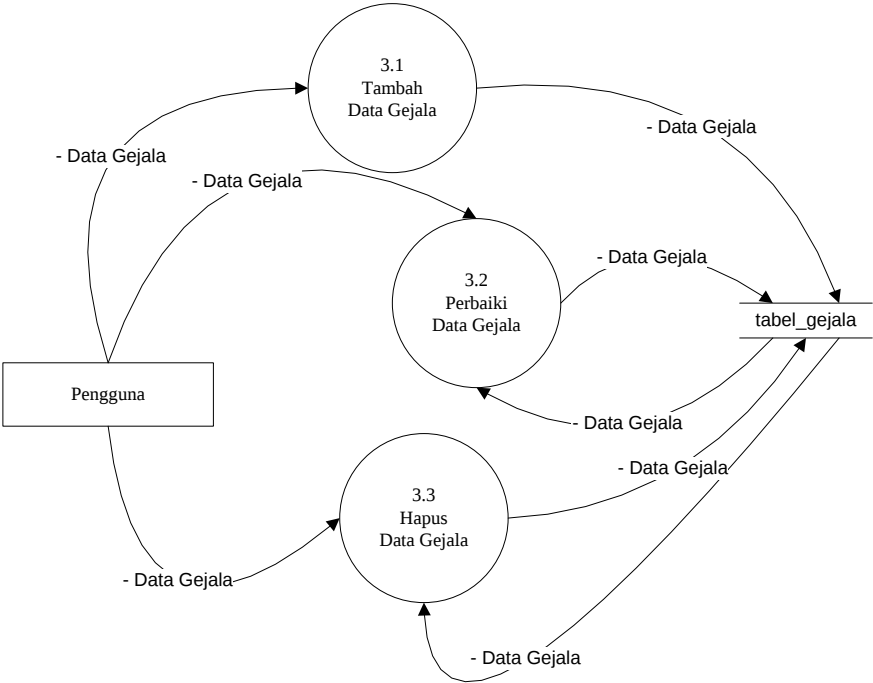
Gambar 12.5. DFD Level 1 dari Proses 1 Pengolahan Data Pasien

12.4.3 DFD Level 1 dari Proses 2 Pengolahan Data Penyakit



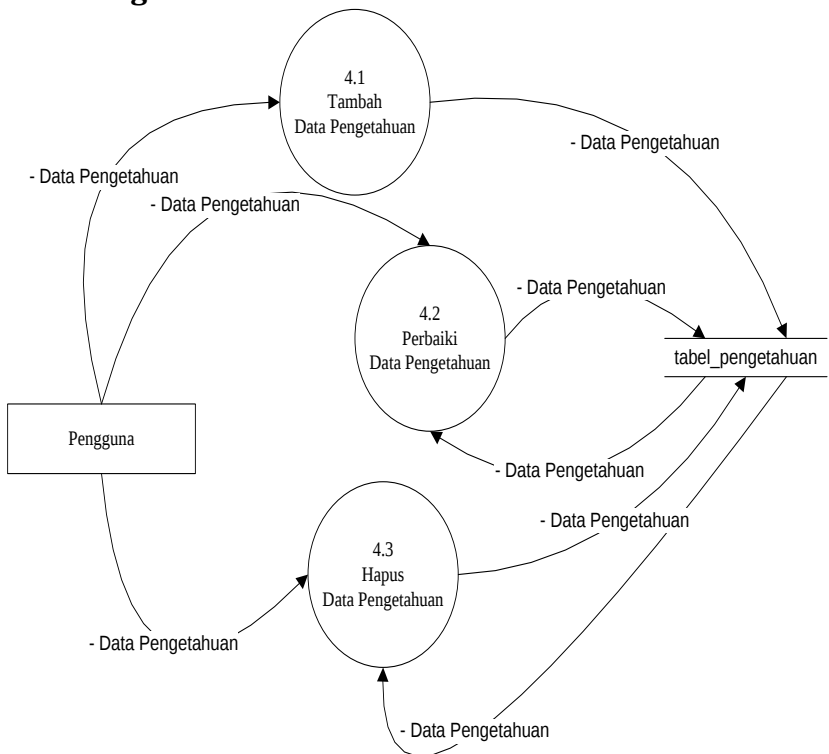
Gambar 12.6. DFD Level 1 dari Proses 2 Pengolahan Data Penyakit

12.4.4 DFD Level 1 dari Proses 3 Pengolahan Data Gejala



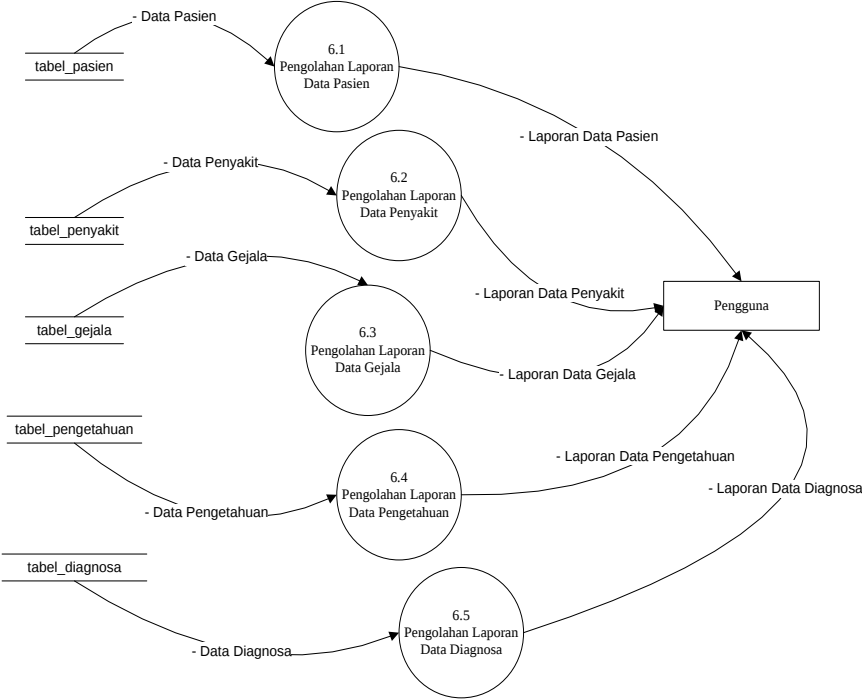
Gambar 12.7. DFD Level 1 dari Proses 3 Pengolahan Data Gejala

12.4.5 DFD Level 1 dari Proses 4 Pengolahan Data Pengetahuan



Gambar 12.8. DFD Level 1 dari Proses 4 Pengolahan Data Pengetahuan

12.4.6 DFD Level 1 dari Proses 6 Pengolahan Laporan



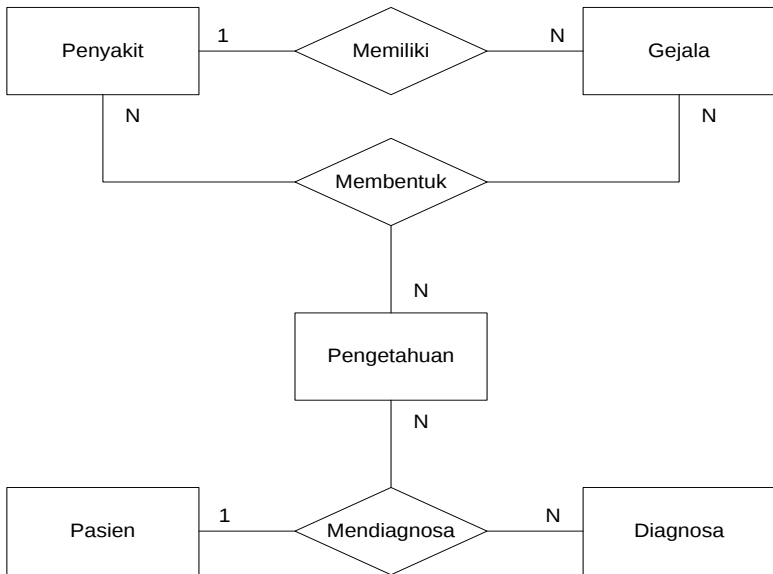
Gambar 12.9. DFD Level 1 dari Proses 6 Pengolahan Laporan

12.7 Rancangan Entity Relationship Diagram (ERD)

Diagram hubungan atau Entitas dikenal dengan sebutan Entity Relationship Diagram (ERD) merupakan suatu model jaringan yang menggambarkan rancangan atau susunan data store dari sistem level pemisah yang tinggi. Diagram E-R ini digunakan untuk menggambarkan hubungan antar simpanan data atau store yang terdapat pada DFD.

Adapun gambar rancangan entity relationship diagram yang penulis buat dalam rancangan keterkaitan antar tabel

untuk pembuatan Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining ini adalah sebagai berikut :



Gambar 12.10. Rancangan *Entity Relationship Diagram (ERD)*

Kamus Data :

Pasien	:	(#Kode_Pasien, Nama, Tempat_Lahir, Tanggal_Lahir, Alamat)
Penyakit	:	(#Kode_Penyakit, Nama_Penyakit, Keterangan)
Gejala	:	(#Kode_Gejala, ##Kode_Penyakit, Nama_Gejala, Keterangan)
Pengetahuan	:	(#Kode_Pengetahuan, ##Kode_Penyakit, ##Kode_Gejala, Keterangan)
Diagnosa	:	(#Kode_Diagnosa, ##Kode_Pasien, Diagnosa, Keterangan)

12.8 Rancangan Struktur Tabel Basis Data

Rancangan struktur tabel basis data merupakan spesifikasi dari file-file yang digunakan dalam sistem yang diusulkan dan dijabarkan dalam bentuk tabel dan tabel tersebut terdapat nama tabel, kunci indeks yang digunakan sert baris (*field*) dan kolom (*record*) Rancangan dari tabel-tabel basis data antara lain sebagai berikut:

Tabel 12.3. Tabel Data Pasien

Nama Tabel : tabel_pasien

Kunci Utama : Kode_pasien

No	Nama Field	Tipe	Ukuran	Keterangan
1.	Kode_Pasien	C	4	Kode Pasien, PK, Not Null
2.	Nama	C	30	Nama Pasien
3.	Tempat_Lahir	C	30	Tempat Lahir Pasien
4.	Tanggal_Lahir	D	-	Tanggal Lahir Pasien
5.	Alamat	C	50	Alamat Pasien

Tabel 12.4. Tabel Data Penyakit

Nama Tabel : tabel_penyakit

Kunci Utama : Kode_Penyakit

No	Nama Field	Tipe	Ukuran	Keterangan
1.	Kode_Penyakit	C	3	Kode Penyakit, PK, Not Null
2.	Nama_Penyakit	C	50	Nama Penyakit (Kanker Serviks, Kista, Mioma dan lain-lain)
3.	Keterangan	C	50	Keterangan Singkat Mengenai Penyakit

Tabel 12.5 Tabel Data Gejala

Nama Tabel : tabel_gejala

Kunci Utama : Kode_Gejala

No	Nama Field	Tipe	Ukuran	Keterangan
1.	Kode_Gejala	C	4	Kode Gejala, PK, Not Null
2	Kode_Penyakit	C	3	Kode Penyakit, merupakan kunci tamu (FK) dari data penyakit
3.	Nama_Gejala	C	50	Nama Gejala Untuk Setiap Penyakit
4.	Keterangan	C	50	Keterangan Singkat Mengenai Gejala

Tabel 12.6. Tabel Data Pengetahuan

Nama Tabel : tabel_pengetahuan

Kunci Utama : Kode_Pengetahuan

No	Nama Field	Tipe	Ukuran	Keterangan
1.	Kode_Pengetahuan	C	7	Kode Pengetahuan, PK, Not Null
2.	Kode_Penyakit	C	3	Kode Penyakit, (FK) dari data penyakit
3.	Kode_Gejala	C	4	Kode Gejala, merupakan kunci tamu (FK) dari data gejala
4.	Keterangan	C	50	Keterangan mengenai data pengetahuan

Tabel 12.7. Tabel Data Diagnosa

Nama Tabel : tabel_diagnosa

Kunci Utama : Kode_Diagnosa

No	Nama Field	Tipe	Ukuran	Keterangan
1.	Kode_Diagnosa	C	5	Kode Diagnosa, PK, Not Null
2.	Tanggal	D	-	Tanggal diagnosa
3.	Kode_Pasien	C	4	Kode Pasien, merupakan kunci tamu (FK) dari data pasien yang melakukan diagnosa penyakit
4.	Kode_Penyakit	C	3	Kode Penyakit, merupakan kunci tamu (FK) dari data penyakit sebagai hipotesa diagnosa penyakit
5.	Nilai	N	-	Nilai presentase diagnosa
6.	Keterangan	C	50	Keterangan mengenai hasil diagnosa

12.9 Rancangan Form dan Laporan

12.9.1 Rancangan Form Menu Utama

Rancangan form utama Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining adalah sebagai berikut:

Sistem Pakar Diagnosa Penyakit Pada Wanita
Menurut Gejala Yang Ditimbulkan
Dengan Menggunakan Metode Backward Chaining

Gambar 12.11. Rancangan Form Utama

12.9.2 Rancangan Form Login

Login

Nama

Password

Gambar 12.12. Rancangan Form Login

12.9.3 Rancangan Form Data Pasien

Data Pasien

X

Kode Pasien

P999

Nama

XXXXX

Tempat Lahir

XXXXX

Tanggal Lahir

dd/mm/yyyy

Alamat

XXXXX

Tambah

Simpan

Perbaiki

Hapus

Tutup

Gambar 12.13. Rancangan Form Data Pasien

12.9.4 Rancangan Form Data Penyakit

Data Penyakit

X

Kode Penyakit

T99

Nama Penyakit

XXXXX

Keterangan

XXXXX

Tambah

Simpan

Perbaiki

Hapus

Tutup

Gambar 12.14. Rancangan Form Data Penyakit

12.9.5 Rancangan Form Data Gejala

Data Gejala

X

Kode Gejala

G999

Nama Gejala

XXXXX

Tambah

Simpan

Perbaiki

Hapus

Tutup

Gambar 12.15. Rancangan Form Data Gejala

12.9.6 Rancangan Form Data Pengetahuan

Data Pengetahuan

X

Kode Pengetahuan

H99999

Kode | Nama Penyakit

T99 | XXXXX

| V

Kode | Nama Gejala

G999 | XXXXX

| V

Keterangan

XXXXX

--	--

Tambah

Simpan

Perbaiki

Hapus

Tutup

Gambar 12.16. Rancangan Form Data Pengetahuan

12.9.7 Rancangan Form Diagnosa

Diagnosa

X

Kode Diagnosa

D99999

Tanggal

dd/mm/yyyy

Kode | Nama Pasien

P999 | XXXXX

| V

Kode | Nama Penyakit

T99 | XXXXX

| V

Data Pertanyaan Gejala-Gejala

Diagnosa

Tutup

Gambar 12.17. Rancangan Form Diganosa

12.9.8 Rancangan Form Data Hasil Diagnosa

Hasil Diagnosa

X

Kode Diagnosa

D99999

Tanggal

dd/mm/yyyy

Kode | Nama Pasien

P99 | XXXXX

| V

Kode | Nama Penyakit

T99 | XXXXX

| V

Keterangan

XXXXX

NILAI DIAGNOSA : 99,99%

Cetak

Tutup

Gambar 12.18. Rancangan Form Data Hasil Diagnosa

12.9.9 Rancangan Laporan Data Pasien

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining LAPORAN DATA PASIEN				
Kode Pasien	Nama	Tempat Lahir	Tanggal Lahir	Alamat
P999	XXXXX	XXXXX	dd/mm/yyyy	XXXXX
{Data Tabel}				

Gambar 12.19. Rancangan Laporan Data Pasien

12.9.10 Rancangan Laporan Data Penyakit

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining LAPORAN DATA PENYAKIT		
Kode Penyakit	Nama Penyakit	Keterangan
T999	XXXXX	XXXXX
{Data Tabel}		

Gambar 12.20. Rancangan Laporan Data Penyakit

12.9.11 Rancangan Laporan Data Gejala

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining LAPORAN DATA GEJALA	
Kode Gejala	Nama Gejala
G999	XXXXX
{Data Tabel}	

Gambar 12.21. Rancangan Laporan Data Gejala

12.9.12 Rancangan Laporan Data Pengetahuan

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining LAPORAN DATA PENGETAHUAN		
Kode Nama Penyakit	P99 XXXXX	
Keterangan	XXXXX	
Gejala-Gejala →	Kode Gejala	Nama Gejala
	G999	XXXXX
{Data Tabel}		

Gambar 12.22. Rancangan Laporan Data Pengetahuan

12.9.13 Rancangan Laporan Data Hasil Diagnosa

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan
Dengan Menggunakan Metode Backward Chaining
LAPORAN DATA DIAGNOSA

Kode DiagnosaD99999Tanggal dd/mm/yyyy

Kode | Nama PasienP999 | XXXXX

Kode | Nama PenyakitT99 | XXXXX

Keterangan

XXXXX

NILAI DIAGNOSA : 99,99%

Gambar 12.22. Rancangan Laporan Data Hasil Diagnosa

12.10 Impelementasi Form dan Laporan

Proses pembuatan aplikasi (*coding*) dengan menggunakan Bahasa Pemrograman Borland Delphi Versi 7.0 dan menggunakan basisdata MySQL

12.10.1 Nama Form Yang Dibuat

Tabel 12.8. Nama Form Yang Dibuat

No	Nama Form
1.	FormUtama, form ini merupakan tampilan awal dari sistem ini.
2.	FormLogin, form ini digunakan untuk admin untuk masuk ke menu pada aplikasi ini.
3.	FormPasien, form ini digunakan untuk melakukan pengolahan data pasien.
4.	FormPenyakit, form ini digunakan untuk melakukan pengolahan data penyakit.

No	Nama Form
5.	FormGejala, form ini digunakan untuk melakukan pengolahan data gejala.
6.	FormPengetahuan, form ini digunakan untuk melakukan pengolahan data pengetahuan.
7.	FormDiagnosa, form ini digunakan untuk melakukan proses diagnosa penyakit.
8.	FormHasilDiagnosa, form ini digunakan untuk menampilkan hasil proses diagnosa penyakit.

12.10.2 Nama Tabel Yang Dibuat

Tabel 12.9. Nama Tabel Yang Dibuat

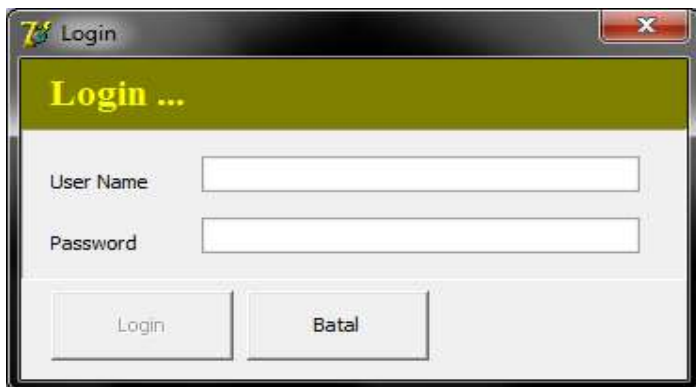
No	Nama Tabel
1.	TabelPasien, tabel ini digunakan untuk menyimpan data pasien.
2.	TabelPenyakit, tabel ini digunakan untuk menyimpan data penyakit.
3.	TabelGejala, tabel ini digunakan untuk menyimpan data gejala.
4.	TabelPengetahuan, tabel ini digunakan untuk menyimpan data pengetahuan.
5.	TabelDiganosa, tabel ini digunakan untuk menyimpan data proses diagnosa.

12.10.3 Form Menu Utama



Gambar 12.23. Form Menu Utama

12.10.4 Form Login



Gambar 12.24. Form Login

12.10.5 Form Data Pasien

Data Pasien ...

Kode Pasien:

Nama:

Tempat Lahir:

Tanggal Lahir:

Alamat:

Kode_Pasien	Nama	Tempat_Lahir	Tanggal_Lahir
P001	NISA	TASIKMALAYA	15/05/2013
P002	NADA	CIAMIS	15/04/2013

Tambah Simpan Perbaiki Hapus Tutup

Gambar 12.25. Form Data Pasien

12.10.6 Form Data Penyakit

Data Penyakit ...

Kode Penyakit:

Nama Penyakit:

Keterangan:

Kode_Penyakit	Nama_Penyakit	Keterangan
T01	CERVIX	Kanker serviks
T02	KISTA	(HEMO)
T03	HIDUNA	(HEMO)

Tambah Simpan Perbaiki Hapus Tutup

Gambar 12.26. Form Data Penyakit

12.10.7 Form Data Gejala

Data Gejala ...

Kode Gejala: **G001**

Nama Gejala: **ALAMI ANEMIA**

Kode_Gejala	Nama_Gejala
G001	ALAMI ANEMIA
G002	ALAMI PERDARAHAN DIDALAM JUMLAH BANYAK BAIK PADA WAKTU HAID
G003	ALAMI RASA NYERI LAYAKNYA BERLANGSUNG INFEKSI PADA RAHIM
G004	ALAMI TEKANAN DIBAGIAN PANGGUL
G005	BASAH ATAU KELUAR DARAH PADA VAGINA YANG KENTAL DAN BERBAU
G006	BERAT BADAN TIDAK STABIL
G007	BERAT BADAN YANG MENURUN SECARA DRASTIS
G008	BILA DIRABA, SISI BAWAH PERUT DI DEKAT RAHIM DAPAT MERASA KENYAL
G009	CEMBELIT
G010	DEPRESI
G011	KERONTOKAN RAMBUT DAN RASA LEMAS DI SELURUH BAGIAN TUBUH
G012	KESEHATAN DAN NAFSU MAKAN MENURUN
G013	MENGALAMI KEPUTIHAN YANG TIDAK NORMAL DISERTAI DENGAN PERDARAHAN DAN JUMLAH
G014	MENGALAMI KETIDAK SUBURAN
G015	MENGALAMI PERDARAHAN SPONTAN

Tambah Simpan Perbaiki Hapus Tutup

Gambar 12.27. Form Data Gejala

12.10.8 Form Data Pengetahuan

Data Pengetahuan ...

Kode Pengetahuan: **H00001**

Kode | Nama Penyakit: **T03 | MIOMA**

Kode | Nama Gejala: **G001 | ALAMI ANEMIA**

Keterangan:

Kode_Pengetahuan	Kode_Penyakit	Nama_Penyakit	Kode_Ge
H00001	T03	MIOMA	G001
H00002	T03	MIOMA	G002
H00003	T03	MIOMA	G003
H00004	T03	MIOMA	G004
H00005	T03	CERVIX	G005
H00006	T03	CERVIX	G006
H00007	T03	KISTA	G007
H00008	T03	MIOMA	G008
H00009	T03	MIOMA	G009
H00010	T03	KISTA	G010
H00011	T03	KISTA	G011

Tambah Simpan Perbaiki Hapus Tutup

Gambar 12.28. Form Data Pengetahuan

12.10.9 Form Diagnosa

Gambar 12.29. Form Diagnosa

12.10.10 Form Hasil Diagnosa

Gambar 12.30. Form Hasil Diagnosa

12.10.11 Laporan Data Pasien

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining

LAPORAN DATA PASIEN

Kode Pasien	Nama	Tempat Lahir	Tanggal Lahir	Alamat
P001	NISA	TASBOHALAYA	15/05/2013	TASBOHALAYA
P002	HANIDA	CIMPEK	15/05/2013	CIMPEK

Gambar 12.38. Laporan Data Pasien

12.10.12 Laporan Data Penyakit

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining

LAPORAN DATA PENYAKIT

Kode Penyakit	Nama Penyakit	Keterangan
T01	CERVIX	Kanker serviks adalah salah satu tumor ganas yang sering ditemukan pada kaum wanita, serta merupakan satu-satunya kanker yang tidak diketahui penyebabnya. Kanker serviks disebabkan oleh virus HPV, dan virus HPV dapat menyebar melalui kontak fisik, pada 10 tahun lebih masa inkubasi, tetapi kanker serviks dapat diobati dan dapat diukur, dengan memahami gejala kanker serviks, dan dengan tespek melakukan deteksi dan diketahui sejak dini, dengan dapat pola deteksi.
T02	KISTA	Kista adalah tumor jinak di yang paling sering ditemui. Biasanya kista, berisi cairan kental, dan ada juga yang berbentuk anggur. Kista juga ada yang berisi lemak, cairan, rambut, ataupun bahan-bahan lainnya. Kista termasuk tumor jinak yang berkembang seperti jaringan lemak. Kumpulan sel-sel tumor itu bertumbuh dengan jaringan normal di sekitarnya dan tidak dapat menyebar ke bagian tubuh lain. Stadi sel-sel tumor jinak relatif mudah diobati dengan jalan pembedahan, dan tidak membahayakan kesehatan penderitanya.
T03	MENOPA	Menopause adalah tumor jinak yang tumbuh dari rahim dan sering muncul selama masa subur. Menopause tidak terkait dengan kanker rahim dan tumor tidak pernah berkembang menjadi kanker. Sebanyak 3 dari 4 wanita memiliki menopause alami, dan biasanya ditemukan pada saat dilakukan pemeriksaan dan di saat sebelum melahirkan.

Gambar 12.39. Laporan Data Penyakit

12.10.13 Laporan Data Gejala

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining

LAPORAN DATA GEJALA

Kode Gejala	Nama Gejala
G001	ALAMI ANEMIA
G002	ALAMI PERDARAHAN DALAM JUMLAH BANYAK BAK PADA WAKTU HAI
G003	ALAMI RASA NYERI LAYANAN BERLANGSUNG INFESI PADA RAHM
G004	ALAMI TEKANAN DEBAGIAN PANGGUL
G005	BASAH ATAU KELUAR DARAH PADA VAGINA YANG KENTAL DAN BERBAU
G006	BERAT BADAN TIDAK STABIL
G007	BERAT BADAN YANG MENURUN SECARA DRASTIS
G008	DELA DERADA, SESI BAWAH PERUT DI DOKAT RAHM DAPAT MERASA KENYAL
G009	CEMBELIT
G010	DEYUSE
G011	KERONTOKAN RAMBUT DAN RASA LEMAS DI SELURUH BAGIAN TUBUH
G012	KESIHATAN DAN NAFSU MAKAN MENURUN
G013	MENGALAMI SPERITUSAL YANG TIDAK NORMAL, SPERITUSAL TERJADI PERUBAHAN DAN SAKIT AKHIR SPERITUS

Gambar 12.40. Laporan Data Gejala

12.10.14 Laporan Data Pengetahuan

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan Dengan Menggunakan Metode Backward Chaining

LAPORAN DATA PENGETAHUAN

Kode | Nama Penyakit: T01 | MIONIA

Keterangan:

Gejala-Gejala -->

Kode Gejala

Nama Gejala

G001

ALAMI ANEMIA

G002

ALAMI PERDARAHAN DALAM JUMLAH BANYAK BAK PADA WAKTU HAI

G003

ALAMI RASA NYERI LAYANAN BERLANGSUNG INFESI PADA RAHM

G004

ALAMI TEKANAN DEBAGIAN PANGGUL

Kode | Nama Penyakit: T01 | CERVIX

Keterangan:

Gejala-Gejala -->

Kode Gejala

Nama Gejala

G005

BASAH ATAU KELUAR DARAH PADA VAGINA YANG KENTAL DAN BERBAU

G006

BERAT BADAN TIDAK STABIL

Kode | Nama Penyakit: T01 | KISTA

Keterangan:

Gambar 12.41. Laporan Data Pengetahuan

12.10.15 Laporan Data Hasil Diagnosa

Sistem Pakar Diagnosa Penyakit Pada Wanita Menurut Gejala Yang Ditimbulkan
Dengan Menggunakan Metode Backward Chaining

LAPORAN DATA HASIL DIAGNOSA

Tanggal	05/01/2019
Kode / Nama Pasien	P001 / NISA
Kode / Nama Penyakit	T01 / CERATK

Nilai Diagnosa : 81,33%

Keterangan:

Kanker serviks adalah salah satu tumor ganas yang sering ditemukan pada kaum wanita, serta merupakan satu-satunya kanker yang bisa diketahui penyebabnya. Kanker serviks bisa disebabkan oleh virus HPV, dan virus HPV dapat menyebar melalui kontak fisik, pada 10 tahun setelah masa infeksi, tetapi kanker serviks dapat dicegah dan dapat diukur, dengan memahami gejala kanker serviks, dan dengan cepat melakukan deteksi dan diketahui sejak dini, dengan cepat pada dokter.

Gambar 12.42. Laporan Data Hasil Diagnosa

DAFTAR PUSTAKA

- Darmansyah, Ilham Chiruddin, Tomy Nada Putra. 2021. Perancangan Sistem Pakar. Jurnal Teknik Informatika dan Sistem Informasi. Pekanbaru : Sekolah Tinggi Teknologi Pekanbaru.
- Didik Setiyadi, S.kom., M. Kom. 2020. SISTEM BASIS DATA DAN SQL. In *Mitra Wacana Media*.
- Fathansyah. 2018. *BASIS DATA*. Bandung: Informatika Bandung.
- Subhan, Mohamad. 2012. *Analisa Perancangan Sistem*. Jakarta: Lentera Ilmu Cendeki
- <http://gejalapenyakitmu.blogspot.com/2013/04/gejala-kanker-serviks-penyebab-dan.html>, Diakses pada hari Senin, tanggal 19 Juni 2023, Pukul 23.07 WIB.

BAB 13

PENERAPAN METODE INFERENSY FUZY DALAM SISTEM PAKAR

Oleh Raimon Efendi

13.1 Pendahuluan

Metode inferensi fuzzy pertama kali diperkenalkan oleh Lotfi A. Zadeh pada tahun 1965. Lotfi A. Zadeh adalah seorang ilmuwan komputer dan matematikawan asal Azerbaijan yang merupakan salah satu tokoh utama dalam pengembangan konsep logika fuzzy (Arisandi and Sari, 2021; Galang Persada Nurani Hakim *et al.*, 2021). Lotfi A. Zadeh memperkenalkan konsep logika fuzzy sebagai suatu perluasan dari logika boolean yang memungkinkan representasi dan penalaran tentang ketidakpastian dan kesamaran dalam sistem komputer. Kontribusinya terhadap logika fuzzy membuka jalan bagi pengembangan metode inferensi fuzzy dalam sistem pakar dan berbagai aplikasi lainnya.

Selama beberapa dekade berikutnya, banyak peneliti dan ilmuwan komputer lainnya terus mengembangkan dan memperbaiki metode inferensi fuzzy dalam sistem pakar (Sulaiman *et al.*, no date). Konsep logika fuzzy dan metode inferensi fuzzy telah menjadi landasan bagi banyak sistem pakar yang menggabungkan aspek kecerdasan buatan dengan pemodelan ketidakpastian dan pengetahuan berbasis aturan (Kandel, 1991). Dengan penemuan dan kontribusi Lotfi A. Zadeh dalam bidang logika fuzzy, ia dianggap sebagai salah satu penemu metode inferensi fuzzy dan pemimpin dalam pengembangan konsep tersebut.

Perkembangan metode inferensi fuzzy dalam sistem pakar terus berlanjut seiring dengan kemajuan dalam bidang kecerdasan buatan dan logika fuzzy. Beberapa perkembangan terkini dalam metode inferensi fuzzy dalam sistem pakar meliputi:

1. Metode inferensi fuzzy telah diintegrasikan dengan algoritma pembelajaran mesin, seperti jaringan saraf tiruan dan pembelajaran mendalam (*deep learning*). Hal ini memungkinkan sistem pakar untuk belajar secara otomatis dari data dan meningkatkan kemampuan penalarannya seiring berjalannya waktu.
2. Metode inferensi fuzzy juga telah digabungkan dengan metode inferensi non-fuzzy, seperti logika probabilistik atau logika ketidakpastian. Pendekatan ini memungkinkan sistem pakar untuk menggabungkan kelebihan dari kedua paradigma tersebut dan meningkatkan ketepatan dan keandalan penalaran.
3. Metode inferensi fuzzy telah dikombinasikan dengan teknik penggalian pengetahuan dari data (*data mining*) untuk mengidentifikasi pola atau hubungan yang tersembunyi dalam data. Ini memungkinkan sistem pakar untuk menggunakan informasi tambahan yang ditemukan dari data untuk meningkatkan kualitas penalaran dan rekomendasi.
4. Selain tingkat keanggotaan fuzzy, metode inferensi fuzzy telah diperluas untuk mempertimbangkan interval fuzzy. Hal ini memungkinkan sistem pakar untuk mengatasi ketidakpastian dengan lebih baik dan menghasilkan solusi yang lebih konservatif.
5. Metode inferensi fuzzy telah diintegrasikan dengan teknologi canggih seperti *Internet of Things* (IoT) dan analisis big data. Hal ini memungkinkan sistem pakar untuk memanfaatkan data yang besar dan real-time untuk penalaran dan pengambilan keputusan yang lebih akurat.

Perkembangan-perkembangan ini terus mendorong kemajuan metode inferensi fuzzy dalam sistem pakar, meningkatkan kemampuan sistem pakar dalam menghadapi ketidakpastian, kompleksitas, dan perubahan dalam lingkungan yang dihadapi.

13.2 Konsep Fuzzy Inference System

Logika fuzzy adalah perluasan atau generalisasi dari logika klasik yang hanya memiliki dua nilai keanggotaan, yaitu 0 dan 1. Dalam logika fuzzy, nilai kebenaran suatu pernyataan dapat berkisar dari sepenuhnya benar hingga sepenuhnya salah. Konsep ini memungkinkan pemodelan ketidakpastian dan kesamaran dalam penalaran (Thukral and Bal, 2019). Dengan menggunakan teori himpunan fuzzy, suatu objek dapat menjadi anggota dari banyak himpunan dengan derajat keanggotaan yang berbeda. Ini berbeda dengan teori himpunan klasik yang hanya mengizinkan objek menjadi anggota atau bukan anggota dari suatu himpunan secara tegas.

Logika fuzzy digunakan untuk menerjemahkan besaran yang diekspresikan dalam bahasa atau istilah linguistik. Misalnya, dalam konteks kecepatan kendaraan, besaran tersebut dapat diekspresikan dengan kata-kata seperti "pelan", "agak cepat", "cepat", dan "sangat cepat". Logika fuzzy memungkinkan kita untuk mengukur derajat keanggotaan sebuah nilai kecepatan dalam himpunan-himpunan ini dengan tingkat ketepatan yang lebih baik daripada logika klasik yang hanya mengizinkan nilai kecepatan tersebut menjadi benar atau salah secara tegas.

Regenerate response Konsep inferensi fuzzy dalam sistem inferensi fuzzy melibatkan penggunaan aturan fuzzy untuk mendapatkan hasil atau kesimpulan berdasarkan input yang diberikan. Inferensi fuzzy melibatkan perbandingan dan kombinasi

himpunan fuzzy yang relevan untuk menghasilkan output fuzzy yang kemudian dikonversi menjadi nilai crisp.

Berikut adalah beberapa konsep penting dalam inferensi fuzzy:

1. *Fuzzy Rules (Aturan Fuzzy):*

Aturan fuzzy terdiri dari pernyataan logika fuzzy yang mendefinisikan hubungan antara himpunan fuzzy input dan himpunan fuzzy output. Setiap aturan memiliki dua bagian: pendahulu (antecedent) dan konsekuensi (consequent). Pendahulu aturan menyatakan kondisi berdasarkan nilai input yang akan mengaktifkan aturan tersebut. Konsekuensi aturan menyatakan kontribusi atau implikasi terhadap himpunan fuzzy output.

2. *Rule Evaluation (Evaluasi Aturan):*

Dalam evaluasi aturan, setiap aturan fuzzy dinilai atau dievaluasi berdasarkan sejauh mana kondisi inputnya cocok dengan nilai input yang diberikan. Evaluasi aturan melibatkan perbandingan nilai input dengan fungsi keanggotaan yang didefinisikan dalam aturan untuk mengukur tingkat kesesuaian.

3. *Degree of Activation (Tingkat Aktivasi):*

Tingkat aktivasi merupakan ukuran sejauh mana aturan fuzzy terpenuhi atau aktif berdasarkan kondisi input yang diberikan. Tingkat aktivasi dapat dihitung menggunakan fungsi keanggotaan atau metode lain, seperti minimum, maksimum, atau rata-rata tertimbang.

4. *Aggregation (Agregasi):*

Agregasi melibatkan kombinasi himpunan fuzzy yang teraktivasi berdasarkan tingkat aktivasi mereka. Operasi logika fuzzy seperti "AND" atau "OR" digunakan untuk menggabungkan himpunan fuzzy yang relevan dan menghasilkan keluaran fuzzy komposit.

5. *Defuzzification (Defuzzifikasi):*

Defuzzifikasi adalah proses mengubah himpunan fuzzy yang teraktivasi menjadi nilai crisp yang dapat dipahami. Metode defuzzifikasi, seperti mencari pusat massa (*centroid*) atau rata-rata tertimbang, digunakan untuk menghasilkan nilai crisp yang merepresentasikan hasil inferensi fuzzy.

Dengan menggunakan konsep-konsep ini, sistem inferensi fuzzy dapat memproses input fuzzy, mengaktifkan aturan fuzzy yang relevan, menggabungkan himpunan fuzzy, dan menghasilkan output fuzzy yang kemudian dikonversi menjadi nilai crisp. Konsep inferensi fuzzy memungkinkan pemodelan dan penalaran berbasis logika fuzzy, yang dapat digunakan dalam berbagai aplikasi untuk mengatasi ketidakpastian dan keambiguan dalam data.

Logika fuzzy didasarkan pada teori himpunan fuzzy, yang merupakan perluasan dari teori himpunan klasik (*crisp*). Konsep logika fuzzy muncul pada tahun 1980an dengan penerapannya pada sistem kontrol. Pada dasarnya, logika fuzzy adalah metode pemetaan ruang input ke ruang output yang sesuai. Ada berbagai cara untuk menetapkan pemetaan input-ke-output ini, seperti melalui sistem linier, jaringan saraf, dan persamaan diferensial. Meskipun ada metode alternatif selain logika fuzzy, namun dianggap memberikan solusi terbaik karena menawarkan keunggulan kecepatan dan biaya, menjadikannya pendekatan yang lebih cepat dan ekonomis. (Kusumadewi and Purnomo, 2004).

13.3 Penerapan Fuzzy Inference System (Studi Kasus pada pemberian Kredit Perbankan)

Penggunaan logika fuzzy dalam tesis ini bertujuan supaya data – data calon debitur yang mengajukan permohonan kredit, bisa dijadikan *input* dan diproses untuk menghasilkan suatu *output*. Data - data calon debitur tersebut mempunyai beberapa atribut yang di ambil dari data persyaratan pengajuan permohonan

kredit. Atribut-atribut tersebut yaitu; atribut *plafond*, atribut *character*, atribut *capacity*, atribut *capital*, atribut *condition*, dan atribut *collateral*.

Proses selanjutnya adalah penganalisaan data yang telah didapat, sehingga data tersebut dapat dikelompokkan menjadi data yang dapat diproses dengan merancang rule-rule menggunakan fuzzy logic. Bank sebagai pihak pemberi kredit (pinjaman) mempunyai hak untuk melakukan proses penyaringan kepada pihak calon debitur yang layak, Proses penyaringan ini sekaligus untuk memperoleh data yang lengkap dan valid tentang calon debitur. Bank harus dapat mengendalikan resiko kredit yang diberikannya. Untuk itu, bank mengembangkan suatu proses seleksi untuk menyaring setiap proposal kredit yang masuk. Melalui proses penyaringan tersebut diharapkan kredit yang diberikan adalah kredit dengan kualitas pengembalian yang terjamin.

Dalam penilaian layak atau tidak suatu kredit disalurkan maka perlu dilakukan suatu penilaian kredit. Penilaian kelayakan suatu kredit dapat dilakukan dengan menggunakan analisis 5 C's namun untuk kredit yang lebih besar jumlahnya perlu dilakukan metode penilaian dengan studi kelayakan. Dalam studi kelayakan ini setiap aspek dinilai apakah memenuhi syarat atau tidak. Adapun aspek-aspek yang dinilai dalam pemberian suatu fasilitas kredit adalah :

1. *Character* (watak), maka penilaiannya meliputi Riwayat hidup calon debitur, mencakup reputasi calon debitur di lingkungan bisnis/usahanya dan riwayat hubungan calon debitur dengan Bank, dimana disini hubungan dengan atau hubungan dengan Bank lain.
2. *Capacity* (kemampuan), penilaiannya meliputi pengalaman dari calon debitur dalam mengelola usahanya, termasuk sumber daya manusia yang dimilikinya.

3. *Capital* (modal), penilaiannya keuangan perusahaan yang dilihat dari laporan keuangan yaitu Neraca dan Laporan Rugi dan Laba 3 tahun terakhir.
4. *Collateral* (jaminan), penilaian meliputi barang jaminan yang diserahkan calon debitur kepada Bank sebagai jaminan atas kredit yang diterimanya.

Analisa data pada penyusunan tesis ini dilakukan dengan menggunakan logika *fuzzy*. Dimana logika *fuzzy* adalah logika yang dapat mempresentasikan keadaan linguistik seperti rendah, sedang, tinggi dan sebagainya. Oleh karena itu *fuzzy logic* memungkinkan nilai keanggotaan antara 0 sampai dengan 1 atau sesuatu dapat dinyatakan sebagian benar dan sebagian salah pada saat yang sama. *Fuzzy logic* juga mempunyai tingkat toleransi pada data yang diolah tanpa mengabaikan kaidah dari data tersebut.

Tabel 13.1. Himpunan *fuzzy* Data Calon debitur kredit

Fungsi	Variabel	Nama	Semesta	Domain
		Himpunan <i>Fuzzy</i>	Pembicaraan	
<i>Input</i>	<i>Plafond</i>	Rendah	0 – 100	0 – 40
		Sedang		20 – 80
		Tinggi		60- 100
	<i>Character</i>	Jelek	0 – 100	0 – 50
		Cukup		30 – 80
		Baik		60 – 100
	<i>Capital</i>	Miskin	0 – 100	0 – 40
		Sederhana		20 - 70
		Kaya		50 - 100
	<i>Collateral</i>	Kurang	0 – 120	0 - 60
		Cukup		40 - 100
		Lebih		80 - 120
	Persentase	PKP 100%	0 – 100	

Fungsi <i>Output</i>	Variabel	Nama	Semesta	Domain
		Himpunan <i>Fuzzy</i>	Pembicaraan	
	Keputusan	PKP 90%		
	Pencairan	PKP 80%		
	(PKP)	PKP 70%		
		PKP 60%		
		PKP 50%		
		PKP 40%		
		PKP 30%		
		PKP 20%		
		PKP 10%		
		PKP 0%		

Identifikasi data untuk menentukan kelayakan pemberian kredit kepada calon debitur dilakukan dengan menentukan variabel yang diperlukan dalam pengolahan dan analisis data. Variabel yang harus dipenuhi, diambil dari data persyaratan kredit dan penilaian lapangan yang dilakukan *account officer*, variabel-variabel tersebut antara lain, jumlah *plafond* yang diajukan, penilaian *character* calon debitur, penilaian *capital* yang dimiliki, dan penilaian jaminan (*collateral*) yang diajukan sebagai agunan kredit.

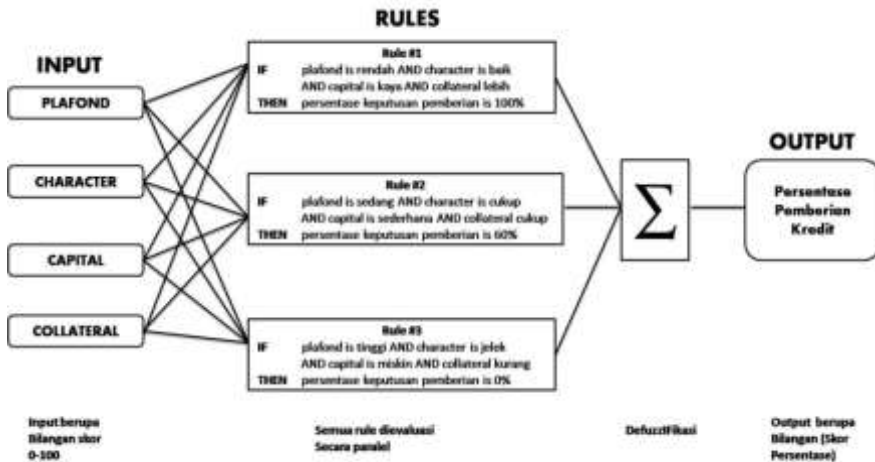
Hasil pengklasifikasian data yang dijadikan sebagai variabel *input* dan perumusan *output* yang diharapkan. Penulis mengambil sampel data dari data-data permohonan yang ada pada Divisi Kredit Bank Mitra, Data tersebut telah di analisa, dan berdasarkan analisa tersebut penulis menemukan bahwa terdapat 4 kelompok data yang dominan sekali mempengaruhi dalam pengambilan keputusan kelayakan pemberian kredit kepada calon debitur. Data tersebut adalah data jumlah *plafond* yang diajukan, penilaian *character*, penilaian kemampuan *capital*, dan penilaian jaminan

yang dimiliki calon debitur. Tabel 1 di bawah ini merupakan hasil pengklasifikasian data ke dalam himpunan *fuzzy*.

Desain Sistem Menentukan Persentase Kelayakan Pemberian Kredit

Pada tahap ini, penulis akan merancang sistem terhadap data yang telah dianalisis supaya pada tahap implementasi ke dalam *software* matlab akan lebih mudah pembahasannya. Pada desain sistem menggunakan *fuzzy* model sugeno ini, terdiri dari empat *step* yaitu desain *input* data calon debitur berdasarkan jumlah *plafond* yang diajukan, nilai *character* calon debitur, penilaian *capital* dan penilaian jaminan yang diajukan kepada bank, serta *output fuzzy* dan proses penetapan *rule*.

Dalam gambar 1 informasi mengalir dari kiri ke kanan. Semua IF-THEN *rule* dieksekusi secara parallel sampai dilakukan proses *defuzzifikasi*. Eksekusi semua IF-THEN *rule* secara paralel merupakan salah satu proses yang penting dalam *fuzzy logic*. Kita bisa membayangkan setiap IF-THEN *rule* akan mewakili sebuah mode operasi tertentu dari sistem yang akan dimodelkan dengan FIS.

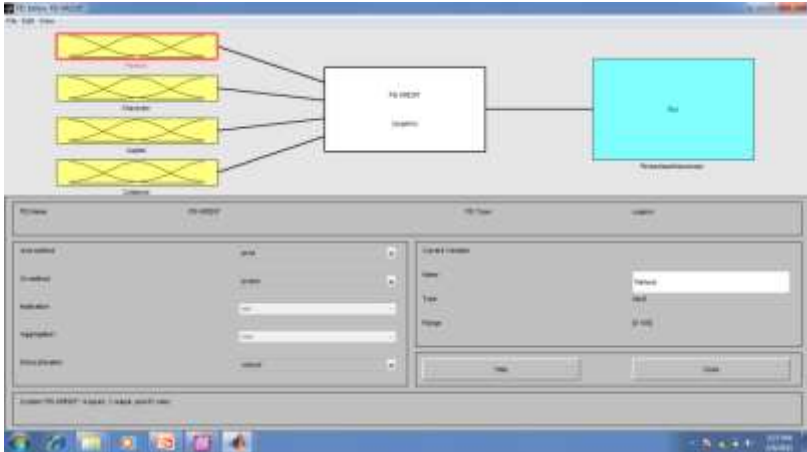


Gambar 13.1. Arsitektur *Fuzzy Inference System*

Proses *fuzzy inference* dalam gambar 2 bisa dibagi dalam lima bagian : *fuzzifikasi* variabel masukan, operasi *fuzzy logic* menggunakan *fuzzy AND* atau *OR* pada bagian *antecedent*, implikasi dari *antecedent* ke *consequent*, agregasi semua *consequent* dari semua IF-THEN rule, dan *defuzzifikasi*. Arsitektur FIS pada gambar 13.2, bila diimplementasikan kedalam *software* matlab, maka akan menghasilkan rangkaian kerja seperti pada gambar 13.2.

Fuzzy Inference System (FIS) mengambil masukan-masukan dan menentukan derajat keanggotaannya dalam semua *fuzzy set*, dengan menggunakan fungsi keanggotaan masing-masing *fuzzy set*. Dalam gambar 13.2, FIS didesain untuk mengambil empat masukan, yaitu *plafond*, *character*, *capital* dan *collateral*, dan menghasilkan sebuah keluaran, yaitu persentase kelayakan pemberian kredit. Nilai linguistik variabel masukan *plafond* dibagi dalam tiga *fuzzy set*: rendah, sedang dan tinggi. Nilai linguistik variabel masukan *character* dibagi dalam tiga *fuzzy set*: baik, cukup dan jelek. Nilai linguistik variabel masukan *capital* juga dibagi dalam

tiga *fuzzy set*: miskin, sederhana dan kaya, begitu juga dengan nilai linguistik variabel masukan *collateral* dibagi dalam tiga *fuzzy set*: kurang, cukup dan lebih. Sebelum evaluasi semua *rule* dilakukan, seluruh masukan harus di *fuzzifikasi* menurut nilai-nilai linguistik di atas.



Gambar 13.2. Model FIS Tipe Sugeno Pada Aplikasi Matlab

Desain *Input Variabel Plafond*

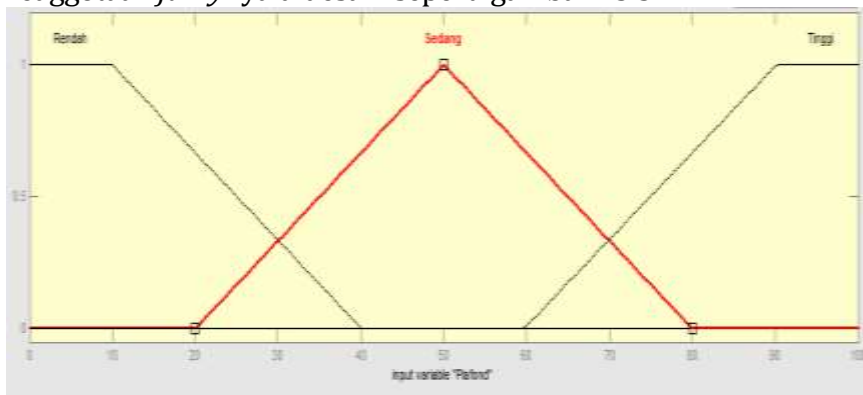
Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya (sering disebut derajat keanggotaan) yang memiliki interval antara 0 sampai 1. Dalam desain *input* data calon debitur untuk variabel *plafond* dibagi berdasarkan jumlah *plafond* yang diajukan calon debitur dengan skor tertentu. Fungsi keanggotaan *plafond* akan didesain sebagai berikut :

$$\mu_{\text{Rendah}}[p] = \begin{cases} 1; & p \leq 10 \\ \frac{(40-p)}{30}; & 10 \leq p \leq 40 \\ 0; & p \geq 40 \end{cases} \quad \dots\dots\dots (13.a)$$

$$\mu_{\text{Sedang}}[p] = \begin{cases} 0; & p \leq 20 \text{ atau } p \geq 80 \\ \frac{(p-20)}{30}; & 20 \leq p \leq 50 \\ \frac{(80-p)}{30}; & 50 \leq p \leq 80 \end{cases} \quad \dots\dots\dots (13.b)$$

$$\mu_{\text{Tinggi}}[p] = \begin{cases} 0; & p \geq 60 \\ \frac{(p-60)}{30}; & 60 \leq p \leq 90 \\ 1; & p \geq 90 \end{cases} \quad \dots\dots\dots (13.c)$$

Dari tiga persamaan di atas (13.a), (13.b), (13.c), bentuk fungsi keanggotaan *fuzzynya* didesain seperti gambar 13.3.



Gambar 13.3. Fungsi Keanggotaan *Input* Untuk Variabel *Plafond*

Gambar 13.3 merupakan *input* pertama dari analisa *fuzzy* tentang kelayakan pemberian kepada calon debitur, yang datanya masih samar sebagai dasar pengambilan keputusan untuk menentukan berapa nilai kelayakan pemberian kredit kepada calon debitur yang mengajukan permohonan pinjamannya kepada bank.

Tabel 13.2. Himpunan *Fuzzy Plafond* Yang Diajukan Untuk *Input 1*

Nama Variabel <i>Fuzzy</i>	Model MF	Parameter	Range
Rendah	trapmf	[0 0 10 40]	0 - 40
Sedang	trimf	[20 50 80]	20 - 80
Tinggi	trapmf	[60 90 100 100]	60 - 100

Setiap variabel mempunyai rentang nilai, yang artinya setiap nilai menyatakan besar pinjaman yang diajukan calon debitur untuk menentukan persentase kelayakan pemberian kredit.

Desain *Output* Persentase Keputusan Pemberian

Tabel 13.3. *Output Fuzzy* Persentase Keputusan Kredit

No	Nama Variabel <i>Output Fuzzy</i>
1	Persentase Keputusan Pemberian 100 %
2	Persentase Keputusan Pemberian 90 %
3	Persentase Keputusan Pemberian 80 %
4	Persentase Keputusan Pemberian 70 %
5	Persentase Keputusan Pemberian 60 %
6	Persentase Keputusan Pemberian 50 %
7	Persentase Keputusan Pemberian 40 %
8	Persentase Keputusan Pemberian 30 %
9	Persentase Keputusan Pemberian 20 %
10	Persentase Keputusan Pemberian 10 %
11	Persentase Keputusan Pemberian 0 %

Hasil (*output*) yang diharapkan dalam sistem pengambilan keputusan kredit berbasis *fuzzy* ini adalah nilai persentase (%) pemberian kredit. Dalam penelitian ini penulis menggunakan *fuzzy*

tipe sugeno, maka output (consequent) sistem tidak berupa himpunan *fuzzy*. Melainkan berupa konstanta atau persamaan linear. Tabel 3 adalah rincian himpunan *fuzzy* untuk *output* dari analisa *fuzzy* tentang kelayakan pemberian kredit kepada calon debitur, yang datanya masih samar sebagai dasar pengambilan keputusan dalam menentukan berapa persentase kelayakan calon debitur untuk menerima kredit dari bank.

Pembentukan Aturan (*rule*) Logika Fuzzy

Berdasarkan jumlah variabel *input* yang telah ditetapkan, maka akan didapatkan 81 kombinasi aturan dari variabel *plafond*, *character*, *capital* dan *collateral*. Untuk itu, aturan *fuzzy* yang didapat juga ada 81 aturan, dengan catatan bahwa setiap aturan yang dibentuk menyertakan semua variabel. Metode Inferensi Fuzzy yang akan digunakan adalah metode Sugeno orde-0. Pada metode ini antecedent (masukan) dipresentasikan dengan proposisi dalam himpunan *fuzzy*, sedangkan konsekuen (keluaran) dipresentasikan dengan sebuah konstanta. Proses pembentukan aturan (*inference*) dilakukan untuk memetakan parameter *input* kepada parameter *output*.

Dari empat *input* variabel *fuzzy* tersebut akan ditetapkan *rule-rule* yang akan menjadi dasar dalam pengambilan keputusan kelayakan calon debitur dalam menerima kredit. Tabel 4.6 merupakan 10 sampel dari 81 *rules* yang telah ditetapkan berdasarkan skor data pada *input* 1 berupa *plafond*, *input* 2 berupa *character*, *input* 3 berupa *capital*, dan *input* 4 berupa *collateral* serta 11 *output* persentase keputusan pemberian.

Tabel 13.4. *Rule Data Untuk Persentase Keputusan Pemberian Kredit*

No	Rule
1	jika <i>plafond</i> rendah dan <i>character</i> baik dan <i>capital</i> kaya dan <i>collateral</i> lebih maka pkp 100 %
2	jika <i>plafond</i> sedang dan <i>character</i> baik dan <i>capital</i> kaya dan <i>collateral</i> kurang maka pkp 90 %
3	jika <i>plafond</i> tinggi dan <i>character</i> baik dan <i>capital</i> sederhana dan <i>collateral</i> kurang maka pkp 70 %
4	jika <i>plafond</i> rendah dan <i>character</i> cukup dan <i>capital</i> kaya dan <i>collateral</i> lebih maka pkp 80 %
5	jika <i>plafond</i> sedang dan <i>character</i> cukup dan <i>capital</i> miskin dan <i>collateral</i> lebih maka pkp 60 %
6	jika <i>plafond</i> sedang dan <i>character</i> baik dan <i>capital</i> kaya dan <i>collateral</i> kurang maka pkp 90 %
7	jika <i>plafond</i> sedang dan <i>character</i> jelek dan <i>capital</i> kaya dan <i>collateral</i> lebih maka pkp 30 %
8	jika <i>plafond</i> rendah dan <i>character</i> jelek dan <i>capital</i> kaya dan <i>collateral</i> lebih maka pkp 40 %
9	jika <i>plafond</i> sedang dan <i>character</i> jelek dan <i>capital</i> sederhana dan <i>collateral</i> kurang maka pkp 10 %
10	jika <i>plafond</i> tinggi dan <i>character</i> jelek dan <i>capital</i> sederhana dan <i>collateral</i> kurang maka pkp 0 %

13.4 Kelebihan dan Kelemahan Fuzzy Logic

Pada setiap kemajuan teknologi juga mempunyai kelebihan yang menjadi karakteristik dari sistem itu. Adapun kelebihan atau keuntungan menggunakan logika *fuzzy* antara lain : 1). Konsep matematis yang mendasari penalarannya sehingga mudah dimengerti ; 2). Memiliki toleransi terhadap data yang tidak tepat; 3) Mudah untuk digabungkan dengan teknik-teknik kendali

konvensional ; 4). Mampu memodelkan suatu sistem secara akurat
5). Pengenalan pola-pola secara mudah dan simpel.

Sistem *fuzzy* mempunyai keunggulan dalam memodelkan aspek kualitatif dari pengetahuan manusia serta proses pengambilan keputusan (*reasoning*) sebagaimana dilakukan oleh manusia dengan menerapkan basis aturan. Sistem *fuzzy* dapat mengatasi kesulitan dalam melukiskan suatu sistem *fisis* yang kompleks dan sukar dimodelkan secara matematis.

Dalam perkembangannya saat ini, Beberapa alasan penggunaan logika *fuzzy* sebagai berikut : 1). Logika *fuzzy* secara konsep lebih mudah untuk dipahami dan dimengerti, 2). Logika *fuzzy* lebih fleksibel dan memiliki toleransi terhadap imprecise data. 3). Logika *fuzzy* mampu memodelkan fungsi-fungsi non linier. 4). Logika *fuzzy* dapat digabungkan dengan teknik konvensional.

Informasi yang digunakan oleh sistem *fuzzy* adalah pengetahuan *kualitatif* tentang sistem dengan sarana *linguistic*. Karena terdiri dari basis aturan dan label linguistik sebagaimana dalam kehidupan manusia, sistem *fuzzy* secara *intuitif* mudah dipahami oleh manusia.

Metode inferensi fuzzy memungkinkan sistem pakar untuk mengatasi ketidakpastian dan kesamaran dalam penalaran. Dengan menggunakan himpunan fuzzy dan tingkat keanggotaan, sistem pakar dapat menangani situasi yang tidak tegas dan memberikan hasil yang lebih manusiawi. Metode inferensi fuzzy memungkinkan integrasi pengetahuan pakar ke dalam sistem pakar. Aturan-aturan fuzzy yang disusun berdasarkan pengetahuan pakar dapat digunakan untuk memberikan rekomendasi dan keputusan yang berkualitas.

Inferensi fuzzy menggunakan bahasa manusia dan himpunan fuzzy yang dapat dipahami dengan mudah oleh pengguna atau pemangku kepentingan. Hasil inferensi dapat

diinterpretasikan dengan jelas dan mengkomunikasikan solusi secara efektif. Metode inferensi fuzzy memiliki toleransi terhadap data noisy atau data yang mengandung gangguan. Dalam metode ini, tingkat keanggotaan fuzzy dapat mengurangi efek dari data noisy, sehingga hasil inferensi menjadi lebih stabil.

Kemampuan Penalaran Fleksibel: Metode inferensi fuzzy memungkinkan penalaran yang lebih fleksibel dengan mempertimbangkan tingkat keanggotaan variabel fuzzy dalam pengambilan keputusan. Hal ini memungkinkan sistem pakar untuk menghasilkan hasil yang lebih adaptif dan responsif terhadap kondisi yang berubah.

Logika *fuzzy* memiliki kelemahan yaitu masih belum terdapat metode standar untuk mentransformasikan pengetahuan dan pengalaman manusia ke dalam basis aturan seta lemah dalam kemampuan belajar terhadap informasi *numeric*. Sehingga, aturan-aturan *fuzzy* yang diperlukan lebih banyak yang dibangun berdasarkan aturan empiris dari pengetahuan operator sistem.

Kekurangan metode inferensi fuzzy dalam sistem pakar:

Subjektivitas dalam Pemilihan Fungsi Keanggotaan: Pemilihan fungsi keanggotaan dalam inferensi fuzzy dapat melibatkan subjektivitas, karena tidak ada metode yang jelas untuk memilih fungsi keanggotaan yang optimal. Hal ini dapat mempengaruhi kualitas inferensi dan membuat hasil yang bervariasi.

Metode inferensi fuzzy mungkin tidak efektif dalam menangani data ekstrem atau anomali. Data yang berada di luar rentang fungsi keanggotaan dapat menghasilkan hasil inferensi yang tidak akurat atau tidak diinginkan.

Komputasi yang Intensif: Sistem pakar dengan metode inferensi fuzzy yang kompleks dapat memerlukan komputasi yang intensif. Hal ini dapat mempengaruhi waktu respons sistem pakar dan memerlukan sumber daya komputasi yang lebih besar.

Kombinasi Aturan yang Rumit: Dalam sistem pakar yang kompleks, kombinasi aturan fuzzy dapat menjadi rumit. Menentukan aturan fuzzy yang tepat dan mengatur urutan eksekusi aturan dapat menjadi tugas yang sulit dan memakan waktu.

Meskipun metode inferensi fuzzy memiliki kekurangan, kelebihan-kelebihan yang dimilikinya membuatnya menjadi alat yang berguna dalam sistem pakar dan berbagai aplikasi lainnya di mana ketidakpastian dan kesamaran perlu ditangani dalam penalaran.

DAFTAR PUSTAKA

- Arisandi, D. and Sari, I. P. 2021. *Sistem Pakar Dengan Fuzzy Expert System*. Jakarta: Gracias Logis Kreatif. Available at: <https://books.google.co.id/books?id=dQkiEAAAQBAJ>.
- Galang Persada Nurani Hakim, S. T. M. T. *et al.* 2021. *SISTEM FUZZY: Panduan Lengkap Aplikatif*. Jakarta: Penerbit Andi. Available at: <https://books.google.co.id/books?id=GqU4EAAAQBAJ>.
- Kandel, A. 1991. *Fuzzy Expert Systems*. Taylor & Francis.
- Kusumadewi, S. and Purnomo, H. 2004. *Aplikasi logika fuzzy untuk pendukung keputusan*/Sri Kusumadewi. Yogyakarta: Graha Ilmu.
- Sulaiman, R. *et al.* (no date) *Himpunan Fuzzy Teori dan Aplikasi*. Jakarta: Zifatama Jawa.
- Thukral, S. and Bal, J. S. 2019. 'Medical Applications on Fuzzy Logic Inference System: A Review', *International Journal of Advanced Networking and Applications*, 10(4), pp. 3944–3950. doi: 10.35444/ijana.2019.10046.

BIODATA PENULIS



Ir. Hasanuddin Sirait, M.T

Dosen Program Studi Teknik Informatika
AMIK Parbina Nusantara

Penulis lahir di Rawang Baru, Asahan – Sumatera Utara tanggal 24 Agustus 1968. Penulis saat ini menjadi dosen pada Program Studi Teknik Informatika pada AMIK Parbina Nusantara. Penulis menyelesaikan pendidikan S1 pada STMIK YPTK Padang dan melanjutkan S2 pada Padaa Universitas Hasanuddin Ujung Pandang dengan fakultas Teknik Elektro dengan Konstentrasi Teknik Informatika dan Telekomunikasi. Penulis sejak menamatkan studi S1 telah banyak membangun aplikasi sistem dan proyek teknologi informatika pada perusahaan swasta dan pemerintah daerah dan di beberapa perguruan tinggi. Penulis telah menerbitkan beberapa buku teks bidang informatika dan komputerisasi serta telah menerbitkan banyak artikel ilmiah hasil penelitian terkait Perkembangan Teknologi Informatika dan Komputerisasi Sistem pada jurnal terakreditasi nasional dan internasional sebagai perwujudan Tridharma Perguruan Tinggi. Penulis juga aktif pada organisasi profesi dan assosiasi perguruan tinggi dibidang komputer sejak tahun 2005, penulis juga

merupakan *editor* dan *reviewer* di beberapa jurnal penelitian nasional yang terakreditasi.

BIODATA PENULIS



Rini Mayasari, M. Kom.

Dosen Program Studi Informatika

Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

Penulis lahir di Karawang, Kabupaten Karawang, Jawa Barat 28 Mei 1991. Menempuh jenjang pendidikan S1 pada program studi Teknik Informatika di Universitas Singaperbangsa Karawang, lulus tahun 2012. Kemudian melanjutkan pendidikan S2 jurusan Ilmu Komputer, lulus tahun 2016 dari Universitas Budi Luhur, Jakarta. Selain menempuh pendidikan formal penulis juga memiliki beberapa sertifikasi profesi, baik lokal maupun internasional. Saat ini penulis aktif sebagai pengajar di program studi Informatika Universitas Singaperbangsa Karawang sejak tahun 2016 sampai sekarang.

BIODATA PENULIS



Nono Heryana, M. Kom.

Dosen Program Studi Sistem Informasi
Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang

Penulis lahir di Lampung Barat tanggal 03 Maret 1989. Menyelesaikan pendidikan S1 pada Jurusan Teknik Informatika di Universitas Singaperbangsa Karawang lulus tahun 2012 dan melanjutkan S2 pada Jurusan Ilmu Komputer di Universitas Budi Luhur lulus tahun 2016. Selain menempuh pendidikan formal penulis juga memiliki beberapa sertifikasi profesi, baik lokal maupun internasional. Seperti: *Google Certified Educator Level 1*, *Certified International of Internal Quality Audit (CIIQA)*, *Scrum Master Accredited Certification™ (SMAC)*, *Scrum Foundation Professional Certificate (SFPC)*, *Certified of Editor (C. Ed)*, *Certified Business Operations Associate (CBOA®)* dan *Remote Worker Professional Certificate (RWPC)*. Penulis adalah dosen tetap pada Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang, sejak tahun 2016 sampai dengan sekarang.

BIODATA PENULIS



Ir. Johni S. Pasaribu, MT.

Dosen Program Studi Sistem Informasi
Fakultas IT Politeknik Piki 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



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



Ronald Belferik, S.Kom., M.Kom

Dosen Program Studi Teknologi Rekayasa Perangkat Lunak
Politeknik Wilmar Bisnis Indonesia

Penulis lahir di Deli Serdang tanggal 05 April 1987. Penulis adalah dosen tetap pada Program Studi Teknologi Rekayasa Perangkat Lunak, Politeknik Wilmar Bisnis Indonesia. Menyelesaikan pendidikan S1 pada Jurusan Sistem Informasi Universitas Mikroskil dan melanjutkan S2 pada Jurusan Teknik Informatika Universitas Sumatera Utara. Penulis menekuni bidang Menulis.

BIODATA PENULIS



Aisyah Mutia Dawis, S.Kom., M.Kom.

Dosen Program Studi Sarjana Sistem dan Teknologi Informasi
Fakultas Sains dan Teknologi Universitas 'Aisiyiah Surakarta

Penulis lahir di Kota Surakarta pada tanggal 3 Juni 1991. Ia Lulus S1 pada tahun 2013 hingga mendapat gelar Sarjana Komputer di Universitas Muhammadiyah Surakarta serta mendapatkan penghargaan sebagai lulusan terbaik Se Fakultas Komunikasi dan Informatika, selanjutnya ia lulus S2 pada tahun 2019 hingga mendapat gelar Magister Komputer di Universitas Amikom Yogyakarta. Saat ini ia tercatat sebagai dosen tetap di Program Studi Sarjana Sistem dan Teknologi Informasi di Universitas 'Aisiyiah Surakarta. Selain mengajar ia aktif dalam kegiatan tridharma lainnya diantaranya ialah penelitian dan pengabdian kepada masyarakat. Kegiatan penelitian internal dan eksternal pernah dilakukannya. Beberapa penelitian berjudul : *"Evaluation of The Website 'Aisiyiah Surakarta of University Performance Based on Search Engine Optimization Using Automated Software Testing Gtmatrix"* dan *"Utilization of Virtual Reality Technology in Knowing the Symptoms of Acrophobia and Nyctophobia"*. Ia sering menjadi *invited speaker* diantaranya menjadi tenaga pengajar BPPTIK Kementerian Kominfo. Serta

sebagai bentuk pengabdian kepada masyarakat, ia pun pernah terlibat aktif sebagai trainer bimbingan teknis bantuan TIK SMP, SMA, Mahasiswa, Tenaga Pendidik, Guru dan Dosen dari tahun 2010 hingga saat ini.

BIODATA PENULIS



Andryanto. A

Penulis lahir di Kota Palopo (Sulawesi Selatan). Pria yang kerap disapa Andry ini adalah anak dari pasangan Aman (ayah) dan Rosmiati (ibu).

BIODATA PENULIS



Drs. Edison Ulung Panggabean Simanihuruk, M.Kom

Dosen Program Studi Teknik Informatika
AMIK PARBINA NUSANTARA

Penulis lahir di Huta Sidauruk Dua, Kabupaten Samosir tanggal 21 Maret 1971. Penulis menyelesaikan pendidikan Sekolah Menengah Atas di SMA Negeri Paguruan Kabupaten Samosir, kemudian melanjutkan pendidikan S1-MIPA jurusan Sains Komputer di Universitas Sisingamangaraja II Medan. Setelah menyelesaikan pendidikan S1 banyak berkecimpung di bidang Hardware dan Software. Kemudian melanjutkan pendidikan Magister Komputer di YPTK UPI Padang dengan konsentrasi Sistem Informasi. Penulis juga aktif sebagai narasumber dalam berbagai bidang informatika di Tingkat Pendidikan Menengah, pemerintah maupun swasta.

BIODATA PENULIS



Reagan Surbakti Saragih, M.Kom

Dosen Program Study Ilmu Komputer
Universitas HKBP Nommensen Pematangsiantar

Penulis lahir di kota Pematangsiantar pada tanggal 09 september 1981. Penulis menyelesaikan pendidikan Sekolah Menengah Atas di SMK Sw GKPS Pematangsiantar, kemudian melanjutkan Pendidikan S1 Jurusan Teknik Informatika Komputer di Universitas Sisingamangraja XII Medan. Setelah menyelesaikan pendidikan S1 banyak berkecimpung di bidang Networking dan Software. Kemudian melanjutkan pendidikan Magister Komputer di YPTK UPI Padang dengan konsentrasi Sistem Informasi. Penulis juga aktif sebagai narasumber dan konsultan di Tingkat Pendidikan Menengah, Pemerintah maupun swasta.

BIODATA PENULIS



Herlina Latipa Sari, S.Kom., M.Kom.

Dosen Prodi Informatika FILKOM Universitas Dehasen Bengkulu

Penulis lahir di Bengkulu tanggal 6 Juli 1979. Penulis merupakan dosen tetap pada Prodi Informatika FILKOM Universitas Dehasen Bengkulu yang mulai berkarir dari tahun 2006 dan menekuni bidang ilmu Sistem Cerdas. Penulis menyelesaikan pendidikan S1 pada Prodi Teknik Informatika Universitas Bina Nusantara Jakarta dan melanjutkan S2 pada Jurusan Teknologi Informasi Universitas UPI YPTK Padang. Selain aktif mengajar penulis juga menerbitkan publikasi baik pada jurnal lokal, nasional dan internasional, serta aktif juga dalam melaksanakan pengabdian masyarakat sesuai bidang keahlian.

BIODATA PENULIS



Dr. Raimon Efendi, S.A.B,M.Kom

Dosen Magister Program Studi Teknologi Pendidikan
FKIP Universitas Dharma Indonesia

Dr. Raimon Efendi. S.A.B., M.Kom., lahir tanggal 10 September 1983, sejak tahun 2012 menjadi dosen di Universitas Dharma Indonesia. Pendidikan S1 Jurusan Ilmu Administrasi Bisnis, Fakultas Ilmu Sosial dan Ilmu Politik Universitas Terbuka, S2 Jurusan Sistem Informasi, Fakultas Ilmu Komputer di Universitas Putra Indonesia 'YPTK' Padang dan S3 Jurusan program studi Pendidikan Teknologi dan Kejuruan Fakultas Teknik Universitas Negeri Padang. Memiliki kepakaran dalam bidang Entrepreneurship, Teknologi Informatika, dan Teknologi Pendidikan Kejuruan. Telah melakukan penemuan-penemuan bidang Entrepreneurship, Pendidikan Teknologi dan Kejuruan terutama pada model-model pembelajaran berbasis IT dan pelatihan Kewirausahaan. Saat ini penulis aktif mengelola Usaha di Bidang Teknologi Informatika dan aktif meneliti dan menulis sebagai wujud sumbangsih keilmuan bagi perkembangan pendidikan Bangsa Indonesia di era digital. Moto: Upgrade Your Skill, Upgrade your Life.