

DATA RAYA DAN PEMOGRAMAN

**Roro Inda Melani
Rizki Dewantara
Reska Mayefis
Sri Rezeki Candra Nursari
Salaki Reynaldo Joshua
Dahlan Susilo
Mutia Delina
Elvi Rahmi**



GET PRESS INDONESIA

DATA RAYA DAN PEMOGRAMAN

Penulis : Roro Inda Melani

Rizki Dewantara

Reska Mayefis

Sri Rezeki Candra Nursari

Salaki Reynaldo Joshua

Dahlan Susilo

Mutia Delina

Elvi Rahmi

ISBN :

Editor : Hendra Nusa Putra, S.Kom., M.Kom.

Penyunting: Yuliatr Novita, M.Hum

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

Penerbit: GET PRESS INDONESIA

Anggota IKAPI No. 033/SBA/2022

Redaksi:

Jl. Palarik RT 01 RW 06, Kelurahan Air Pacah

Kecamatan Koto Tengah, Padang, Sumatera Barat

website: www.getpress.co.id

email: adm.getpress@gmail.com

Cetakan pertama, Agustus 2024

Hak cipta dilindungi undang-undang

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

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, buku “Data Raya dan Pemrograman” ini dapat tersusun dengan baik. Buku ini disusun untuk memberikan mengenai perkembangan teknologi dan informasi, pentingnya literasi digital di era informasi, etika dalam dunia digital, serta konsep dasar dalam pemrograman dan pengelolaan data. Melalui bab-bab yang terstruktur, pembaca akan diajak memahami struktur data, jenis-jenis data, teori pemrograman, metode analisis data, hingga teknik memvisualisasikan data secara efektif.

Kami berharap buku ini dapat menjadi sumber belajar yang bermanfaat bagi mahasiswa, praktisi, dan siapa saja yang tertarik dalam bidang teknologi informasi dan pemrograman. Dengan pengetahuan yang mendalam tentang data dan pemrograman, diharapkan pembaca dapat mengaplikasikan ilmu ini dalam berbagai bidang untuk mendukung pengambilan keputusan yang lebih baik serta berkontribusi dalam pengembangan teknologi di masa depan. Semoga buku ini dapat memberikan inspirasi dan wawasan yang luas bagi semua pembacanya.

Padang, Juli 2024

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
DAFTAR GAMBAR.....	v
DAFTAR TABEL	vii
BAB 1 PERKEMBANGAN TEKNOLOGI DAN INFORMASI.....	1
1.1 Selayang Pandang Data Raya.....	1
1.2 Data vs Informasi.....	6
1.3 Sumber Koleksi Data.....	8
1.4 Evolusi Pengelolaan Data.....	9
1.5 Perlindungan bagi pemilik data	13
DAFTAR PUSTAKA	16
BAB 2 URGENSI LITERASI DIGITAL DI ERA INFORMASI DIGITAL	19
2.1 Pendahuluan.....	19
2.2 Macam Literasi Digital di Era Informasi Digital.....	22
2.3 Contoh Urgensi Literasi Digital di Era Informasi Digital.....	26
2.4 Keberlakuan Literasi Digital di Era Informasi Digital	30
2.5 Dampak Negatif Literasi Digital	35
2.6 Kesimpulan	36
DAFTAR PUSTAKA	37
BAB 3 ETIKA DALAM DUNIA DIGITAL	39
3.1 Definisi Etika Digital.....	39
3.2 Pentingnya Etika Digital	42
3.3 Tantangan Etika Digital.....	44
3.4 Strategi untuk Mempromosikan Perilaku Etis Secara Online	46
3.5 Peran Pendidikan dalam Menumbuhkan Etika Digital.....	47
3.6 Kesimpulan	48

DAFTAR PUSTAKA.....	49
BAB 4 STRUKTUR DATA.....	51
4.1 Pendahuluan.....	51
4.2 Tipe Struktur Data	51
4.3 Fungsi Utama Struktur Data	61
4.4 Jenis Struktur Data.....	62
DAFTAR PUSTAKA.....	64
BAB 5 JENIS DATA	65
5.1 Pendahuluan.....	65
5.2 Tipe Data Primitif.....	67
5.3 Tipe Data Komposit.....	71
5.4 Tipe Data Komposit.....	76
5.5 Tipe Data yang Didefinisikan oleh Pengguna	81
5.6 Konversi dan Casting Tipe Data	86
5.7 Tipe Data dalam Berbagai Bahasa Pemrograman	90
5.8 Memilih Tipe Data yang Tepat.....	95
DAFTAR PUSTAKA.....	100
BAB 6 TEORI TENTANG PEMROGRAMAN	103
6.1 Pendahuluan.....	103
6.2 Pengertian Pemrograman.....	104
6.3 Sejarah Pemrograman.....	106
6.4 Tahapan Pemrograman	112
6.5 Prinsip Pemrograman	116
6.6 Fungsi Bahasa Pemrograman.....	119
6.7 Jenis Bahasa Pemrograman	121
DAFTAR PUSTAKA.....	124
BAB 7 METODE ANALIS DATA.....	127
7.1 Pendahuluan.....	127
7.2 Metode Kualitatif.....	131

7.3 Metode Kuantitatif.....	132
7.4 Prosedur Analisis Data.....	136
7.5 Pemanfaatan Tools untuk Analisis Data	137
DAFTAR PUSTAKA	140
BAB 8 MEMVISUALISASIKAN DATA.....	141
8.1 Pendahuluan.....	141
8.2 Jenis Chart/Grafik.....	142
8.3 Pemilihan Jenis Visualisasi	147
8.4 Alat dan Perangkat Lunak untuk Visualisasi Data	151
DAFTAR PUSTAKA	154

DAFTAR GAMBAR

Gambar 2. 1. Ukuran Keterampilan Literasi Digital	26
Gambar 2. 2. Foto Timbal Balik antara Pemateri dengan mahasiswa	28
Gambar 5. 1. Contoh dalam C++	68
Gambar 5. 2. Diagram	69
Gambar 5. 3. Integer Overflow	70
Gambar 5. 4. Visualisasi Tipe Komposit	72
Gambar 5. 5. Pencarian Biner	73
Gambar 5. 6. Pencocokan Pola dalam String	74
Gambar 5. 7. Contoh Permainan Sederhana	74
Gambar 5. 8. Visualisasi ADT	77
Gambar 5. 9. Web browser's back button	78
Gambar 5. 10. Algoritma Dijkstra	79
Gambar 5. 11. Tipe Yang Ditentukan Pengguna	82
Gambar 5. 12. Vending Machine	83
Gambar 5. 13. Sistem Perpustakaan	84
Gambar 5. 14. Inheritance, Polymorphism, And Generics	85
Gambar 5. 15. Visualisasi Konversi	87
Gambar 5. 16. Mixed-Type Expression	88
Gambar 5. 17. Implicit Conversions Can Lead to Subtle Bugs	88
Gambar 5. 18. Explicit casts	88
Gambar 5. 19. Circle	90
Gambar 5. 20. Dynamic Cast	90
Gambar 5. 21. High-Level Comparison Of Data Typing Across Language Paradigms	91
Gambar 5. 22. Statically-typed languages (C++, Java, Rust)	92
Gambar 5. 23. Dynamically-typed languages (Python, JavaScript, Ruby)	92
Gambar 5. 24. Key Difference Is How Languages Handle Type Conversion	92
Gambar 5. 25. Object-oriented languages introduce inheritance	93
Gambar 5. 26. Functional languages often have advanced type features	93
Gambar 5. 27. Factors That Influence Type Choice	96
Gambar 5. 28. Warns against using floating-point types	97

Gambar 5. 29. The choice often depends on the character set needed.....	98
Gambar 5. 30. The choice of container type	98
Gambar 5. 31. Bit-level data types.....	98
Gambar 6. 1. Siklus Pengolahan Data	105
Gambar 6. 2. Evolusi Bahasa Pemrograman.....	107
Gambar 6. 3. Tahapan Pemrograman.....	113
Gambar 6. 4. Prinsip-prinsip Pemrograman	117
Gambar 7. 1. Contoh penyajian data dalam diagram Pie.....	128
Gambar 7. 2. Contoh penyajian data dengan diagram batang.....	129
Gambar 7. 3. Contoh penyajian data dengan diagram titik atau scater plot	130
Gambar 7. 4. Salah satu contoh statistika inferensia	136
Gambar 7. 5. Membuat diagram dengan bantuan excel.....	137
Gambar 7. 6. Microsoft Excel menyediakan beberapa tipe diagram dan grafik.	138
Gambar 7. 7. <i>Feature</i> pada Microsoft Excel untuk mengolah data.	139

DAFTAR TABEL

Tabel 5. 1. Different data types are suited for different kinds of data and operations	66
Tabel 5. 2. The Characteristics and Use Cases of Primitive Types	70
Tabel 5. 3. The Characteristics and Use Cases of Primitive Types	75
Tabel 5. 4. The Characteristics and Use Cases of These ADTs	79
Tabel 5. 5. The Characteristics and Use Cases of These User-Defined Types.....	84
Tabel 5. 6. Conversion Rules And Dangers	89
Tabel 5. 7. Key Differences.....	94
Tabel 5. 8. Key Differences.....	97
Tabel 7. 1. Penyajian data dalam tabel	128
Tabel 7. 2. Jumlah kasus kekerasan terhadap anak	130
Tabel 7. 3. Hasil pengukuran tinggi badan siswa	133
Tabel 7. 4. Data tinggi badan siswa yang telah diurutkan	134
Tabel 7. 5. Jumlah kemunculan (frekwensi) tinggi badan siswa	134

BAB 1

PERKEMBANGAN TEKNOLOGI DAN INFORMASI

Oleh Roro Inda Melani

1.1 Selayang Pandang Data Raya

Kajian tentang Data Raya atau yang biasa disebut sebagai Big Data tidak bisa lepas dari konsep dasar tentang data dan perubahan-perubahan yang terjadi dalam performa pengaturan data (*data management*). Data Raya sendiri adalah sebuah frasa istilah yang digunakan untuk menggambarkan suatu kumpulan data terstruktur maupun data tidak terstruktur yang volumenya luar biasa besar sehingga akan sulit diproses bila harus dipaksakan teknologi software dan database yang tradisional. (Beal, 2024).

Pada banyak perusahaan, terdapat kondisi di mana volume data yang sangat besar, dengan variasi jenis data yang beragam dan pergerakannya sangat cepat. Setiap transaksi di situs web, setiap informasi yang dikumpulkan oleh perwakilan penjualan, dan setiap dokumen yang melewati email perusahaan berkontribusi pada bentuk data perusahaan, dan semua informasi itu bertambah dengan sangat cepat. Ketika ditangkap, diformat, disimpan, dan dianalisis, data ini memberikan informasi dan wawasan bagi perusahaan untuk meningkatkan pendapatan, bagaimana cara menarik pelanggan baru atau mempertahankan pelanggan lama, dan meningkatkan operasi.

Data raya bukanlah suatu teknologi yang berdiri sendiri. Menggunakan data raya artinya berhadapan dengan kombinasi teknologi terbaru dengan teknologi sebelumnya dalam rangka mendapatkan wawasan aksi yang lebih menjanjikan.

1.1.1 Sejarah Singkat Data Raya

Dalam bahasa Indonesia (KBBI) *Big Data* diterjemahkan sebagai Maha Data. Dalam buku ini istilah yang digunakan adalah Data Raya. Istilah “Data Raya” pertama kali digunakan pada tahun 1990an. Seiring berjalannya waktu istilah ini semakin sering digunakan pada tahun 2000an. Pada tahun 90an, manajemen data melibatkan *database* dan analisis data terstruktur. Pada tahun 2000-an, dengan semakin banyaknya penggunaan Internet, data yang dikumpulkan dari lalu lintas web dan e-commerce memiliki variasi yang lebih banyak. Pada tahun 2010-an, perangkat seluler dan sensor, serta sumber data yang ada, menghasilkan begitu banyak informasi sehingga pembuatan data meningkat secara eksponensial selama kurun waktu beberapa dekade terakhir.

Data raya adalah evolusi yang panjang dari pengumpulan data dan penggunaan data dan sungguh bukan suatu fenomena yang baru.

1.1.2 Mengapa Data Raya Menjadi Penting ?

Jawaban singkat dari pertanyaan tersebut adalah karena data sama artinya dengan potensi keuntungan dalam suatu proses bisnis. Data adalah aset yang amat penting bagi suatu perusahaan. Pengelolaan data yang buruk bisa berakibat fatal bagi kelangsungan suatu bisnis.

Data, jika digunakan dengan tepat, akan menghasilkan keuntungan bagi perusahaan. Data tentang pelanggan, produk, pemasaran, kantor, toko, dan karyawan mengungkapkan rincian tentang setiap aspek operasi bisnis perusahaan.

Data raya dengan karakteristiknya menjadikannya sulit untuk dikelola. Selain karena jumlahnya sangat banyak, memiliki variasi yang banyak dan juga arus penambahannya sangat cepat. Hal ini menjadikan tantangan tersendiri untuk menghasilkan analisis. Proses manajemen data harus memilih antara data penting dari data yang tidak penting, kemudian membedakan antara wawasan yang berguna dan data yang tidak berguna dan berbahaya. Data yang akurat, relevan, dan terorganisir menghasilkan pengambilan keputusan organisasi yang lebih baik. Diperlukan *tools* atau alat pengelolaan dan sistem yang sesuai untuk mengatur dan memahaminya dalam rangka memanfaatkan data raya tersebut.

Data raya memiliki arti yang penting bukan hanya bagi pihak yang mengumpulkan, namun juga memiliki arti penting bagi sumber dari mana data tersebut ditarik. Sudah seharusnya mekanisme pembentukan kumpulan data yang masif ini juga memberikan ruang perlindungan bagi individu baik sebagai pelanggan maupun sebagai warga negara. Di akhir bab ini akan dibahas lebih rinci tentang bagaimana perlindungan hak individu pemilik data ini diatur.

1.1.3 Apa yang Bisa Dilakukan dengan Data Raya ?

Untuk dapat memahami keajaiban yang bisa dilakukan dengan menggunakan data raya maka penjelasan ini akan tampil dalam bentuk cerita fiksi namun realistis. Sebagai contoh, seorang pria dengan inisial A membeli sejumlah buku pada toko perusahaan penjualan buku berinisial X. Perusahaan X berusaha mendapatkan data berupa nama, usia, tempat kerja, nomor telepon dan email dari pria A dan tentu saja judul buku-buku yang dibeli pria A di toko perusahaan X. Data-data tersebut bisa diperoleh perusahaan X dengan menawarkan potongan harga dengan minimal pembelian. Kata-kata promosi tersebut bisa tertulis seperti ini, "Dapatkan diskon 15% setiap pembelian Rp.

150.000 dengan menjadi member toko X. Isikan biodata Anda di sini”.

Perusahaan X telah menerapkan manajemen big data mengelola data perusahaannya. Maka dengan menggunakan data-data dari pelanggan pria A ini maka toko X akan dapat memberikan rekomendasi judul buku yang akan terbit berdasarkan judul-judul buku yang telah pernah dibeli pria A. Penawaran ini bisa disampaikan melalui pesan *broadcast* di email atau *whatsapp* secara berkala. Perusahaan X juga dapat memantau frekuensi pembelian buku yang dilakukan pria A di perusahaan X melalui toko-toko mereka sekalipun lokasinya tersebar di berbagai tempat.

Pendekatan aksi apa lagi yang bisa dilakukan perusahaan X menggunakan data raya? Bila perusahaan X mampu mengumpulkan seluruh data penjualan buku dari seluruh toko mereka maka minimal akan terkumpul judul-judul buku dengan jumlah pembelian terbanyak. Dengan demikian maka perusahaan X akan bisa merencanakan pembelian buku selanjutnya dengan tema sejenis. Misal telah terdata tema: politik, pertanian mikro, motivasi bisnis, sejarah PD 2, dan kisah sukses bisnis youtuber adalah top 5 tema buku dengan terbanyak dalam 5 bulan terakhir. Maka data 5 bulan tersebut dapat digunakan rencana pembelian stok buku dengan tema serupa di bulan ke-6. Bukankah hal itu sudah lazim dilakukan perusahaan-perusahaan sejak jaman dahulu? Bukankah untuk melakukan perencanaan pembelian sedemikian rupa itu berarti perusahaan X tidak perlu menggunakan data raya, begitu pula perusahaan lainnya?

Adalah benar bahwa prediksi pembelian stok adalah hal yang lazim harus dilakukan oleh perusahaan X dan perusahaan sejenis untuk menjaga kelangsungan bisnis perusahaan mereka. Namun dengan karakteristik dasar menjadikan ciri dasar dari data raya maka hasil yang didapatkan dari analisa data menjadi

berbeda. Karakter yang sering ditulis dengan kode V tersebut adalah : *volume*, *velocity* dan *variety*. Dengan terdapat 3 karakteristik maka ditulis sebagai 3V.

Mari kita lanjutkan kisah fiksi realistik dari perusahaan X. Dalam rangka mengikuti perkembangan zaman, perusahaan X memutuskan untuk melakukan *engagement* dengan pelanggan melalui website dan beberapa akun media sosial. Selain toko yang memiliki lokasi fisik (*offline store*), perusahaan X juga menyediakan *online store* baik berupa *landing page* berbasis *website* maupun membuka akun toko di *marketplace*. Akun sosial yang digunakan perusahaan X antara lain adalah akun X, akun Instagram dan akun Youtube. Di channel youtubenya mereka memberikan ulasan dari produk yang disediakan dan juga untuk berinteraksi dengan *user* mereka.

Dengan dibukanya berbagai channel interaksi dengan pengguna dan pelanggannya, perusahaan X memiliki potensi untuk menerima berbagai macam variasi input dari pengguna channel mereka tersebut. Perusahaan X bisa menerima *tweet* melalui akun X mereka, menerima komentar, *like* dan *share* serta kumpulan insight melalui akun instagram. Begitu juga dengan akun Youtube dan akun toko di *marketplace*. Sehingga selain data penjualan yang sifatnya statis, datang pula data yang lebih dinamis dan tidak terstruktur dalam bentuk komentar dan testimoni dari pengguna, pembeli dan komentator. Variasi yang lebih beragam (*variety*) tersebut dibarengi pula dengan kecepatan (*velocity*) trafik akses data yang meningkat dan mengakibatkan lonjakan jumlah (*volume*) data. Akan amat disayangkan bila perusahaan X mempertahankan pengolahan data secara tradisional sehingga akan kesulitan untuk dapat menyimpan, mengelola, mengolahnya menjadi analisa dari data-data tidak terstruktur tersebut. Bila perusahaan X mau berkembang lebih pesat maka pengelolaan data raya menjadi alternatif yang menjanjikan sekaligus menantang.

Contoh hasil pengelolaan data-data tersebut antara lain adalah :

- Perusahaan X mampu memiliki gambaran user persona yang lebih tertarget sehingga dapat menentukan akan melakukan pendekatan dengan cara yang lebih efektif.
- Perusahaan X memiliki informasi trend penjualan buku beberapa tahun ke belakang dan prediksi 1 tahun ke depan, sehingga bisa menentukan akan melakukan kerjasama dengan pihak yang lebih menguntungkan.
- Perusahaan memiliki *funnel non targeted user* sehingga dapat menentukan akan melakukan manuver promosi sesuai user persona tersebut.
- Perusahaan X dapat menentukan pendekatan yang lebih sesuai dengan memperbaiki user journal untuk meningkatkan kenyamanan *user* dalam melakukan pembelian secara *offline* dan *online*.

Sebelum era kematangan data raya, perusahaan seringkali menyimpan sejumlah besar data-data tersebut tanpa tahu harus dimanfaatkan dengan cara apa.

1.2 Data vs Informasi

Data didefinisikan sebagai suatu nilai (tunggal) atau sekumpulan nilai (jamak) yang mewakili suatu konsep atau konsep tertentu. Data dapat mencakup, namun tidak terbatas pada, 1) data geospasial 2) data tidak terstruktur, 3) data terstruktur. Data dapat berubah menjadi informasi ketika dianalisis dan mungkin digabungkan dengan data lain untuk mengekstrak makna, dan memberikan konteks. Sedangkan Informasi bisa berarti segala komunikasi atau representasi pengetahuan seperti fakta, data , atau opini dalam media atau

bentuk apa pun, termasuk bentuk tekstual, numerik, grafik, kartografi, naratif, atau audiovisual. (resources.data.gov, 2024).

1.2.1 Data Terstruktur

Data terstruktur memiliki kaitan yang kuat dengan istilah tradisional database. Terdapat kondisi di mana data disimpan dalam kolom dan baris dengan struktur yang terdefiniskan dengan organisasi dan model tertentu. Pendefinisian karakter organisasi, model hingga ke penentuan tipe data memungkinkan data disimpan dan diolah dengan lebih sistematis dan mudah dipahami.

Keterbatasan yang dimiliki dari pengelolaan data terstruktur ini adalah biaya yang cenderung tinggi dengan kapasitas penyimpanan data yang terbatas. Teknik pemrosesan data menjadikan RDBMS (*Relational Data Base Management System*) sebagai satu-satunya jalur untuk menyimpan dan mengolah data secara efektif (Patil, 2018).

1.2.2 Data Tidak Terstruktur

Data tidak terstruktur mendominasi bisnis di jaman modern dan telah terdapat cara yang lebih bisa dilakukan dalam mengeksploitasi data tersebut (Eberendu, 2018). Menurut Feldman & Sanger, 2007, data tidak terstruktur biasanya mencakup file dengan format bitmap atau objek, teks, email dan tipe data selain yang biasa ada di database.

Isi percakapan dalam email dan komentar interaksi antar user di sosial media termasuk ke dalam data tidak terstruktur. Selanjutnya Patil, 2018 menjelaskan data terstruktur adalah data yang tidak memiliki struktur tertentu sehingga sulit untuk disimpan dalam format baris dan kolom.

1.2.3 Data Semi Terstruktur

Data semi terstruktur dicirikan sebagai data tanpa struktur tertentu, mungkin tidak beraturan, tidak lengkap dan memiliki struktur yang berubah dengan cepat atau tidak dapat diprediksi, tidak sesuai dengan skema eksplisit. Data dalam golongan ini bertentangan dengan konsep data terstruktur dan tidak bisa disimpan dalam format baris dan kolom. Artinya, ia tidak berorientasi tabel seperti pada model basis data relasional atau grafik yang diurutkan seperti pada basis data objek.

Menurut Hanig, Schierle dan Trabold (2010) model data semi terstruktur memungkinkan informasi dari beberapa sumber, dengan properti terkait tetapi berbeda, untuk digabungkan menjadi satu kesatuan, misalnya email, XML, file Doc.

Volume data jenis ini berkembang sangat pesat sehingga sangat sulit untuk dikelola dan dianalisa secara sepenuhnya tanpa menggunakan alat yang sesuai. Untuk dapat menganalisis data tidak terstruktur dan data semi terstruktur diperlukan pengetahuan teknologi canggih.

1.3 Sumber Koleksi Data

Data raya dengan karakteristik *variety* artinya data dikoleksi dengan ragam yang bervariasi dari sumber yang beragam. Sumber data bisa data dari sensor, data yang dikoleksi dari mesin, data dari input yang dilakukan manusia dan data dari proses bisnis.

Meledaknya jumlah data yang dikoleksi dari sensor menjadi resiko dari berkembangnya teknologi di bidang robotik dan otomatisasi peralatan kehidupan manusia termasuk alat rumah tangga dan sensor lalu lintas kendaraan di jalan raya. Sebagian dari sensor tersebut memang ditujukan untuk

memberi informasi dengan kepentingan yang tinggi seperti pada sensor cuaca dan yang menyangkut kebencanaan.

Dapat dibayangkan bagaimana data-data yang terkumpul saat GPS (*Global Positioning System*) di perangkat *mobile* milik pengguna dinyalakan telah menyumbang volume, variasi, dan visualisasi yang membutuhkan akses data cepat secara *real time*. Informasi semacam peta lokasi, titik koordinat, gambar, alur dan lintasan, kondisi kemacetan dan informasi penyerta lainnya menjadi sistem yang membantu pengguna untuk dapat menentukan keputusan yang diambilnya hari itu.

Data kompleks berikutnya dikumpulkan dari mesin misalnya data dalam bidang bio informatika dan bidang kedokteran. Informasi yang harus diperas dari beragamnya data dari hasil foto radiologi pasien, hasil test darah, hasil elektrokardiogram yang mencatat aktifitas listrik jantung pasien.

Data yang dikumpulkan dari manusia beberapa di antaranya juga merupakan hasil dari proses bisnis. Data ini bisa berupa teks email, data web logs, post dari *website* dan sosial media, *customer feedback*, file suara, file video, berbagai jenis berkas dokumen dan presentasi, data kehadiran dan lain sebagainya.

1.4 Evolusi Pengelolaan Data

Pengelolaan data sering dipandang dari kacamata perangkat lunak walaupun harusnya bisa dipandang secara lebih menyeluruh. Hal ini karena setiap kemunculan suatu gelombang tren pengelolaan data yang baru, hal ini selalu dibarengi dengan perkembangan baru di bidang perangkat keras, media penyimpanan, jaringan komputer, serta kemajuan

model komputasi seperti teknologi virtual dan komputasi awan (*cloud computing*).

Data raya hadir bukan sebagai teknologi yang berdiri sendiri, bahkan ia lebih merupakan kombinasi kompleks dari perkembangan 50 tahun evolusi teknologi. Setiap fase pengelolaan data hadir sebagai jawaban untuk masalah yang muncul di saat tertentu. Contohnya saat RDBMS hadir, perusahaan diajak untuk memahami relasi antara elemen data. Saat kemudian perusahaan mulai menyimpan data-data tidak terstruktur (seperti : file suara, text, dan video), maka dibutuhkan jenis pendekatan baru seperti NLP (*Natural Language Processing*) untuk memberikan wawasan baru dalam menghasilkan analisa data. Dalam memahami gelombang tiap fase pengelolaan data ini, kita diajak bergerak dari keadaan di mana teknologi dirancang untuk menjawab “berapa banyak item yang terjual ke sejumlah pembeli?”, menuju pertanyaan yang lebih menantang. Semisal: “Bisakah data-data yang ada ini membentuk pola tren informasi yang paling berharga untuk keputusan bisnis? Bagaimana perusahaan berkelindan dengan sejumlah volume data yang masiv dengan cara yang tepat? Maka untuk memahami data raya adalah penting untuk memperhatikan bagaimana perubahan paradigma pengelolaan data berkembang menjawab tantangan sesuai eranya.

1.4.1 Gelombang 1 Manajemen Struktur Data

Akhir tahun 1960an adalah era di mana bidang komputasi merambah ke pasar komersial, di mana data disimpan dalam bentuk yang aslinya. Saat itu data disimpan tanpa mempertimbangkan bentuk struktur penyimpanan data, artinya data yang terlibat saat ini adalah data tidak terstruktur. Perusahaan harus melakukan metode *brute-force* bila ingin melihat data secara detail. Itu artinya semua data akan ditelusuri secara keseluruhan tanpa pengecualian dan seringkali tanpa prioritas pencarian.

Sekitar satu dekade berikutnya, *relational data model* dan RDBMS hadir menawarkan kehandalan dalam pengelolaan data dan pengelolaan laporan berkat adanya SQL (*Structural Query Language*). Di akhir tahun 1970an ini programmer lebih mudah memberikan jawaban yang memuaskan bagi perkembangan permintaan bisnis dalam menggali nilai data.

Pemenuhan kebutuhan penyimpanan atas meledaknya volume saat ini dapat diatasi dengan teknologi data warehouse. Data warehouse memungkinkan perusahaan pengelola data memilih subset data untuk disimpan dan diambil informasinya. Teknologi data warehouse dimunculkan dengan maksud membantu perusahaan melakukan analisa terhadap sejumlah besar volume data terstruktur perusahaan. Hal ini dilakukan dengan cara mengurangi volume data tersebut menjadi lebih kecil sehingga bisa lebih fokus pada data yang berkaitan dengan bisnis. Untuk selanjutnya data warehouse sering digunakan untuk menyimpan data dari beberapa tahun yang lalu agar perusahaan dapat mengambil analisa tren dan menyingkap suatu pola perilaku. Data warehouse dikomersilkan pada tahun 1990an dan masih berlaku hingga sekarang dengan berbagai improvisasi dalam sisi skala perangkat keras, teknologi virtualisasi dan kemampuannya mengintegrasikan perangkat lunak dan keras.

1.4.2 Gelombang 2: Manajemen Website dan Konten Website

Teknologi web mulai menjadi tren pada tahun 1990an menggoda perusahaan untuk juga mengelola data website berupa data-data tidak terstruktur semacam *web content*, gambar, audio dan video. Pasar kemudian memanfaatkan teknologi web untuk mengetengahkan solusi bisnis yang lebih solid dalam model yang dapat menyatukan pengelolaan model bisnis, pengelolaan kontrol pengguna, pengenalan informasi, pengelolaan teks dalam satu kolaborasi. Model solusi ini menjadi

alternatif yang sangat penting memenuhi kebutuhan perusahaan untuk mengelola semua urusan bisnis tersebut dengan cara yang logis.

Teknologi berbasis web ini menyadarkan banyak manusia di muka bumi ini tentang adanya potensi untuk saling terhubung antar individu dalam jaringan yang luas secara *real time*. Teknologi ini mendapat sambutan yang luas dan antusiasme yang tinggi. Terlebih lagi saat media sosial merebak dan digunakan secara luas oleh hampir semua orang dari berbagai kalangan dan usia. Pengalaman berinteraksi dan saling terhubung ini membawa konsekuensi ledakan data. Perkembangan interaksi perusahaan dengan perusahaan (B2B), perusahaan dengan konsumen (B2C) dan jenis model bisnis lainnya melalui konten web membawa konsekuensi semakin membesarnya volume data yang dihasilkan, semakin bervariasi data yang terkumpul, semakin tinggi kecepatan arus akses data. Hal ini mengingatkan perusahaan bahwa saluran data baru mengakibatkan terkumpulnya jenis data tertentu yang artinya dibutuhkan adanya cara pendekatan yang sesuai dengan jenis data yang baru.

1.4.3 Gelombang 3 : Pengelolaan Data Raya

Pengelolaan data raya menempati tempat tertinggi dalam evolusi praktek pengelolaan data dalam 5 dekade terakhir ini. Seperti telah dibahas sebelumnya bahwa pendekatan pengelolaan data raya ini hadir menjawab terjadinya kompulan data dengan volume yang amat besar, dimana terdapat permintaan akses data dalam waktu yang singkat, di mana data hadir dalam variasi format penyimpanan yang sangat beragam.

Pergerakan data raya bukan hanya berlaku pada dunia komersial dan bisnis saja. Aktivitas penelitian di bidang sains dan proses bisnis dalam pemerintahan juga saling terkait dalam memajukan teknologi data raya ini. Dengan adanya data raya kini dimungkinkan untuk memvirtualisasikan data sehingga

dapat disimpan secara efisien. Memanfaatkan penyimpanan berbasis awan juga dapat lebih menghemat biaya. Selain itu, peningkatan kecepatan dan keandalan jaringan telah menghilangkan keterbatasan fisik lainnya dalam kemampuan mengelola data dalam jumlah besar dengan kecepatan yang dapat diterima. Ditambah lagi dampak perubahan harga dan kecanggihan memori komputer. Dengan semua transisi teknologi ini, maka kini dapat dibayangkan cara-cara yang bisa dilakukan perusahaan untuk memanfaatkan data yang tidak terbayangkan beberapa tahun yang lalu.

1.5 Perlindungan bagi pemilik data

Pemanfaatan data raya diharuskan tunduk pada banyak persyaratan hukum. Perusahaan pengelola data raya diharuskan mematuhi peraturan perlindungan data, yang melindungi informasi pribadi pelanggan dari penggunaan yang tidak diinginkan. Peraturan tersebut, termasuk diantaranya adalah GDPR dan CCPA yang mengharuskan perusahaan untuk secara ketat melacak siapa yang mengakses data, membatasi akses tersebut, dan memberi tahu pelanggan bagaimana data mereka digunakan. Perusahaan yang gagal mengatur datanya tidak akan dapat mematuhi peraturan dan memiliki potensi menghadapi denda yang besar. GDPR merupakan singkatan dari *The General Data Protection Regulation*, sedangkan CCPA adalah *The California Consumer Privacy Act of 2018*.

1.5.1 GDPR

GDPR merupakan singkatan dari *The General Data Protection Regulation*. Dari situs resminya terdapat artikel utama yang berjudul “What is GDPR, the EU’s new data protection law?” yang menyediakan penjelasan tentang bagaimana GDPR akan memberlakukan sanksi yang berat terhadap mereka yang melanggar standar privasi dan

keamanannya, berupa denda yang mencapai puluhan juta euro. GDPR sendiri adalah Undang-undang privasi dan keamanan data di Eropa yang mulai berlaku pada tanggal 25 Mei 2018, mencakup persyaratan baru yang berjumlah ratusan halaman untuk organisasi di seluruh dunia dan merupakan undang-undang privasi dan keamanan terberat di dunia. Meskipun rancangan undang-undang ini dirancang dan disahkan oleh Uni Eropa (UE), peraturan ini mewajibkan organisasi di mana pun, selama mereka menargetkan atau mengumpulkan data terkait orang-orang di UE. Peraturan ini.

Peraturan GDPR sebagai hak atas privasi ini tidak lepas dari sejarah Konvensi Hak Asasi Manusia Eropa tahun 1950, yang menyatakan, “Setiap orang berhak untuk menghormati kehidupan pribadi dan keluarganya, rumah dan korespondensinya.” Atas dasar ini, Uni Eropa berupaya menjamin perlindungan hak ini melalui undang-undang.

1.5.2 CCPA

CCPA adalah singkatan dari *he California Consumer Privacy Act of 2018*. Informasi yang bisa diambil dari situs resminya (Bonta, 2024) menjelaskan bahwa Undang-Undang Privasi Konsumen California tahun 2018 (CCPA) memberi konsumen kontrol lebih besar atas informasi pribadi yang dikumpulkan oleh bisnis tentang mereka dan peraturan CCPA memberikan panduan tentang cara menerapkan undang-undang tersebut.

Undang-undang penting ini menjamin hak privasi baru bagi konsumen California, termasuk:

- Hak untuk mengetahui informasi pribadi yang dikumpulkan oleh suatu bisnis tentang mereka dan bagaimana informasi tersebut digunakan dan dibagikan;
- Hak untuk menghapus informasi pribadi yang dikumpulkan dari mereka (dengan beberapa pengecualian);

- Hak untuk tidak ikut serta dalam penjualan atau pembagian informasi pribadi mereka; Dan
- Hak atas non-diskriminasi dalam menjalankan hak CCPA mereka.

1.5.3 Undang-Undang Nomor 27 Tahun 2022

Pemerintah NKRI juga telah membuat peraturan terhadap data pribadi. Terdapat Undang-Undang Nomor 27 Tahun 2022 tentang Perlindungan Data Pribadi (UU PDP) yang mengatur hal tersebut. Disebutkan dalam laman www.mkri.id bahwa orang perorangan termasuk yang melakukan kegiatan bisnis atau e-commerce di rumah dapat dikategorikan sebagai pengendali data pribadi. Sehingga ia bertanggung jawab secara hukum atas pemrosesan data pribadi yang diselenggarakannya dan memenuhi ketentuan yang ada dalam UU PDP. Oleh karena itu, dalil kerugian dari Pemohon Perkara 108/PUU-XX/2022 sesungguhnya telah terakomodir dalam Pasal 2 ayat (1) UU PDP yang mengatur bahwa norma tersebut berlaku untuk setiap orang.

Keterangan yang disampaikan oleh Direktur Jenderal Aplikasi Informatika Kementerian Komunikasi dan Informatika Samuel Abrijani Pangerapan yang dalam sidang keempat Perkara Nomor 108/PUU-XX/2022 sekaligus Perkara Nomor 110/PUU-XX/2022 bertugas mewakili Pemerintah. Undang-Undang Nomor 27 Tahun 2022 tersebut dapat diakses pada laman <https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022>. UU ini mengatur mulai dari mengenai asas; jenis data pribadihingga ketentuan pidana terkait perlindungan data pribadi. Pemerintah NKRI berusaha memberikan perlindungan data pribadi sebagai keseluruhan upaya untuk melindungi data pribadi dalam rangkaian pemrosesan data pribadi sehingga dapat menjamin hak konstitusional subjek pemilik data pribadi tersebut.

DAFTAR PUSTAKA

- A. Katal, M. Wazid, and R. H. Goudar. *Big data: issues, challenges, tools and good practices*. in Proceedings of the 6th International Conference on Contemporary Computing (IC3 '13), pp. 404–409, IEEE, 2013
- Beal, V. (2024). *Big Data. Webopedia (2013)*. Available at: <https://www.webopedia.com/> (diakses pada Juni 2024)
- Bonta, Rob. (2024). *California Consumer Privacy Act (CCPA)*. Available at : <https://oag.ca.gov/privacy/ccpa> (diakses pada Juni 2024)
- BPK.go.id. (2022). *Undang-undang (UU) Nomor 27 Tahun 2022. Pelindungan Data Pribadi*. Available at : <https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022>. (diakses pada Juni 2024)
- Chunarkar-Patil P, Bhosale A. *Big data analytics*. Open Access J Sci. 2018;2(5):326–335. DOI: 10.15406/oajs.2018.02.00095
- Eberendu, A. Cecilia. *Unstructured Data: an overview of the data of Big Data*. International Journal of Computer Trends amd Technology (IJCTI) Vol. 38 number 1, pp 45-50, . DOI:10.14445/22312803/IJCTT-V38P109
- Feldman, R., & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press.
- GDPR.EU. (2024). *What is GDPR, the EU's new data protection law?*. Available at : <https://gdpr.eu/what-is-gdpr>. (diakses pada Juni 2024)

- Hanna, K.T, & Stedman, Craig. (2021). *Unstructured data*. Available at : <https://www.techtarget.com/searchbusinessanalytics/definition/unstructured-data>. (diakses pada Juni 2024)
- Hänig, C., Schierle, M., & Trabold, D. (2010). *Comparison of structured vs. unstructured data for Industrial Quality Analysis*. In Proceedings of The World Congress on Engineering and Computer Science, ISBN: 978-988-17012-0-6
- Mahkamah Institusi Republik Indonesia. *Pemerintah: UU Perlindungan Data Pribadi Beri Perlindungan Hukum*. Available at : <https://www.mkri.id/index.php?page=web.Berita&id=18915>. (diakses pada Juni 2024)
- Resources.data.gov. (2024). *Glossary: Data vs. information*. Available at: <https://resources.data.gov/glossary/data-vs-information/>. Diakses pada Juni 2024

BAB 2

URGENSI LITERASI DIGITAL DI ERA INFORMASI DIGITAL

Oleh Rizki Dewantara

2.1 Pendahuluan

Literasi digital memiliki urgensi yang sangat besar di era informasi digital saat ini. Di antara unsur-unsur penting budaya, kognitif, konstruktif, komunikatif, kreatif, kritis, dan kewarganegaraan keterampilan literasi digital sangat penting untuk melawan radikalisme. Selain itu, juga dapat memanfaatkan hal ini untuk meningkatkan keterampilan literasi digital. Faktanya, generasi milenial mampu menyebarkan informasi yang diterima dan memverifikasi keasliannya secara lebih menyeluruh berkat literasi digital (Bastian *et al.*, 2021). Dengan menekankan pentingnya pemahaman teknologi, kompetensi digital, perilaku online yang etis, dan etika digital, literasi digital dapat membantu mahasiswa mengembangkan standar moral dan etika yang tinggi dalam penggunaan teknologi digital. Upaya kolaboratif dibandingkan berbagai pihak berasal dari karakter pendidikan di era Society 5.0 dan mengatasi tantangan dan memastikan integrasi digital yang efektif (Sugiarto and Farid, 2023). Berikut adalah beberapa alasan mengapa literasi digital menjadi sangat penting:

1. Akses Luas ke Informasi: Di era digital saat ini, mengakses informasi sangatlah mudah melalui internet. Untuk meminimalkan risiko menerima informasi yang salah atau menyesatkan, literasi digital memungkinkan

individu mengakses, mengevaluasi, dan menggunakan informasi dengan hati-hati.

2. Pencegahan Penyebaran Informasi Palsu (Hoaks): Dengan kemudahan dalam menyebarkan informasi melalui berbagai platform digital, literasi digital membantu individu untuk mengidentifikasi dan memfilter informasi yang dapat menyesatkan atau merugikan. Hal ini penting dalam mencegah penyebaran hoaks dan informasi palsu.
3. Keamanan Digital: Literasi digital juga mencakup pemahaman terhadap keamanan digital. Pengguna yang melek digital dapat melindungi diri dari ancaman seperti malware, phishing, dan serangan cyber lainnya. Mereka juga dapat memahami pentingnya menggunakan kata sandi yang kuat, menjaga privasi online, dan menggunakan perangkat lunak keamanan.
4. Pengembangan Keterampilan Teknologi: Literasi digital membantu dalam mengembangkan keterampilan teknologi yang dibutuhkan untuk berpartisipasi dalam dunia kerja dan pendidikan yang semakin didigitalisasi. Ini melibatkan penggunaan perangkat lunak, aplikasi, dan platform online.
5. Partisipasi Aktif di Masyarakat Digital: Dengan literasi digital, individu dapat lebih aktif berpartisipasi dalam masyarakat digital. Mereka dapat terlibat dalam diskusi online, berkontribusi pada konten digital, dan memahami etika online yang diperlukan untuk berinteraksi secara positif di dunia maya.
6. Pemahaman Hak dan Tanggung Jawab Digital: Literasi digital juga mencakup pemahaman tentang hak dan tanggung jawab dalam ruang digital. Ini mencakup hak privasi, hak cipta, dan tanggung jawab dalam menggunakan teknologi dengan etika.
7. Pendidikan dan Pengembangan Karir: Banyak pekerjaan dan peluang pendidikan saat ini membutuhkan

keterampilan digital. Literasi digital menjadi dasar untuk pembelajaran dan pengembangan karir di berbagai sektor. Untuk menumbuhkan literasi digital, sektor pendidikan harus menerapkan perubahan tertentu, seperti:

- Memasukkan literatur digital ke dalam kurikulum nasional pendidikan anak usia dini
- untuk memampukan mahasiswa, dosen, tenaga kependidikan dan pimpinan untuk mengakses, memahami, dan mengoperasikan dengan media digital, alat komunikasi, dan jaringan.
- Membuat informasi baru dan menyajikannya secara ringkas

Literasi digital di kelas dapat membantu masyarakat mencapai tujuan pendidikan yang lebih baik dan mengembangkan keterampilan dalam penggunaan teknologi digital dengan mengikuti langkah-langkah berikut.

8. Adaptasi Terhadap Perubahan Teknologi: Dengan cepatnya perkembangan teknologi, literasi digital memungkinkan individu untuk lebih mudah beradaptasi dengan perubahan tersebut. Jika pemilik usaha kecil lebih sadar akan pentingnya literasi digital, mereka dapat memanfaatkan peluang yang ada di era ekonomi digital sebagai hasil dari kemampuan mereka dibandingkan dengan kebutuhan mereka dalam ekonomi digital. Misalnya, jika kebutuhan mereka adalah untuk meningkatkan hasil bisnis, mereka dapat memanfaatkan kemampuan mereka di bidang teknologi digital, yang didasarkan pada manfaat media sosial (Muhammad Ihsan Muzakki *et al.*, 2022). Kemampuan untuk memahami dan menguasai teknologi baru menjadi kunci untuk tetap relevan dalam dunia yang terus berkembang.

Secara keseluruhan, literasi digital bukan hanya sekadar keterampilan teknis, tetapi juga kemampuan untuk berpikir kritis, bersikap bijak, dan berpartisipasi secara efektif dalam masyarakat digital yang semakin kompleks. Selain itu, perlu adanya mekanisme untuk mengatasi fenomena ini karena dapat membingungkan dan mengasingkan masyarakat terhadap berita-berita kesehatan, dan informasi lainnya. Salah satu topik penting yang perlu dibahas adalah banyaknya berita terkait kesehatan di media sosial. Salah satu cara yang dapat dilakukan oleh pihak-pihak terkait, seperti institusi kesehatan, pemerintah, akademi, atau komunitas untuk mengatasi masalah ini adalah dengan memberikan literasi digital kepada masyarakat umum (Fitriarti, 2019).

2.2 Macam Literasi Digital di Era Informasi Digital

Literasi digital mencakup beberapa aspek yang mencerminkan kemampuan individu untuk beroperasi, berpartisipasi, dan berinteraksi secara efektif di dunia digital. Literasi digital seseorang dapat didefinisikan sebagai kemampuan seseorang dalam menggunakan dan memahami media digital dan alat komunikasi untuk mencari dan menciptakan informasi yang akurat dan dapat diandalkan. Di era digital ini, literasi digital penting bagi masyarakat secara keseluruhan karena membantu individu menghadapi ancaman digital, meningkatkan keterampilan dalam menggunakan teknologi digital, meningkatkan kemampuan berpikir kritis, kreatif, dan inovatif, mengurangi keterampilan berpikir negatif, meningkatkan kemampuan berpikir negatif, dan meningkatkan kemampuan berpikir negatif. literasi digital, dan membantu mereka mencapai tujuan pendidikan yang lebih baik. Literasi digital juga membantu individu dalam mengakses, memahami, dan berpartisipasi secara efektif di dunia digital. Di dunia yang semakin digital ini, tingkat literasi digital yang tinggi akan

memberikan banyak manfaat, terutama bagi kelompok yang sudah banyak berinvestasi pada manfaat teknologi. Untuk meningkatkan literasi digital, sektor pendidikan harus memasukkan literasi digital sebagai kurikulum inti nasional di sekolah dasar dan menengah, membekali siswa, guru, dan administrator sekolah dengan keterampilan untuk mengakses, memahami, dan menggunakan media digital, alat komunikasi, dan jaringan, serta untuk membuat dan menyebarkan informasi baru secara tepat waktu. Berikut adalah beberapa macam literasi digital yang penting di era informasi digital:

1. Literasi Informasi

Keterampilan untuk mencari, mengevaluasi, dan menggunakan informasi secara kritis dan efektif. Belanja online, keamanan data, dan penggunaan mata uang digital dengan hati-hati semuanya tercakup dalam hal ini.

2. Literasi Media

Pemahaman tentang cara media beroperasi, termasuk pemahaman terhadap pesan yang disampaikan oleh media, kemampuan membaca visual (seperti grafik atau video), dan kemampuan untuk memahami niat dan konteks di balik konten media.

3. Literasi Data

Kemampuan untuk bekerja dengan data, termasuk pengumpulan, analisis, dan interpretasi data. Literasi data melibatkan pengenalan pola, pemahaman visualisasi data, dan keterampilan analisis statistik dasar.

4. Literasi Teknologi

Pemahaman tentang bagaimana teknologi berfungsi dan kemampuan untuk menggunakan perangkat lunak dan perangkat keras dengan efektif. Ini mencakup pemahaman tentang jaringan, sistem operasi, dan aplikasi digital.

5. Literasi Keamanan Digital

Pemahaman tentang cara melindungi diri dari ancaman keamanan digital, termasuk penggunaan kata sandi yang kuat, identifikasi phishing, dan perlindungan terhadap malware dan serangan cyber lainnya.

6. Literasi Privasi

Kesadaran tentang hak privasi dan keterampilan untuk mengelola pengaturan privasi pada platform digital. Ini termasuk pemahaman tentang bagaimana data pribadi dikumpulkan, digunakan, dan disimpan oleh berbagai layanan online.

7. Literasi Etika Digital

Pemahaman tentang perilaku etis dalam interaksi online, termasuk penerapan norma-norma sosial dan etika dalam komunikasi digital, penggunaan media sosial, dan partisipasi dalam masyarakat digital.

8. Literasi Interkultural

Kemampuan untuk berkomunikasi dan berinteraksi secara efektif dengan individu dari latar belakang budaya yang berbeda dalam konteks digital. Ini melibatkan pemahaman tentang keberagaman dan penghargaan terhadap perspektif yang berbeda.

9. Literasi Kritis Digital

Kemampuan untuk berpikir kritis terhadap informasi dan konten digital, termasuk kemampuan untuk mengenali manipulasi informasi, bias, dan propaganda yang mungkin ditemui di dunia digital.

10. Literasi Pembuat Konten Digital

Kemampuan untuk membuat, mengedit, dan mendistribusikan konten digital. Ini mencakup keterampilan pengeditan gambar, video, atau audio, serta pemahaman tentang lisensi dan hak cipta. Kemampuan menganalisis informasi digital, berkolaborasi dalam menciptakan materi pembelajaran, strategi pembelajaran, dan alat penilaian, menggunakan jaringan

digital dan penyimpanan data, serta memahami norma, etika, dan standar sosial di dunia digital merupakan empat pilar yang menjadi landasan literasi digital. Di sisi lain, pentingnya dan manfaat literasi digital bagi guru antara lain: (1) penggunaan waktu dan proses pembelajaran lebih efisien dan efektif; (2) penyajian materi pembelajaran yang lebih kreatif, inovatif, dan interaktif; (3) penggunaan uang yang lebih efisien dan produktif; (4) dukungan terhadap kebijakan pemerintah dan masyarakat yang lebih aman dan Sejahtera (Rifai, 2021).

11. Literasi Partisipasi

Keterampilan untuk aktif berpartisipasi dalam masyarakat digital, termasuk kontribusi pada platform sosial, kolaborasi online, dan pemanfaatan alat-alat kolaboratif digital. Fungsi utama literasi digital saat ini adalah sebagai berikut: (1) mampu memanfaatkan berbagai hal secara digital; (2) tidak takut terhadap hal-hal digital seperti berpikir kreatif dan inovatif; (3) mampu bersosialisasi dengan hal-hal digital; (4) mampu berkolaborasi dengan banyak orang. Dengan cara ini, mahasiswa didorong untuk mengembangkan berbagai kemampuan, seperti kemampuan membaca, menulis, berbicara, berkolaborasi, bahkan berkeluarga. Inilah sebabnya mengapa mahasiswa mempunyai pengaruh yang signifikan terhadap kemajuan Indonesia menuju bangsa modern (Muliani *et al.*, 2021). Literasi digital tidak hanya mencakup aspek teknis, namun juga aspek sosial, etika, dan kritis. Kemampuan individu dalam beradaptasi terhadap perubahan dan menjadi pengguna yang andal dan mandiri di era informasi digital berkorelasi langsung dengan kualitas literasi digitalnya.

2.3 Contoh Urgensi Literasi Digital di Era Informasi Digital



Gambar 2. 1. Ukuran Keterampilan Literasi Digital

Diketahui ada beberapa indikator perolehan literasi digital pada generasi milenial, sesuai Gambar 2.1 di atas. Saat ini, literasi bukan hanya tentang membaca dan menulis; ini juga tentang bagaimana memahami informasi digital. Kami berharap dengan memaksimalkan literasi digital akan lebih mudah memahami konsep-konsep materi pelajaran. Mendorong untuk belajar mandiri dan menyesuaikan kegiatan pembelajaran dengan keterampilan kerja saat ini adalah tujuan dari pendidikan literasi digital (Salman Alparis Sormin, Ali Padang Siregar and Cipto Duwi Priyono, 2017). beberapa contoh urgensi literasi digital di era informasi digital:

1. Pencegahan Penyebaran Informasi Palsu

Literasi digital membantu individu untuk mengidentifikasi dan memverifikasi informasi sebelum menyebarkannya, sehingga dapat mengurangi risiko penyebaran berita palsu atau hoaks.

2. Keamanan Data Pribadi:

Melalui literasi digital, individu dapat belajar cara melindungi data pribadi mereka, menggunakan kata sandi yang kuat, dan mengenali upaya phishing atau serangan cyber lainnya.

3. Etika Online dan Cyberbullying

Literasi digital membantu dalam memahami etika online, termasuk cara berinteraksi secara positif dan menghindari perilaku cyberbullying. Ini membantu menciptakan lingkungan digital yang lebih aman dan ramah.

4. Kesadaran Privasi

Individu yang melek digital akan lebih sadar tentang bagaimana dan di mana data pribadi mereka digunakan. Mereka dapat mengelola pengaturan privasi mereka dan membuat keputusan yang lebih baik terkait penggunaan data.

5. Penggunaan Teknologi untuk Pendidikan

Literasi digital memungkinkan pendidik dan mahasiswa untuk memanfaatkan teknologi secara efektif dalam proses pembelajaran. Ini mencakup penggunaan platform pembelajaran online, sumber daya digital, dan alat kolaborasi untuk meningkatkan kualitas hidup, termasuk kesehatan dan gizi anak dan keluarga, literasi digital memungkinkan masyarakat umum mengakses, mengevaluasi, dan memahami berbagai bentuk informasi. Selain itu, dengan menyampaikan aspirasi melalui berbagai saluran, mereka dapat berpartisipasi dalam masyarakat, politik, dan seni. Masyarakat dapat menyampaikan pandangan dan pendapatnya tanpa melibatkan pihak lain melalui media digital (Meilinda, Malinda and Aisyah, 2020).



Gambar 2. 2. Foto Timbal Balik antara Pemateri dengan mahasiswa

Literasi digital penting bagi anak-anak dan orang dewasa dalam mengembangkan kemampuan kognitif, kecerdasan emosional, kematangan sosial, ketelitian akademis, keterampilan berpikir kritis, dan kematangan psikologis. Dengan meningkatkan kemampuan kognitif dan kemahiran berbahasa anak, literasi digital memungkinkan orang tua dengan mudah menemukan konten sesuai usia seperti cerita, gambar, lagu, dll (Lindriany, Hidayati and Muhammad, 2023). Penting bagi seluruh elemen dalam dunia pendidikan untuk mengembangkan literasi digital. Orang tua yang memiliki kemampuan literasi digital diyakini akan lebih mampu membimbing anaknya dalam menggunakan media digital sebagai sarana pembelajaran yang menarik dan menstimulasi, serta sebagai sarana mencegah kecanduan internet pada anak.

6. Adaptasi di Tempat Kerja:

Di lingkungan kerja, literasi digital sangat penting. Kemampuan untuk menggunakan alat dan aplikasi digital, berkolaborasi secara online, dan mengelola informasi digital menjadi faktor kunci dalam produktivitas dan keberhasilan karir.

7. Partisipasi Aktif dalam Diskusi Online:

Literasi digital memungkinkan individu untuk berpartisipasi dalam diskusi online dengan cara yang konstruktif dan informatif. Ada beberapa program literasi yang dapat dilaksanakan antara lain: 1) Sehingga 15 menit saat pembelajaran tidak dimulai, bukan meningkatkan keterampilan dan minat dalam membaca mahasiswa dan menguasai secara baik. Pojok baca kelas yang membuat mahasiswa membawa buku non pelajaran di setiap tahun untuk diumbangkan di sekolah 2 Ketiga, setiap minggu mahasiswa diwajibkan memberikan garis besar, rangkuman, dan pendapat berdasarkan buku yang dibacanya; setelah itu, mereka diharapkan mampu membaca nyaring di kelas. 4) Mengadakan kunjungan ke perpustakaan daerah untuk meningkatkan kemampuan membaca mahasiswa bisa. Mahasiswa akan sangat antusias mencari buku-buku untuk dibaca, seperti dongeng, novel, esai, dan cerita kejadian terkini yang mereka temukan di perpustakaan sekolah, poster motivasi yang berisi kutipan dari buku-buku yang pernah dibaca mahasiswa, dan buku catatan sastra yang berisi ringkasan buku yang telah dibaca mahasiswa (Fitriyani and Nugroho, 2022). Mereka dapat menyumbangkan pandangan mereka dengan lebih bijaksana dan memahami implikasi dari kata-kata mereka secara online.

8. Pemahaman Risiko Keamanan:

Mengenali dan menghindari risiko keamanan digital seperti malware, ransomware, dan serangan phishing adalah bagian dari literasi digital. Ini membantu melindungi perangkat dan data dari ancaman keamanan.

9. Penggunaan Platform Media Sosial yang Bertanggung Jawab:

Literasi digital membantu individu untuk menggunakan media sosial dengan bijak, termasuk memahami dampak penggunaan jangka panjang,

mengelola privasi, dan memilah informasi yang diterima dari berbagai sumber.

10. Pemahaman tentang Hak Cipta dan Lisensi:

Literasi digital membantu individu memahami hak cipta dan lisensi yang terkait dengan konten digital. Hal ini mendorong penggunaan dan berbagi konten secara legal dan etis.

Dengan literasi digital, individu dapat lebih efektif mengelola tantangan dan peluang yang muncul dalam era informasi digital, membantu mereka menjadi pengguna yang cerdas dan bertanggung jawab di dunia maya.

2.4 Keberlakuan Literasi Digital di Era Informasi Digital

Di era informasi digital ini, literasi digital menjadi hal yang cukup penting. Di dunia di mana begitu banyak informasi tersedia secara online, kemampuan untuk memahami dan menggunakan teknologi digital merupakan kebutuhan untuk kesuksesan sehari-hari. Literasi digital adalah sebuah pola pikir yang tidak dapat diabaikan lagi; ini berfungsi sebagai landasan untuk mengembangkan kinerja di bidang akademik, swasta, dan profesional. Mahasiswa diharapkan dapat mengarungi dunia digital dengan cara yang berbeda dibandingkan masa lalu, terutama di era Society 5.0. Mereka perlu berkomitmen terhadap lingkungan belajar yang lebih kreatif di kelas dan menyesuaikan tujuan pendidikan mereka untuk meningkatkan metode pengajaran yang pada akhirnya dapat mengarahkan mahasiswa menuju karier yang sukses. (Yuniarto and Yudha, 1945). Literasi digital melibatkan pemahaman tentang bagaimana mencari informasi secara efektif, mengevaluasi kebenaran informasi, dan melindungi diri dari ancaman online.

1. Pentingnya literasi digital di dunia modern

Literasi digital tidak hanya penting dalam kehidupan sehari-hari, tetapi juga sangat penting di dunia bisnis. Para pengusaha yang memiliki pemahaman mendalam tentang teknologi dan media digital memiliki keuntungan kompetitif yang besar dalam memajukan bisnis mereka. Mereka dapat memanfaatkan media sosial, pemasaran online, dan alat digital lainnya untuk mencapai pasar yang lebih luas dan meningkatkan daya saing mereka. Siapa pun dapat mengakses informasi secara efisien dan efektif, melakukan analisis kritis terhadap informasi, dan memanfaatkan informasi tersebut dengan lebih baik melalui literasi digital ini (Sujana and Rachmatin, 2019).

Namun, masih banyak orang yang kurang memahami literasi digital. Beberapa alasan mengapa hal ini terjadi termasuk kurangnya akses terhadap teknologi, kurangnya pelatihan dan pendidikan, dan kurangnya kesadaran tentang pentingnya literasi digital. Oleh karena itu, penting bagi kita untuk meningkatkan pemahaman dan kesadaran kita tentang literasi digital agar dapat menghadapi tantangan dunia digital dengan lebih baik.

2. Statistik dan tren literasi digital

Sebelum membahas lebih lanjut tentang pentingnya literasi digital, mari kita lihat beberapa statistik dan tren yang terkait dengan literasi digital. Menurut sebuah penelitian, hanya sekitar 50% populasi dunia yang memiliki akses ke internet. Jumlah ini menunjukkan adanya kesenjangan digital yang signifikan di berbagai negara.

Selain itu, penelitian juga menunjukkan bahwa banyak orang yang tidak memiliki keterampilan dasar dalam menggunakan teknologi digital. Mereka kesulitan dalam mencari informasi secara online, membedakan informasi yang benar dan salah, dan melindungi diri dari

ancaman online. Ini menunjukkan perlunya meningkatkan literasi digital secara global.

3. Dampak literasi digital pada pendidikan dan prospek karir

Literasi digital memiliki dampak yang signifikan pada pendidikan dan prospek karir seseorang. Di dunia pendidikan, literasi digital memainkan peran penting dalam meningkatkan cara belajar dan mengajar. Guru dapat menggunakan teknologi digital untuk menyampaikan materi pelajaran dengan cara yang lebih interaktif dan menarik, sehingga meningkatkan minat dan pemahaman mahasiswa. Telah terjadi transformasi pesat dalam cara pendanaan pendidikan, khususnya dalam bidang pengajaran di kelas. Untuk memenuhi tuntutan perkembangan kurikulum dan kehidupan modern, integrasi teknologi informasi ke dalam proses pembelajaran menjadi suatu kondisi yang penting. Proses pembelajaran tidak terbatas pada ruang kelas, seperti halnya pendidikan tradisional; memang, hal ini masih didukung dan dilaksanakan; Namun di era digital ini, pembelajaran juga bisa dilakukan secara virtual, online (Kholid, 2020).

Literasi digital juga berpotensi meningkatkan prospek karir seseorang. Dalam lingkungan kerja yang didorong oleh teknologi, kemampuan untuk menggunakan alat dan platform digital merupakan komoditas yang sangat berharga. Orang-orang dengan literasi digital yang kuat mempunyai peluang lebih besar untuk meraih kesuksesan dan kemajuan dalam karier mereka.

4. Keterampilan dan kompetensi literasi digital

Untuk menjadi literat digital, ada beberapa keterampilan dan kompetensi yang perlu dimiliki. Pertama, seseorang perlu menguasai keterampilan dasar penggunaan komputer dan internet. Ini meliputi pemahaman tentang bagaimana menggunakan browser,

mencari informasi secara efektif, dan mengelola file dan data secara online.

Pentingnya literasi digital tidak bisa dilebih-lebihkan dan telah menjadi prioritas dalam pendidikan, khususnya selama krisis pendidikan berbasis Daring. Peran literasi digital dalam pendidikan bukan lagi sebuah pilihan, namun sangat penting (Simanjuntak, 2022). Selain itu, seseorang juga perlu memahami cara mengelola privasi dan keamanan online. Ini melibatkan pemahaman tentang bagaimana melindungi data pribadi, menghindari penipuan online, dan mengelola kata sandi dengan aman. Keterampilan literasi digital juga mencakup kemampuan untuk memahami dan mengevaluasi informasi yang ditemukan secara online, serta menggunakan alat-alat digital seperti program pengolah kata dan spreadsheet.

5. Strategi untuk meningkatkan literasi digital

Untuk meningkatkan literasi digital, ada beberapa strategi yang dapat diikuti. Pertama, penting untuk meningkatkan akses terhadap teknologi digital. Ini dapat dilakukan melalui program pemerintah yang menyediakan akses internet gratis atau subsidi untuk perangkat digital. Selain itu, terdapat komponen penting untuk mengembangkan literasi digital (Ahmad, 2022), yaitu sebagai berikut. 1) Kognitif, yaitu daya pikir dalam menilai konten; 2) Budaya, yaitu pemahaman ragam konteks pengguna secara digital; 3. konstruktivis 4) Komunikasi, yaitu memahami peran branding dan komunikasi digital; 5) Rasa percaya diri yang tidak tergoyahkan; 6) Kreativitas, melakukan hal-hal baru dengan cara-cara baru; 7) Krisis dalam evaluasi isi; dan 8) Kepercayaan diri sosial.

Selain itu, pendidikan dan pelatihan literasi digital juga perlu ditingkatkan. Sekolah dan perguruan tinggi dapat menawarkan program yang fokus pada pengembangan keterampilan literasi digital. Program-

program ini dapat mencakup pelatihan dalam penggunaan teknologi, penilaian informasi online, dan keamanan digital.

6. Sumber daya untuk pelatihan dan pendidikan literasi digital

Ada beberapa sumber daya yang tersedia untuk membantu meningkatkan literasi digital. Sejumlah institusi pendidikan dan organisasi nirlaba menyediakan kursus online gratis yang mencakup berbagai topik literasi digital. Kemampuan memahami dan menggunakan informasi yang tidak bias serta dapat diakses kapan saja dan dari mana saja melalui internet merupakan inti dari literasi digital. Pentingnya literasi digital, termasuk teknologi informasi, media, dan komunikasi, tidak bisa dilebih-lebihkan. Mahasiswa tidak hanya diharapkan memahami bentuk sastra tradisional, seperti menulis dan membaca, namun mereka juga harus memahami pentingnya literasi digital. Selain itu, ada juga banyak buku, video tutorial, dan blog yang dapat membantu meningkatkan pemahaman tentang literasi digital. Penting untuk mengambil inisiatif dalam mencari dan memanfaatkan sumber daya ini untuk meningkatkan literasi digital kita.

7. Peran literasi digital dalam berpikir kritis dan literasi media

Literasi digital tidak hanya berhubungan dengan penggunaan teknologi, tetapi juga berhubungan dengan kemampuan berpikir kritis dan literasi media. Strategi yang dapat diterapkan oleh orang dewasa untuk meningkatkan keterampilan literasi digital anak-anak termasuk meningkatkan pemahaman membaca anak-anak, menyediakan situs web dan aplikasi berbasis pendidikan, dan memfasilitasi pengembangan literasi digital mereka (Ningsih, Widodo and Asrin, 2021). Literasi digital melibatkan kemampuan untuk memahami dan mengevaluasi informasi yang ditemukan secara online.

Dalam era informasi yang berlimpah, penting bagi kita untuk dapat membedakan antara informasi yang benar dan salah. Literasi digital membantu kita mengembangkan keterampilan kritis yang diperlukan untuk memilah dan menilai informasi dengan bijak.

8. Literasi digital dan keamanan online

Selain memahami dan menggunakan teknologi digital, penting juga untuk memahami ancaman dan risiko online. Dalam dunia yang terhubung secara digital, kita rentan terhadap serangan cyber, penipuan online, dan pelanggaran privasi.

Oleh karena itu, literasi digital juga meningkatkan pemahaman tentang keamanan online. Di antara fungsinya adalah pencegahan penggunaan kata yang ceroboh, penghapusan petunjuk dan saran yang menyesatkan, dan penyimpanan informasi pribadi yang aman.

2.5 Dampak Negatif Literasi Digital

Bagi masyarakat umum, menurunnya literasi digital dapat menimbulkan dampak negatif, seperti:

1. penyebaran berita cetak (*hoaks*): Maraknya literasi digital berpotensi membuat masyarakat lebih mudah menerima berita cetak (*hoaks*).
2. Masalah privasi: Pengguna yang tidak menyadari pentingnya praktik privasi dan keamanan digital mungkin merupakan ancaman.
3. kompetensi teknologi: penurunan literasi digital dapat menyebabkan individu mengembangkan kompetensi teknologi dan menghabiskan lebih banyak waktu di dunia digital.

4. Perpenahanan sosial: Berita palsu dan opini yang tidak akurat dengan jaringan digital terkadang memicu konflik dan perpenahanan sosial.
5. Kejahatan dunia maya merupakan ancaman bagi masyarakat di era digital ini. Kemampuan literasi digital diperlukan agar masyarakat umum dapat mengidentifikasi kejahatan siber.

Menurunnya literasi digital dapat membuat masyarakat semakin rentan terhadap berbagai ancaman dan tantangan di dunia digital. Oleh karena itu, penting bagi sektor pendidikan untuk memasukkan literasi digital sebagai kurikulum nasional di sekolah dasar, membekali siswa, guru, dan administrator sekolah dengan keterampilan untuk mengakses, memahami, dan menggunakan media digital, alat komunikasi, dan jaringan, serta menciptakan informasi baru dan menyajikannya secara seimbang.

2.6 Kesimpulan

Mengadopsi literasi digital merupakan faktor keberhasilan dan pertumbuhan penting bagi pengembangan pribadi dan profesional di dunia yang semakin terhubung secara digital. Kemampuan membaca, menulis, berbicara, dan memahami teks digital memungkinkan kita melindungi diri dari predator online, meningkatkan pendidikan dan karier, serta memanfaatkan teknologi dan informasi digital secara efektif. Untuk meningkatkan literasi digital, penting untuk meningkatkan akses terhadap teknologi, meningkatkan pendidikan dan pelatihan literasi digital, dan inovatif dalam mencari sumber daya yang tersedia. Dengan mengadopsi literasi digital, kita dapat memanfaatkan kekuatan teknologi dan informasi digital untuk pertumbuhan pribadi dan profesional kita.

DAFTAR PUSTAKA

- Ahmad, I. F. (2022) 'Urgensi Literasi Digital di Indonesia pada Masa Pandemi COVID-19: Sebuah Tinjauan Sistematis', *Nusantara: Jurnal Pendidikan Indonesia*, 2(1), pp. 1–18.
- Bastian, O. A. *et al.* (2021) 'Urgensi Literasi Digital dalam Menangkal Radikalisme pada Generasi Millenial di Era Revolusi Industri 4 . 0', *Dinamika Sosial Budaya*, 23(1), pp. 126–133.
- Fitriarti, E. A. (2019) 'Urgensi Literasi Digital Dalam Menangkal Hoax Informasi Kesehatan Di Era Digital', *Journal Of Communication Studies*, 4(2).
- Fitriyani and Nugroho, A. T. (2022) 'Literasi Digital di Era Pembelajaran Abad 21', *El-Mujtama: Jurnal Pengabdian Masyarakat*, 2(2), pp. 201–208. doi: 10.47467/elmujtama.v2i2.1088.
- <https://aptika.kominfo.go.id/2020/06/urgensi-literasi-digital-bagi-masa-depan-ruang-digital-indonesia/>
- <https://www.kompasiana.com/ghosalina15839/654b8a94ee794a09181733f2/dampak-literasi-digital>
- Kholid (2020) 'Pentingnya Literasi Digital Bagi Guru Pada Lembaga Pendidikan Tingkat Dasar Dan Implikasinya Terhadap Penyelenggaraan Kegiatan Belajar Mengajar', *Jurnal Horizon Pedagogia*, 1(1).
- Lindriany, J., Hidayati, D. and Muhammad, D. (2023) 'Urgensi Literasi Digital Bagi Anak Usia Dini Dan Orang Tua', *JET: Journal of Education and Teaching*, 4(1), pp. 35–49. doi: 10.51454/jet.v4i1.201.
- Meilinda, N., Malinda, F. and Aisyah, S. M. (2020) 'Literasi Digital Pada Remaja Digital (Sosialisasi Pemanfaatan Media Sosial Bagi Pelajar Sekolah Menengah Atas)', *Jurnal Abdimas Mandiri*, 4(1), pp. 62–69.

- Muhammad Ihsan Muzakki *et al.* (2022) 'Pentingnya Edukasi Literasi Digital Bagi Pelaku Usaha Mikro (Kelompok 3 Divisi UMKM)', *HASPI Jurnal Penelitian & Pengabdian*, 1(1), pp. 1–8.
- Muliani, A. *et al.* (2021) 'Pentingnya Peran Literasi Digital bagi Mahasiswa di Era Revolusi Industri 4.0 untuk Kemajuan Indonesia', *Journal of Education and Technology*, 1(2), pp. 87–92.
- Ningsih, I. W., Widodo, A. and Asrin (2021) 'Urgensi Kompetensi Literasi Digital dalam Pembelajaran pada Masa Pandemi Covid-19', *Jurnal Inovasi Teknologi Pendidikan Volume*, 8(2), pp. 132–139.
- Rifai, A. (2021) 'Urgensi Literasi Digital Bagi Guru Smp Yabujah Di Masa Pandemi Covid-19', *Jurnal Sinau*, 7(2), pp. 58–70.
- Salman Alparis Sormin, Ali Padang Siregar and Cipto Duwi Priyono (2017) 'Konsepsi Literasi Digital Dalam Pembelajaran Sejarah Di Era Disruptif', *Seminar Nasional Sejarah ke 4 Jurusan Pendidikan Sejarah Universitas Negeri Padang*, pp. 647–662.
- Simanjuntak, M. M. (2022) 'Analisis Urgensi Penggunaan Literasi Digital dalam Pelaksanaan Pendidikan pada Masa Pandemi di Sekolah Menengah Pertama', *Edukatif: Jurnal Ilmu Pendidikan*, 4(2), pp. 2599–2608.
- Sugiarto and Farid, A. (2023) 'Literasi Digital Sebagai Jalan Penguatan Pendidikan Karakter Di Era Society 5.0', *Cetta: Jurnal Ilmu Pendidikan*, 6(3), pp. 580–597.
- Sujana, A. and Rachmatin, D. (2019) 'Literasi digital abad 21 bagi mahasiswa PGSD: apa , mengapa , dan bagaimana', *Conference Series Journal*, 1(1), pp. 1–7.
- Yuniarto, B. and Yudha, R. P. (1945) 'Literasi Digital Sebagai Penguatan Pendidikan Karakter Menuju Era Society 5.0', *X(2)*, pp. 176–194.

BAB 3

ETIKA DALAM DUNIA DIGITAL

Oleh Reska Mayefis

3.1 Definisi Etika Digital

Dunia digital telah menjadi bagian penting dari kehidupan kita sehari-hari, yang mempengaruhi cara kita berkomunikasi, bekerja, dan berinteraksi dengan orang lain. Seiring dengan kemajuan teknologi yang pesat, penting untuk mempertimbangkan implikasi etis yang menyertainya. Konsep etika di dunia digital yang dikenal dengan etika digital berfokus pada prinsip dan nilai panduan yang mengatur perilaku kita secara online. Hal ini mencakup isu-isu seperti privasi data, keamanan siber, dan dampak kecerdasan buatan terhadap masyarakat. Dalam bab ini, kita akan mengeksplorasi pentingnya menegakkan standar etika di dunia digital dan bagaimana hal ini dapat membentuk teknologi masa depan. Dengan mengkaji pertimbangan etis dalam interaksi digital, kita dapat memastikan pendekatan yang lebih transparan dan bertanggung jawab dalam memanfaatkan teknologi demi kemajuan masyarakat.

Etika digital merupakan bidang studi yang membahas perilaku dan keputusan etis dalam konteks dunia digital, termasuk penggunaan teknologi informasi dan komunikasi. Definisi etika digital seringkali mencakup prinsip-prinsip moral yang harus diikuti dalam interaksi online, pengelolaan data, privasi, keamanan siber, dan dampak teknologi terhadap masyarakat.

Beberapa unsur yang sering diakui dalam etika digital melibatkan:

1. Privasi dan Keamanan:

- a. Bagaimana data pribadi dihormati dan dijaga.
- b. Upaya untuk mencegah akses yang tidak sah terhadap informasi pribadi.

Etika digital mencakup perlindungan terhadap privasi dan keamanan dalam penggunaan teknologi informasi. Prinsip ini menekankan pentingnya menghormati dan menjaga data pribadi individu agar tidak disalahgunakan. Selain itu, etika digital juga menitikberatkan pada upaya pencegahan terhadap akses yang tidak sah terhadap informasi pribadi oleh pihak yang tidak berwenang.

2. Kejujuran dan Transparansi:

- a. Bagaimana informasi disampaikan secara jujur kepada pengguna.
- b. Pemberian informasi yang cukup kepada pengguna terkait pengumpulan dan penggunaan data.

Dalam konteks etika digital, kejujuran dan transparansi berkaitan dengan bagaimana informasi disampaikan secara jujur kepada pengguna. Hal ini mencakup memberikan informasi yang memadai kepada pengguna mengenai pengumpulan dan penggunaan data. Prinsip ini menekankan perlunya komunikasi yang terbuka dan jelas agar pengguna dapat membuat keputusan informasi dengan pemahaman yang baik.

3. Tanggung Jawab Digital:

- a. Kesadaran akan dampak dari tindakan online.
- b. Mengambil tanggung jawab atas perilaku dan konten yang dibagikan di dunia digital.

Etika digital mendorong kesadaran akan dampak dari tindakan online dan menuntut tanggung jawab atas perilaku serta konten yang dibagikan di dunia digital. Ini mencakup pemahaman bahwa tindakan online dapat memiliki konsekuensi nyata, dan individu perlu bertanggung jawab atas kontribusi mereka terhadap lingkungan digital.

4. Keragaman dan Inklusivitas:

Mendorong keberagaman pendapat dan memastikan bahwa teknologi dapat diakses oleh semua lapisan masyarakat. Prinsip etika digital ini menekankan pentingnya mendorong keberagaman pendapat dan memastikan bahwa teknologi dapat diakses oleh semua lapisan masyarakat. Ini bertujuan untuk menghindari eksklusivitas digital dan mempromosikan partisipasi yang merata dari berbagai kelompok dan komunitas.

5. Kekuatan Algoritma

Memahami dan mengatasi potensi bias dalam algoritma dan keputusan otomatis. Etika digital mencerminkan kekhawatiran terhadap potensi bias dalam algoritma dan keputusan otomatis. Prinsip ini menuntut pemahaman dan mitigasi terhadap ketidaksetaraan yang mungkin muncul akibat penggunaan teknologi, dengan memastikan bahwa algoritma bekerja adil dan memberikan hasil yang netral.

6. Kode Etik Profesional:

Panduan etika untuk para profesional IT dan teknologi informasi. Dalam lingkup etika digital, terdapat panduan etika khusus untuk para profesional IT dan teknologi informasi. Ini mencakup tanggung jawab moral dalam pengembangan, implementasi, dan pemeliharaan teknologi. Para profesional diharapkan untuk beroperasi sesuai dengan norma-norma etika yang diterapkan di dalam industri mereka.

3.2 Pentingnya Etika Digital

Di dalam dunia digital, pentingnya etika tidak bisa dianggap tidak penting. Ketika kehidupan kita semakin terkait dengan teknologi, memastikan standar etika ditegakkan sangatlah penting bagi kesejahteraan individu dan masyarakat secara keseluruhan. Pertimbangan etis di dunia digital mencakup berbagai masalah, mulai dari masalah privasi dan keamanan data hingga dampak algoritma terhadap proses pengambilan keputusan. Tanpa landasan etika yang kuat, potensi penyalahgunaan dan penyalahgunaan teknologi digital akan sangat besar. Sebagaimana dikemukakan (F. A. Ferreira et al., 2019), prinsip etika memberikan kerangka kerja untuk memandu perilaku dan pengambilan keputusan di ruang digital, membantu melindungi pengguna dan mendorong penggunaan teknologi yang adil dan bertanggung jawab. Oleh karena itu, memasukkan pertimbangan etis ke dalam praktik digital sangat penting untuk menciptakan dunia digital yang lebih etis dan berkelanjutan.

Etika digital menjadi sangat penting dalam kehidupan saat ini karena perkembangan teknologi informasi dan komunikasi yang pesat, yang telah mengubah cara kita berinteraksi, bekerja, dan hidup sehari-hari. Berikut adalah beberapa alasan mengapa etika digital memiliki peran yang krusial:

1. **Privasi dan Keamanan:** Etika digital memberikan perlindungan terhadap privasi individu dalam lingkungan digital. Dengan melibatkan prinsip-prinsip etika, pengguna dapat merasa lebih aman dan yakin bahwa informasi pribadi mereka tidak akan disalahgunakan atau diakses tanpa izin. Sebagai contoh, adopsi kebijakan privasi oleh perusahaan teknologi membantu melindungi data pengguna dan membangun kepercayaan.
2. **Tanggung Jawab Individu:** Etika digital mendorong tanggung jawab individu terhadap perilaku online mereka. Hal ini melibatkan kesadaran akan dampak dari tindakan di

- dunia digital, termasuk penggunaan sosial media dan berbagi informasi. Dengan penerapan etika digital, individu dapat menghindari tindakan yang dapat merugikan diri mereka sendiri maupun orang lain.
3. Pencegahan Bias Algoritma: Dalam konteks kecerdasan buatan dan penggunaan algoritma, etika digital menjadi penting untuk mencegah dan mengurangi bias yang mungkin muncul. Misalnya, dalam penggunaan algoritma rekrutmen, upaya etika digital diperlukan untuk memastikan bahwa algoritma tidak memperkuat atau menciptakan ketidaksetaraan dalam kesempatan kerja.
 4. Perlindungan Terhadap Kejahatan Siber: Etika digital membantu melindungi individu dan organisasi dari ancaman keamanan siber. Ini melibatkan perilaku yang aman secara digital, penggunaan sandi yang kuat, dan kesadaran terhadap potensi serangan phishing. Keamanan siber yang diperkuat oleh etika digital dapat mengurangi risiko pencurian identitas dan pelanggaran data.
 5. Kesetaraan dan Akses: Etika digital mendukung kesetaraan dan inklusivitas dalam akses teknologi. Prinsip ini memastikan bahwa teknologi dapat diakses oleh berbagai kelompok masyarakat, termasuk mereka yang mungkin memiliki keterbatasan akses atau pengetahuan teknologi yang terbatas.
 6. Kode Etik Profesional: Dalam dunia bisnis dan profesi teknologi, kode etik profesional menjadi panduan dalam menjalankan tanggung jawab moral. Etika digital membantu melindungi kepercayaan konsumen dan memastikan bahwa produk dan layanan teknologi dikembangkan dan dikelola dengan integritas dan kejujuran.

Dengan menerapkan etika digital, kita dapat menciptakan lingkungan digital yang lebih aman, adil, dan berkelanjutan, yang mendukung perkembangan teknologi tanpa mengabaikan nilai-nilai etika yang mendasar.

3.3 Tantangan Etika Digital

Perkembangan teknologi digital dalam masyarakat modern telah membawa tantangan etika yang besar yang harus diatasi. Munculnya internet, media sosial, dan data besar telah merevolusi cara kita berkomunikasi, mengakses informasi, dan menjalankan bisnis. Namun, revolusi digital ini juga menimbulkan kekhawatiran mengenai privasi data, keamanan siber, dan implikasi etika dari kecerdasan buatan. Saat kita bergulat dengan isu-isu seperti berita palsu, pelecehan online, dan bias algoritma, penting bagi kita untuk mengembangkan kerangka kerja komprehensif untuk perilaku digital yang etis.

Tantangan etika di bidang digital menjadi semakin kompleks seiring dengan kemajuan teknologi. Masalah-masalah seperti privasi data, keamanan siber, dan kecerdasan buatan menimbulkan kekhawatiran etika yang signifikan. Misalnya, pengumpulan dan penggunaan data pribadi oleh perusahaan dan pemerintah tanpa persetujuan, melanggar hak privasi individu (Numa Numa et al., 2024). Selain itu, penerapan AI dalam proses pengambilan keputusan dapat menimbulkan hasil yang bias dan meningkatkan kesenjangan sosial. Ketika teknologi semakin terintegrasi ke dalam kehidupan kita sehari-hari, kebutuhan akan pedoman dan peraturan etika di dunia digital menjadi sangat penting. Pedoman ini harus memastikan transparansi, akuntabilitas, dan perlindungan hak-hak individu dalam dunia digital yang berkembang pesat. Pertimbangan etis harus menjadi yang terdepan dalam perkembangan teknologi untuk mencegah bahaya dan meningkatkan kepercayaan terhadap sistem digital.

Implikasi etis dari teknologi digital telah menjadi perhatian yang mendesak dalam masyarakat kontemporer. Ketika kemajuan teknologi ini terus merasuki setiap aspek kehidupan kita, pertanyaan tentang privasi, keamanan, dan penggunaan data yang etis pun muncul. Pesatnya perkembangan kecerdasan

buatan, bioteknologi, dan sistem pengawasan menimbulkan isu-isu penting mengenai otonomi, persetujuan, dan akuntabilitas. (Silvia Rahmelia dkk., 2023) Individu rentan terhadap potensi pelanggaran informasi pribadi mereka, sehingga menimbulkan kekhawatiran tentang manipulasi dan pengawasan data. Selain itu, pertimbangan etis seputar penggunaan algoritme dalam proses pengambilan keputusan menyoroti perlunya transparansi dan keadilan dalam sistem digital. Penting untuk menavigasi tantangan etika ini dengan pertimbangan yang cermat dan langkah-langkah proaktif untuk memastikan bahwa teknologi digital memberikan manfaat bagi kebaikan bersama dengan tetap menghormati prinsip-prinsip etika yang mendasar.

Tantangan dalam menjaga etika digital mencakup berbagai kompleksitas yang muncul dalam lanskap teknologi dan informasi yang terus berkembang. Salah satu tantangan besarnya adalah implikasi etika seputar privasi dan keamanan data (Sabilar Rosyad dkk., 2023). Ketika individu dan organisasi memanfaatkan kekuatan platform digital, isu-isu seperti pelanggaran data, akses tidak sah, dan komodifikasi informasi pribadi menjadi perhatian utama. Selain itu, pesatnya kemajuan teknologi sering kali melampaui perkembangan pedoman dan peraturan etika, sehingga menyebabkan keterlambatan dalam mengatasi dilema etika yang muncul. Pertimbangan etis di bidang-bidang seperti *Artificial Intelligence* (AI) dan pembelajaran mesin semakin memperumit lanskap ini, sehingga menyoroti perlunya kerangka kerja komprehensif yang memprioritaskan prinsip-prinsip etika. Untuk mengatasi tantangan ini secara efektif, para pemangku kepentingan harus terlibat dalam dialog, pendidikan, dan kolaborasi berkelanjutan untuk menumbuhkan budaya kesadaran etis dan tanggung jawab di dunia digital.

3.4 Strategi untuk Mempromosikan Perilaku Etis Secara Online

Strategi untuk mempromosikan perilaku etis secara online sangatlah penting di dunia digital yang berkembang pesat. Salah satu pendekatan yang efektif adalah dengan menerapkan kode etik atau pedoman etika yang secara jelas mendefinisikan perilaku yang dapat diterima dan konsekuensi atas pelanggaran. Pedoman ini harus dikomunikasikan secara efektif kepada semua individu yang terlibat secara online untuk memastikan pemahaman bersama mengenai ekspektasi etika. Strategi penting lainnya adalah menumbuhkan budaya akuntabilitas dan transparansi, yang mana individu didorong untuk melaporkan perilaku tidak etis tanpa rasa takut akan pembalasan. Selain itu, pendidikan memainkan peran penting dalam mempromosikan perilaku etis secara online dengan meningkatkan kesadaran akan potensi konsekuensi dari tindakan tidak etis dan memberikan individu alat yang diperlukan untuk membuat keputusan etis. Dengan menggabungkan strategi-strategi ini, organisasi dan individu dapat berkontribusi pada lingkungan digital yang lebih etis yang mendorong kepercayaan dan integritas (Zachary Kilhoffer et al., 2023).

Selain itu, melalui program pendidikan dan kesadaran yang menekankan pentingnya perilaku etis di internet juga perlu dilakukan. Mendidik pengguna tentang konsekuensi perilaku tidak etis, seperti penindasan maya atau pelanggaran privasi, dapat mendorong mereka untuk bertindak lebih bertanggung jawab saat online. Selain itu, organisasi dapat menerapkan kode etik yang jelas dan komprehensif untuk platform online mereka, menguraikan perilaku yang diharapkan dan konsekuensi dari pelanggaran. Pedoman ini dapat membantu menciptakan budaya perilaku etis di kalangan pengguna. Selain itu, penerapan solusi teknologi seperti alat moderasi konten yang didukung AI dapat membantu mengidentifikasi dan mengatasi

perilaku tidak etis secara real-time, sehingga mendorong lingkungan online yang lebih aman bagi semua pengguna. Dengan menggabungkan strategi pendidikan, organisasi, dan teknologi, perilaku etis online dapat dipromosikan dan dipertahankan secara efektif. (Novi Sekar Sari dkk., 2023)

3.5 Peran Pendidikan dalam Menumbuhkan Etika Digital

Pendidikan berperan penting dalam membentuk sikap dan perilaku individu, khususnya dalam ranah etika digital. Melalui program pendidikan terstruktur, siswa dapat dibekali dengan pengetahuan dan keterampilan yang diperlukan untuk menghadapi dilema etika kompleks yang melekat di dunia digital. Dengan mengintegrasikan etika digital ke dalam kurikulum, pendidik dapat menumbuhkan pemikiran kritis, empati, dan nilai-nilai seperti integritas dan menghormati privasi. Selain itu, penalaran etis dan proses pengambilan keputusan dapat diasah melalui studi kasus dan contoh dunia nyata, mempersiapkan siswa untuk membuat pilihan berdasarkan informasi dalam lanskap teknologi yang berkembang pesat. Pada akhirnya, pendidikan berfungsi sebagai landasan untuk mengembangkan warga digital yang bertanggung jawab dan menjunjung standar etika dalam interaksi online dan penyebaran informasi. Dengan menekankan pentingnya pertimbangan etis dalam ruang digital, pendidikan dapat menciptakan masyarakat yang lebih sadar etis dan bertanggung jawab secara sosial di era digital. (Maureen E. Pengawal, 2020).

3.6 Kesimpulan

Kesimpulannya, studi etika di dunia digital sangat penting dalam memastikan penggunaan teknologi secara bertanggung jawab dan perlindungan hak-hak individu. Melalui analisis terhadap berbagai kerangka etika, terlihat jelas bahwa prinsip kemurahan hati, non-maleficence, keadilan, dan otonomi memberikan landasan yang kuat dalam pengambilan keputusan etis dalam konteks digital. Dengan menerapkan pedoman etika dan kode etik, individu dan organisasi dapat menavigasi lanskap interaksi digital yang kompleks dengan integritas dan akuntabilitas. Sangat penting bagi pembuat kebijakan, pengembang teknologi, dan pengguna untuk memprioritaskan pertimbangan etis dalam desain, pengembangan, dan penggunaan teknologi digital untuk menjunjung tinggi martabat manusia dan mendorong perilaku etis. Pada akhirnya, tantangan etika di dunia digital memerlukan refleksi, dialog, dan adaptasi yang berkelanjutan untuk memastikan bahwa teknologi memberikan manfaat bagi kebaikan bersama dan berkontribusi terhadap masyarakat yang lebih adil dan setara.

DAFTAR PUSTAKA

- Fernanda A. Ferreira, Conceição Castro, Cristina Lopes; Does corruption in the public sector reduce corporate ethics? – A panel data analysis. *AIP Conf. Proc.* 10 December 2019; 2186 (1): 090004. <https://doi.org/10.1063/1.5138000>
- Numa, N., Godoy, G., & Teixeira, J. (2024). At the frontier of the metaverse: NFTs, artistic expression, and digital immersions. *Metaverse*, 5(1), 2449. <https://doi.org/10.54517/m.v5i1.2449>
- Z. Kilhoffer et al. (2023). “How technical do you get? I’m an English teacher”: Teaching and Learning Cybersecurity and AI Ethics in High School. *IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2023, pp. 2032-2032, doi: [10.1109/SP46215.2023.10179333](https://doi.org/10.1109/SP46215.2023.10179333)
- Maureen E. Squires. (2020). *Ethics in Higher Education*. Nova Science Publishers.
- Silvia Rahmelia, Chris Apandie. (2023). Civic Virtue dalam Pendidikan Kristen guna Memperkuat Etika Digital di Era 4.0.
- Sabilar Rosyad, M. Alif. (2023). Hadis di Era Digital: Tantangan dan Peluang Penggunaan Teknologi dalam Studi Hadis. *Jurnal Ilmu Agama: Mengkaji Doktrin, Pemikiran, dan Fenomena Agama*. 24(2), 185–197. <https://doi.org/10.19109/jia.v24i2.18979>
- Sari, N. S. ., Ratnasari, R. T. ., Osman, I. ., & Rusanti, E. (2023). Materialism and Environmental Knowledge as a Mediator for Relationships between Religiosity and Ethical Consumption. *Jurnal Ekonomi Syariah Teori Dan Terapan*, 10(5), 467–481. <https://doi.org/10.20473/vol10iss20235pp467-481>

BAB 4

STRUKTUR DATA

Oleh Sri Rezeki Candra Nursari

4.1 Pendahuluan

Struktur data merupakan bagian dari ilmu pemrograman dasar yang bertujuan membuat struktur penyimpanan dan pengorganisasian data pada memori komputer sehingga dapat digunakan secara efisien. Pada struktur data merupakan metode untuk melakukan penyimpanan, pengaturan, pengelolaan data dalam komputer dan memudahkan manipulasi data agar dapat di akses dan diproses dengan efisien di sistem komputer.

4.2 Tipe Struktur Data

Struktur data terdiri dari beberapa tipe dan setiap tipe strukturnya digunakan untuk menyimpan data di dalam komputer, beberapa tipe dari struktur data adalah sebagai berikut : (Elliot B. Koffman, 2005), (Hubbard Jhon, R., 2000), (A. T. Berztiss, 2014)

1. Array
2. Stack
3. Queue
4. Tree
5. Linked List
6. Graph

7. Hash Table

4.2.1 Array

Array adalah variabel yang menyimpan sekumpulan data dengan tipe sama. Setiap elemen data menempati lokasi atau alamat memori yang berbeda dan biasanya disebut elemen *array*. Elemen *array* merupakan variabel dalam *array* dan dapat di akses melalui di dalam elemen. Variabel *array* dideklarasikan dengan menentukan tipe dan nama variabel serta jumlah lokasi penyimpanan yang Anda dibuat. Alokasi memori *array* global bersifat statis sedangkan alokasi memori *array* lokal bersifat dinamis.

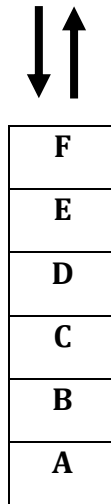
Contoh Array :

1. `int Angka[12]`
 - Nama *array* adalah Angka dengan jumlah elemen sebanyak 12 dan menggunakan *type* data *integer*.
2. `char Kata[22]`
 - Nama *array* adalah Kata dengan jumlah elemen sebanyak 22 dan menggunakan *type* data karakter.
3. `int Data[3][4]`
 - Nama *array* adalah Data dengan jumlah elemen baris sebanyak 3, jumlah elemen kolom sebanyak 4 dan menggunakan *type* data *integer*.

4.2.2 Stack

Stack adalah struktur data linier (SDL). *Stack* adalah konsep struktur data dengan sistem kerja. Data yang terakhir yang dimasukkan adalah data keluar pertama. Artinya data terakhir yang ditempatkan pada *stack* adalah data pertama yang diambil dari *stack*. *Stack* juga biasa disebut **tumpukan**. *Stack* adalah tempat penyimpanan yang mempunyai sifat *Last in First Out (LIFO)*. Proses *stack* data pertama masuk adalah A yang

selanjutnya data yang masuk B lalu C sampai dengan data yang masuk terakhir adalah F, pada saat data diambil/dikeluarkan maka data A menjadi paling akhir, sedangkan data pertama yang diambil/dikeluarkan adalah F. Proses dapat dilihat seperti dibawah ini :



Keunggulan dari *stack* adalah data yang diolah lebih efisien, membersihkan obyek secara otomatis dan mengontrol memori data dengan mandiri.

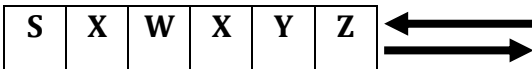
Kekurangan dari *stack* adalah memiliki kapasitas memori terbatas, mengalami *overflow* apabila obyek data banyak dan tidak dapat mengakses data secara acak.

Contoh *Stack* dalam kehidupan sehari-hari :

1. Tumpukan buku
2. Tumpukan coin
3. Tumpukan piring

4.2.3 Queue

Queue adalah struktur data linier (SDL). *Queue* merupakan sebuah konsep struktur data dalam sistem perilaku dimana data pertama masuk juga merupakan data pertama keluar. Artinya, data terakhir yang dimasukkan ke dalam *queue* adalah data terakhir yang dapat dikeluarkan dari *queue*, sedangkan data pertama kali dimasukkan ke dalam *queue* adalah data pertama yang dapat dikeluarkan dari quwu. *Queue* juga biasa disebut **antrian**. *Queue* adalah tempat penyimpanan yang mempunyai sifat *First in First Out (FIFO)*. Proses *queue* data pertama masuk adalah S yang selanjutnya data yang masuk X lalu W sampai dengan data yang masuk terakhir adalah Z, pada saat data diambil/dikeluarkan maka data Z menjadi paling akhir, sedangkan data pertama yang diambil/dikeluarkan adalah S. Proses dapat dilihat seperti dibawah ini :



Keunggulan dari *queue* adalah proses *queue*/antrian lebih cepat dan optimal dan dapat menangani beberapa tipe data sekaligus.

Kekurangan dari *queue* adalah memiliki kapasitas memori terbatas, memerlukan banyak memori apabila antrian besar.

Contoh *Queue* dalam kehidupan sehari-hari :

1. Sistem pelayanan di bank
2. Sistem pembayaran di kasir
3. Sistem mencetak dimana dokumen yang akan dicetak diurutkan dalam antrian untuk diproses.

4.2.4 Tree

Tree adalah struktur data nonlinier yang menggambarkan hubungan hierarki (hubungan *one to many*) antar elemen terutama pada proses penyimpanan data. *Tree* adalah suatu bentuk struktur data yang terdiri dari akar dan simpul dibawahnya. *Tree* dapat juga didefinisikan sebagai kumpulan simpul/*node/vertex* dengan elemen khusus yang disebut *ROOT*, dan *node* yang lainnya dibagi menjadi kumpulan *node* yang saling tidak berhubungan (disebut *SUBTREE*), yang dapat dituliskan sebagai :

$$T = (V, E)$$

dimana : $T = \text{tree}$; $V = \text{simpul/node/vertex}$; $E = \text{busur/arc/edge}$

Tree merupakan bagian dari *graph*, dengan simbol matematik :

$$T = \varepsilon G$$

dimana : $T = \text{tree}$; $G = \text{graph}$

Istilah-istilah umum dalam *tree* yaitu :

1. Root

Simpul/*node/vertex* yang tidak mempunyai superordinat. Merupakan simpul/*node/vertex* khusus dalam *tree* yang tidak memiliki *predecessor*.

2. Superordinat

Superordinat disebut juga *father/bapak/parent*. *Father/bapak/parent* merupakan *predecessor* satu level di atas suatu simpul/*node/vertex*.

3. Subordinat

Subordinat disebut juga *son/anak/child*. *Son/anak/child* merupakan *successor* satu level di bawah suatu simpul/*node/vertex*.

4. *Leaf/daun*

Seluruh simpul/*node/vertex* yang tidak memiliki subordinat. *Leaf/daun* merupakan simpul/*node/vertex* dalam *tree* yang tidak memiliki *successor*.

5. *Ancestor*

Semua simpul/*node/vertex* yang mendahului simpul/*node/vertex* tertentu dan berada pada jalur yang sama.

6. *Descendant*

Semua simpul/*node/vertex* setelah simpul/*node/vertex* tertentu dan berada pada lintasan yang sama.

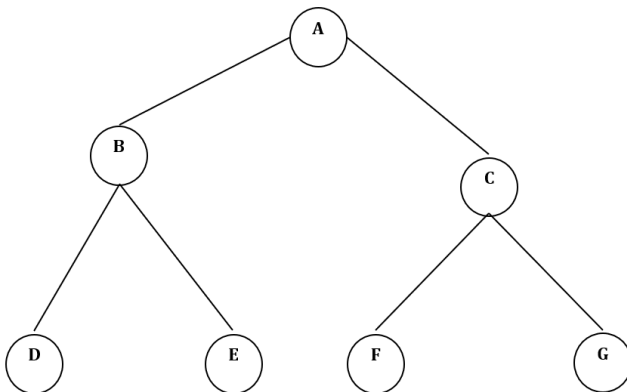
7. *Sibling*

Simpul/*node/vertex* yang memiliki *father/bapak/parent* yang sama dengan suatu simpul/*node/vertex*.

Keunggulan dari *tree* adalah proses pencarian data lebih cepat.

Kekurangan dari *tree* adalah membutuhkan waktu yang lama untuk proses memasukkan data karena harus sesuai dengan urutannya. (Storer, 2002)

Contoh *Tree* :



Ancestor (F) = A

Descendant (A) = F,G & D,E

Parent (D)	= B
Child (A)	= B,C
Sibling (F)	= G
Root	= A
Leaf	= D,E,F,G

4.2.5 Linked List

Linked list merupakan struktur data linier (SDL). *Linked list* adalah array dinamis. *Linked list* adalah sejumlah obyek yang *link* atau dihubungkan sehingga membentuk suatu *list*. *List* termasuk kedalam himpunan dinamis dan data *linked list* diakses secara manual. Tidak ada batasan untuk jumlah elemen *list*, hanya dibatasi secara fisik oleh memori yang tersedia. Setiap elemen dapat memiliki tipe data tersendiri yang berbeda dengan tipe data elemen lainnya. Sebuah *list* mempunyai dua bagian, yaitu

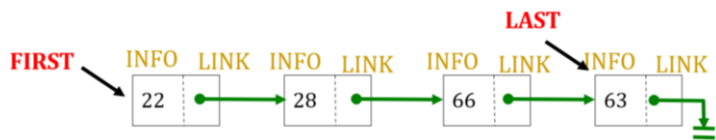
1. *Head/Kepala*, yang digunakan untuk menyimpan alamat elemen pertama atau elemen terakhir dari sebuah *list*.
2. *Badan list*, yang berisi kumpulan-kumpulan elemen *list*.

Struktur *linked list* terbagi menjadi empat macam, yaitu

1. *Singly Linked List*

Merupakan *linked list* lurus dengan pointer tunggal, dimana dalam struktur simpulnya hanya ada satu elemen yang bertipe pointer yang berisi alamat simpul berikutnya atau *next node*.

Contoh :

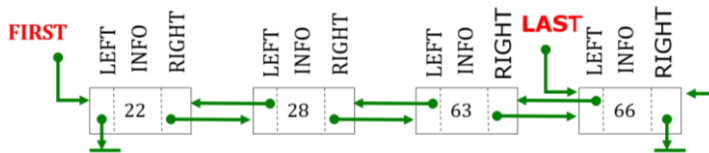


- Terdapat empat simpul.
- Setiap simpulnya terdiri dari dua elemen, yaitu info dan link.
- Elemen info bertipe integer yang menyimpan nilai 22, 28, 66 dan 63.
- Elemen link bertipe pointer yang menyimpan alamat simpul.
- Simpul pertama ditunjuk oleh pointer *FIRST*.
- Simpul terakhir ditunjuk oleh pointer *LAST*.

2. Doubly Linked List

Merupakan *doubly linked list* lurus dengan pointer ganda, dimana dalam struktur simpulnya ada dua elemen yang bertipe pointer yang berisi alamat simpul sebelumnya dan kedua menunjuk simpul berikutnya atau *next node*.

Contoh :

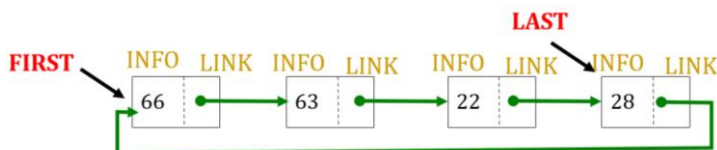


- Setiap simpulnya terdiri dari tiga elemen, yaitu *left*, info dan *right*.
- Elemen info bertipe integer yang menyimpan nilai 22, 28, 63 dan 66.

3. Circular Linked List

Merupakan *linked list* dimana *link* simpul terakhir tidak berisi *NULL*, melainkan terisi dengan alamat simpul pertama, sehingga menciptakan efek melingkar sesuai arah jarum jam.

Contoh :

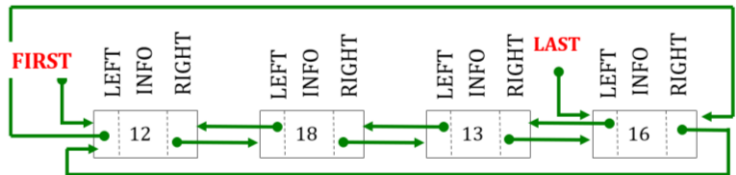


- Setiap simpulnya terdiri dari dua elemen, yaitu info dan *link*.
- Elemen info bertipe integer yang menyimpan nilai 66, 63, 22 dan 28.
- Elemen *link* bertipe pointer yang menyimpan alamat simpul.

4. Circular Doubly Linked List

Merupakan *doubly linked list* dimana pointer *RIGHT* simpul paling kanan berisi alamat simpul paling kiri, dan pointer *LEFT* simpul paling kiri berisi alamat simpul paling kanan, sehingga menciptakan efek melingkar baik menurut arah jarum jam maupun arah kebalikannya.

Contoh :



- Terdapat empat simpul.
- Elemen info bertipe integer yang menyimpan nilai 12, 18, 13 dan 16.

Keunggulan dari *linked list* adalah ukuran lebih dinamis, penambahan dan pengurangan data prosesnya lebih cepat serta alokasi memori dapat disesuaikan kebutuhan.

Kekurangan dari *linked list* adalah membutuhkan waktu yang lama untuk proses *transversal* lebih panjang karena tidak secara langsung diakses.

4.2.6 Graph

Graph merupakan konsep struktur data non linear yang terdiri simpul dan busur serta merupakan gambaran bentuk hubungan antar elemen secara sejajar tidak berbentuk hierarki.

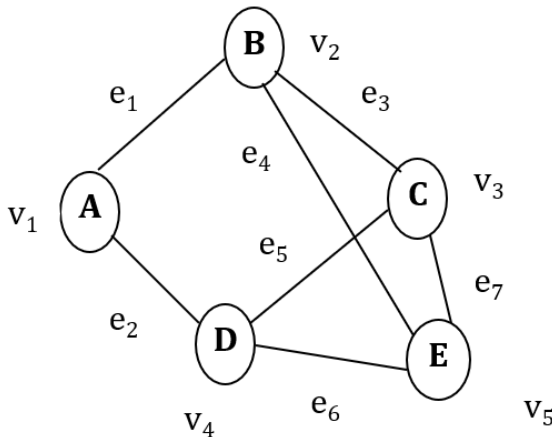
Relasi antar elemennya berupa *many-to-many*, secara matematis dapat dinyatakan sebagai berikut :

$$G = (V, E)$$

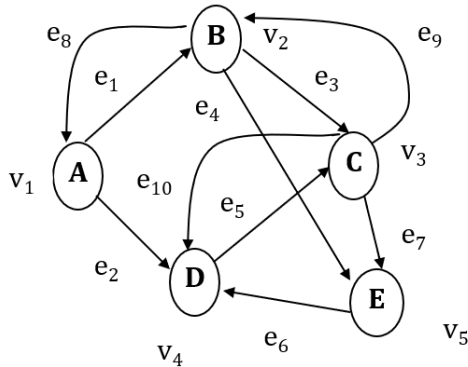
dimana : $G = \text{graph}$; $V = \text{simpul/vertex}$; $E = \text{busur/edge}$

Graph dibedakan menjadi dua macam :

1. *Graph* tak berarah (*undirect graph/non-direct graph*)
Diidentikkan dengan sebuah jalan yang menghubungkan dua buah simpul/node/titik, yang dapat dilalui secara dua arah. Urutan simpul pada busur e_1 , dapat disebut busur A,B atau B,A. Contoh *Graph* Tak Berarah :



2. *Graph* berarah (*direct graph/ digraph*)
Sebuah jalan yang menghubungkan dua buah simpul/node/titik. Diidentikkan apabila A ke B dapat menggunakan busur e_1 , sedangkan dari B ke A hanya menggunakan busur e_8 . Contoh *Graph* Berarah :



Keunggulan dari *graph* adalah proses memeriksa hubungan node satu dengan yang lainnya lebih cepat . (Dr. Prabhakar Gupta, Vineet Agarwal, 2007), (A. T. Berztiss, 2014)

4.2.7 Hash Table

Hash table digunakan untuk menyimpan data secara asosiatif dan dalam bentuk format array, sehingga dapat diakses datanya dengan sangat cepat karena menggunakan indeks. Operasi yang sering dilakukan pada tipe ini adalah menyisipkan elemen, mencari elemen dan menghapus elemen. (Gakhov, 2022)

4.3 Fungsi Utama Struktur Data

Fungsi utama struktur data, agar dapat membuat program dengan baik antara lain :

1. Menyimpan dan mengatur data
2. Membuat proses pemrograman yang lebih mudah
3. Meningkatkan kualitas algoritma
4. Mengelola sumber dan layanan data

4.4 Jenis Struktur Data

Data raya biasa juga disebut dengan *big data* merupakan kumpulan data yang besar dan memiliki jenis data yang beragam. Salah satu karakteristik pada *big data* adalah *variety* (variasi). Variety terjadi karena adanya data yang beragam, mulai dari format maupun struktur dari isi datanya. Jenis data yang digunakan terdiri dari : (Dileep Kumar G., 2016)

1. Data Terstruktur
2. Data Semi-Terstruktur
3. Data Tidak Terstruktur

4.4.1 Data Terstruktur

Data terstruktur merupakan data yang tersusun dengan baik dan terdefiniskan. Data disimpan, dianalisis dan diproses dengan format yang sama.

Contoh data struktur :

NO	NAMA_LANGGANAN	NO_LANGGANAN	KELOMPOK	BULAN_INI	BULAN_LALU	PEMAKAIAN	TARIF	TAGIHAN_KOTOR	METERAI	TAGIHAN_BERSIH
1	Bagus Krishna	SS.A1	Sosial	1000	995	5	250	2500	0	2750
2	Rizan Cahyatama	RT.A1	Rumah Tangga	4250	1235	3015	500	1128125	3000	1243938
3	Salsabela	NI.A1	Niaga	1500	1077	423	1000	312250	0	343475
4	Yulia Niken	IN.A1	Industri	5000	3965	1035	2000	1542500	3000	1699750
5	Dwi Kartika	SS.A2	Sosial	2500	2456	44	300	8400	0	9240
6	Astri	RT.A2	Rumah Tangga	3000	2699	301	550	121143	0	133554
7	Syah Susilawati	NI.A2	Niaga	3500	2444	1056	1250	983750	0	1082125
8	Erfan Nurcahya	IN.A2	Industri	1250	800	450	2250	748125	0	822938
9	Eka	SS.A3	Sosial	5500	5000	500	350	129500	0	142450
10	Jalilani	RT.A3	Rumah Tangga	7854	7150	704	600	313800	0	345180
11	Joko Pitoyo	NI.A3	Niaga	11250	4725	6525	1500	7333125	6000	8072438
12	Sari	IN.A3	Industri	4250	2952	1298	2500	2421250	3000	2666375

4.4.2 Data Semi Terstruktur

Data semi terstruktur merupakan data yang belum diklasifikasikan tetapi mengandung informasi penting yang memisahkan elemen individu dalam data.

Contoh data semi terstruktur :


```

<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/srcandra_n/ontologies/2023/11/SourceCervical073/"
  xml:base="http://www.semanticweb.org/srcandra_n/ontologies/2023/11/SourceCervical073/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xm1="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:SourceCervical073="http://www.semanticweb.org/srcandra_n/ontologies/2023/11/SourceCervical073#">
  <owl:Ontology rdf:about="http://www.semanticweb.org/srcandra_n/ontologies/2023/11/SourceCervical073">
    <rdfs:comment xml:lang="en">Source : This is an ontology cervical cancer.</rdfs:comment>
  </owl:Ontology>

  <!--
  //
  // Object Properties
  //
  -->

```

Data hasil ontology dalam format .XML. Sumber:
https://ichantix.com/serviks/index.php?page=hasil_ontology

4.2.3 Data Tidak Terstruktur

Data tidak terstruktur merupakan data tidak memiliki bentuk atau struktur khusus.

Contoh data tidak terstruktur :



Contoh data tidak terstruktur berupa foto. Sumber:
<https://pixabay.com>

DAFTAR PUSTAKA

- A. T. Berztiss (2014) *Data Structure Theory and Practice*. Elsevier Science.
- Dileep Kumar G., M. K. S. (2016) *Effective Big Data Management and Opportunities for Implementation*. IGI Global.
- Dr. Prabhakar Gupta, Vineet Agarwal, M. V. (2007) *Data Structure Using C*. Firewall Media.
- Elliot B. Koffman, P. A. T. W. (2005) *Objects, Abstraction, Data Structures and Design Using Java*. John Wiley & Sons.Inc.
- Gakhov, A. (2022) *Probabilistic Data Structures and Algorithms for Big Data Applications*. Books on Demand.
- Hubbard Jhon, R., P. . (2000) *Schaum's Outline Of Theory and Problems of Data Structures With C++*. Available at: McGraw-Hill.
- Storer, J. A. (2002) *An introduction to data structures and algorithms*. Birkhäuser Boston.

BAB 5

JENIS DATA

Oleh Salaki Reynaldo Joshua

5.1 Pendahuluan

Tipe data adalah konsep dasar dalam pemrograman komputer yang menentukan sifat data dan cara memanipulasinya. Pada intinya, tipe data menentukan jenis nilai yang dapat ditampung oleh variabel, seperti angka, karakter, atau entitas yang lebih kompleks seperti daftar atau objek kustom. Memahami tipe data sangat penting karena menentukan bagaimana data disimpan dalam memori, operasi apa yang dapat dilakukan pada data, dan bagaimana data berinteraksi dengan bagian lain dari suatu program (Sebesta, 2015).

Pada masa-masa awal komputasi, data terutama direpresentasikan sebagai digit biner (0 dan 1) dalam memori. Seiring berkembangnya bahasa pemrograman, mereka memperkenalkan abstraksi pada representasi biner ini untuk memudahkan pemrogram bekerja dengan data. Abstraksi inilah yang sekarang kita sebut tipe data. Mereka menyediakan cara untuk menafsirkan data biner dengan cara yang bermakna, memungkinkan pemrogram untuk melakukan operasi yang masuk akal untuk jenis data tersebut (Cormen et al., 2009).

\$\$

$\text{Data Type} = \{\text{Domain}, \text{Operations}, \text{Representation}\}$

\$\$

Persamaan di atas, terinspirasi dari definisi formal dalam “Struktur Data dan Algoritma” oleh Aho et al. (1983), merangkum tiga aspek kunci dari tipe data:

- 1. Domain: Kumpulan nilai yang dapat diambil oleh tipe data.
- 2. Operasi: Kumpulan fungsi yang dapat diterapkan pada nilai jenis ini.
- 3. Representasi: Bagaimana data disimpan dalam memori komputer.

Misalnya, tipe data bilangan bulat mungkin memiliki domain bilangan bulat dari -2^{31} hingga $2^{31}-1$ (dalam sistem 32-bit), operasi seperti penjumlahan dan perkalian, dan representasi biner menggunakan komplemen dua.

Sama seperti Anda tidak akan menggunakan gergaji untuk memalu paku atau kunci pas untuk mengebor lubang, Anda tidak akan menggunakan tipe data string untuk melakukan aritmatika atau bilangan bulat untuk menyimpan teks. Setiap alat (tipe data) dirancang untuk tugas tertentu (operasi) pada materi tertentu (domain). Menggunakan alat yang tepat membuat pekerjaan Anda efisien dan mencegah kesalahan. Untuk lebih memahami ragam tipe data, perhatikan tabel 5.1.

Tabel 5. 1. Different data types are suited for different kinds of data and operations

Data Type	Example Values	Common Operations	Typical Use Cases
Integer	-5, 0, 42	+, -, *, /, %	Counting, indexing
Float	3.14, -0.001	+, -, *, /, sqrt	Scientific calculations

Data Type	Example Values	Common Operations	Typical Use Cases
Boolean	true, false	AND, OR, NOT	Control flow, flags
Character	'a', '5', '\$'	comparison	Text processing
String	"Hello"	concatenation, substring	User input, file names
Array	[1, 2, 3]	indexing, sorting	Lists, matrices

Sumber: Robert C. Martin, 2008

Tabel 5.1 diadaptasi dari "Clean Code" oleh Robert C. Martin (2008), menunjukkan bagaimana tipe data yang berbeda cocok untuk jenis data dan operasi yang berbeda.

Kesimpulannya, tipe data adalah blok penyusun pemrograman. Mereka menyediakan struktur pada data, memungkinkan penggunaan memori yang efisien, memfasilitasi pemeriksaan kesalahan, dan membuat kode lebih mudah dibaca dan dipelihara. Saat Anda mempelajari pemrograman lebih dalam, Anda akan menemukan bahwa menguasai tipe data adalah kunci untuk menulis kode yang efisien, bebas bug, dan ekspresif.

5.2 Tipe Data Primitif

Tipe data primitif, juga dikenal sebagai tipe data dasar atau bawaan, adalah jenis data paling sederhana yang disediakan oleh bahasa pemrograman. Tipe ini didukung langsung oleh bahasa dan berfungsi sebagai blok penyusun untuk struktur data yang lebih kompleks. Tipe primitif biasanya diimplementasikan langsung oleh perangkat keras, membuat pengoperasiannya cepat dan efisien (Stroustrup, 2013).

Konsep tipe primitif sudah ada sejak awal komputasi. Dalam makalahnya yang berjudul "Fungsi Rekursif Ekspresi Simbolik dan Perhitungannya oleh Mesin, Bagian I," John McCarthy (1960) memperkenalkan gagasan simbol atom dalam bahasa pemrograman LISP, yang dapat dianggap sebagai pendahulu tipe primitif modern.

Kebanyakan bahasa pemrograman mendukung empat kategori utama tipe primitif:

1. Integer (int): Mewakili bilangan bulat, positif atau negatif, tanpa bagian pecahan apa pun.
2. Floating-point (float, double): Mewakili bilangan real dengan bagian pecahan.
3. Karakter (char): Mewakili satu karakter Unicode.
4. Boolean (bool): Mewakili nilai kebenaran, benar atau salah.

Jangkauan dan ketepatan sebenarnya dari jenis ini dapat bervariasi tergantung pada bahasa pemrograman dan perangkat keras yang mendasarinya. Misalnya, di C++ (Gambar 5.1).

```
int a = -42;           // Integer
float b = 3.14f;       // Single-precision float
double c = 2.71828;    // Double-precision float
char d = 'A';          // Character
bool e = true;         // Boolean
```

Gambar 5. 1. Contoh dalam C++

Penyimpanan dan representasi tipe-tipe ini dalam memori sangat menarik. Perhatikan diagram berikut (Gambar 5.2)

```

Memory (32-bit system)

int (4 bytes):
+---+---+---+---+
|00 |00 |00 |2A | = 42 (decimal)
+---+---+---+---+

float (4 bytes):
+---+---+---+---+
|40 |48 |F5 |C3 | ≈ 3.14 (IEEE 754)
+---+---+---+---+

char (typically 1 byte):
+---+
|41 | = 'A' (ASCII)
+---+

bool (often 1 byte, but varies):
+---+
|01 | = true
+---+

```

Gambar 5. 2. Diagram

Diagram ini (Gambar 5.2), terinspirasi oleh diskusi dalam "Sistem Komputer: Perspektif Pemrogram" (Bryant & O'Hallaron, 2015), menunjukkan bagaimana tipe ini biasanya disimpan dalam memori. Integer dan float sering kali menggunakan 4 byte (32 bit), memungkinkan rentang nilai yang luas. Float menggunakan standar IEEE 754 untuk aritmatika floating-point biner, yang mencakup bit untuk tanda, eksponen, dan mantissa.

Pentingnya memahami representasi ini menjadi jelas ketika menangani masalah seperti integer overflow atau presisi floating-point. Contohnya (Gambar 5.3)

```
int x = 2147483647; // Largest 32-bit signed int
x = x + 1;          // Overflow: x becomes -2147483648

float y = 0.1f;
float z = y + y + y;
assert(z == 0.3f); // Assertion fails due to floating-point imprecision
```

Gambar 5. 3. Integer Overflow

Contoh-contoh ini, diadaptasi dari "C++ Efektif" (Meyers, 2005), menyoroti kesalahan umum pada tipe primitif.

Untuk meringkas karakteristik dan kasus penggunaan tipe primitif, perhatikan tabel 5.2.

Tabel 5. 2. The Characteristics and Use Cases of Primitive Types

Data Type	Range (typical)	Precision	Use Cases
int	-2^{31} to $2^{31}-1$	Perfect	Counting, indexing, bit operations
float	$\pm 1.18\text{e-}38$ to $\pm 3.4\text{e}38$	~7 digits	Scientific calculations, graphics
double	$\pm 2.23\text{e-}308$ to $\pm 1.8\text{e}308$	~15 digits	High-precision calculations
char	0 to 255 (ASCII)	N/A	Text processing, encryption
bool	true, false	N/A	Control flow, flags, logical operations

Untuk meringkas karakteristik dan kasus penggunaan tipe primitif, perhatikan tabel 5.2.

Tabel 5.2 ini, terinspirasi oleh "Bahasa Pemrograman C++" (Stroustrup, 2013), menunjukkan mengapa penting untuk memilih tipe yang tepat untuk data Anda. Menggunakan int untuk nilai yang memerlukan presisi tinggi, atau double untuk tanda sederhana, dapat menyebabkan penggunaan memori tidak efisien atau bug halus.

Kesimpulannya, tipe data primitif adalah atom pemrograman. Mereka menyediakan alat dasar untuk merepresentasikan angka, karakter, dan nilai kebenaran. Memahami kemampuan dan keterbatasannya sangat penting untuk menulis kode yang benar dan efisien. Seperti yang dikatakan Donald Knuth, penulis "The Art of Computer Programming," yang terkenal, "Teorema dasar matematika tidak selalu benar dalam aritmatika komputer" (Knuth, 1997). Kebijakan ini berlaku langsung pada nuansa bekerja dengan tipe primitif.

5.3 Tipe Data Komposit

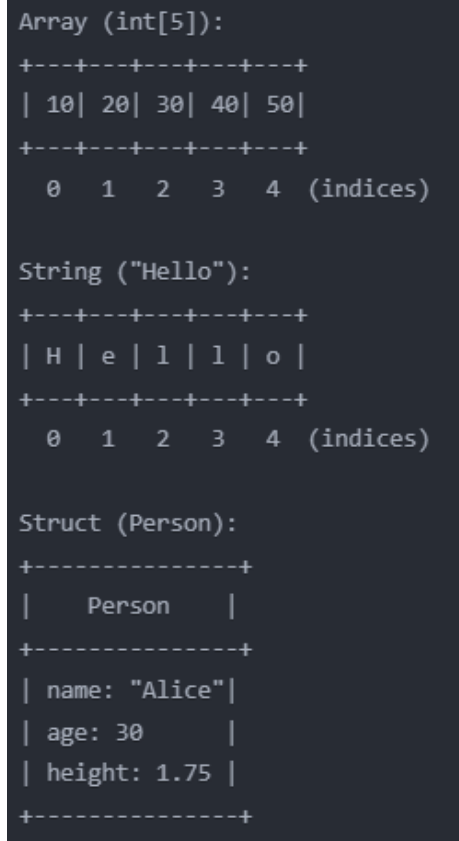
Tipe data komposit, juga dikenal sebagai tipe gabungan atau terstruktur, adalah tipe data yang terdiri dari beberapa tipe primitif atau tipe komposit lainnya. Mereka menyediakan cara untuk mengelompokkan item data terkait bersama-sama, memungkinkan pemrogram untuk mewakili entitas yang lebih kompleks daripada yang dimungkinkan oleh tipe primitif. Konsep tipe komposit sangat penting dalam pemodelan objek dunia nyata dan hubungan dalam suatu program (Booch et al., 2007).

Ide tipe komposit berakar pada karya awal abstraksi data. Dalam makalah mereka "Pemrograman dengan Tipe Data Abstrak," Liskov dan Zilles (1974) berpendapat pentingnya menciptakan tipe yang ditentukan pengguna yang merangkum data dan operasi pada data tersebut. Ini meletakkan dasar bagi tipe komposit modern.

Tiga tipe komposit mendasar yang ditemukan di sebagian besar bahasa pemrograman adalah:

1. Array: Kumpulan elemen bertipe sama yang diurutkan, diakses berdasarkan indeks.
2. String: Urutan karakter, seringkali dengan operasi khusus untuk manipulasi teks.
3. Struktur (struct): Kumpulan item data terkait, mungkin tipenya berbeda.

Mari kita visualisasikan tipe-tipe ini (Gambar 5.4).



Gambar 5. 4. Visualisasi Tipe Komposit

Diagram (Gambar 5.4) ini, terinspirasi oleh "Struktur Data dan Analisis Algoritma dalam C++" (Weiss, 2013), menunjukkan bagaimana tipe komposit mengatur data. Array dan string

menggunakan lokasi memori yang berdekatan, memungkinkan akses acak cepat berdasarkan indeks. Struktur mengelompokkan item-item terkait, bahkan dari tipe yang berbeda, ke dalam satu unit.

Array merupakan hal mendasar dalam banyak algoritma. Misalnya, pertimbangkan pencarian biner (Gambar 5.5)

```
int binarySearch(int arr[], int n, int x) {  
    int left = 0, right = n - 1;  
    while (left <= right) {  
        int mid = left + (right - left) / 2;  
        if (arr[mid] == x) return mid;  
        if (arr[mid] < x) left = mid + 1;  
        else right = mid - 1;  
    }  
    return -1;  
}
```

Gambar 5. 5. Pencarian Biner

Algoritma ini, dijelaskan dalam "Pengantar Algoritma" (Cormen et al., 2009), secara efisien menemukan elemen dalam array yang diurutkan dengan membagi ruang pencarian menjadi dua berulang kali. Kompleksitas waktunya adalah $O(\log n)$, jauh lebih baik daripada $O(n)$ pencarian linier.

String, meskipun secara konseptual mirip dengan array, sering kali memiliki operasi khusus. Algoritme klasik Knuth-Morris-Pratt (KMP) untuk pencocokan pola dalam string menggambarkan hal ini (Gambar 5.6).

```
void computeLPS(string pat, int lps[]) {
    int len = 0, i = 1;
    lps[0] = 0;
    while (i < pat.length()) {
        if (pat[i] == pat[len]) {
            lps[i++] = ++len;
        } else {
            if (len != 0) len = lps[len - 1];
            else lps[i++] = 0;
        }
    }
}
```

Gambar 5. 6. Pencocokan Pola dalam String

Fungsi ini, bagian dari algoritma KMP seperti yang dijelaskan dalam "Algoritma String" (Crochemore & Lecroq, 2004), memproses string terlebih dahulu untuk mengaktifkan pencarian substring yang efisien, tugas umum dalam pemrosesan teks dan bioinformatika.

Struktur (atau kelas dalam bahasa berorientasi objek) sangat berguna untuk pemodelan. Pertimbangkan permainan sederhana (Gambar 5.7).

```
struct Player {
    string name;
    int health;
    int score;
};

Player players[MAX_PLAYERS];
players[0] = {"Mario", 100, 0};
players[1] = {"Luigi", 100, 0};
```

Gambar 5. 7. Contoh Permainan Sederhana

Kode ini, mengingatkan pada contoh dalam "Game Engine Architecture" (Gregory, 2018), menunjukkan bagaimana struct dapat memodelkan entitas game, menggabungkan berbagai jenis data ke dalam unit yang kohesif.

Untuk meringkas karakteristik dan kasus penggunaan jenis ini (Tabel 5.3).

Tabel 5. 3. The Characteristics and Use Cases of Primitive Types

Type	Key Features	Common Operations	Use Cases
Arrays	Fixed size, random access	Indexing, sorting, searching	Lists, matrices, lookup tables
Strings	Variable length, null-terminated (C)	Concatenation , substring, pattern matching	Text processing, user input, parsing
Structs	Heterogeneous data, member access	Field access, copying	Game entities, database records, GUI components

Tabel ini (Tabel 5.3), terinspirasi oleh diskusi dalam "Pemrograman: Prinsip dan Praktek Menggunakan C++" (Stroustrup, 2014), menyoroti bagaimana fitur setiap jenis membuatnya cocok untuk tugas yang berbeda.

Kesimpulannya, tipe data komposit adalah alat penting untuk menangani data kompleks dalam pemrograman. Array menyediakan penyimpanan dan akses yang efisien untuk data homogen, string berspesialisasi dalam manipulasi teks, dan struktur memungkinkan pemodelan entitas heterogen. Seperti yang diungkapkan Niklaus Wirth, perancang Pascal, dalam bukunya "Algorithms + Data Structures = Programs" (Wirth,

1976), cara kita menyusun data sangat memengaruhi algoritme yang dapat kita gunakan dan masalah yang dapat kita selesaikan secara efisien. Oleh karena itu, menguasai tipe komposit merupakan langkah kunci untuk menjadi seorang programmer yang mahir.

5.4 Tipe Data Komposit

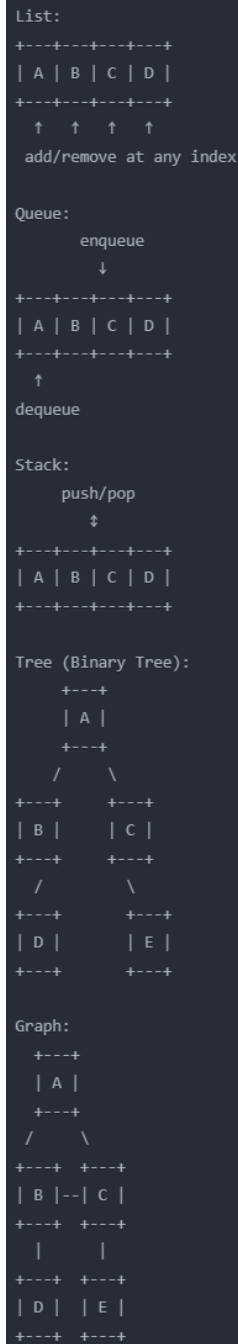
Tipe Data Abstrak (ADT) mewakili lompatan dalam evolusi tipe data. Tidak seperti tipe primitif dan komposit yang berfokus pada penyimpanan data, ADT merangkum data dan operasi yang dapat dilakukan pada data tersebut. Mereka memberikan tingkat abstraksi yang memisahkan apa (antarmuka) dari bagaimana (implementasi), memungkinkan pemrogram untuk menggunakan struktur data yang kompleks tanpa mengkhawatirkan cara kerja internal mereka (Aho et al., 1983).

Konsep ADT/*Abstract Data Types* diresmikan pada akhir tahun 1970an. Dalam makalah penting mereka "Tentang Tipe Data Abstrak," Guttag dan Horning (1978) mendefinisikan ADT sebagai "kelas objek abstrak yang sepenuhnya dicirikan oleh operasi yang tersedia pada objek tersebut." Definisi ini menekankan bahwa esensi ADT terletak pada perilakunya, bukan representasinya.

ADT umum meliputi:

1. Lists: Koleksi terurut yang memungkinkan penyisipan dan penghapusan di posisi mana pun.
2. Queues:: Koleksi yang mengikuti disiplin First-In-First-Out (FIFO).
3. Stacks: Koleksi yang mengikuti disiplin Last-In-First-Out (LIFO).
4. Trees: Struktur hierarki dengan akar dan simpul anak.
5. Graphs : Kumpulan simpul (node) dan sisi yang menghubungkannya.

Mari kita visualisasikan ADT ini (Gambar 5.7).



Gambar 5. 8. Visualisasi ADT

Diagram (Gambar 5.7) ini, terinspirasi oleh "Struktur Data dan Algoritma di Java" (Goodrich et al., 2014), menggambarkan organisasi konseptual ADT ini. Perhatikan bahwa implementasi sebenarnya (array, struktur tertaut, dll.) disembunyikan dari pengguna.

Kekuatan ADT terlihat jelas dalam penerapannya. Pertimbangkan tombol kembali browser web, yang dapat diimplementasikan menggunakan tumpukan (Gambar 5.9).

```
class BrowserHistory:
    def __init__(self):
        self.history = [] # stack of URLs

    def visit(self, url):
        self.history.append(url)

    def back(self):
        if len(self.history) > 1:
            self.history.pop()
            return self.history[-1]
        return None # can't go back further
```

Gambar 5. 9. Web browser's back button

Kode ini, mengingatkan pada contoh dalam "Cracking the Coding Interview" (McDowell, 2015), menunjukkan bagaimana perilaku LIFO tumpukan secara alami memodelkan riwayat browser.

Demikian pula, pertimbangkan algoritma Dijkstra untuk menemukan jalur terpendek dalam grafik (Gambar 5.10).


```
import heapq

def dijkstra(graph, start):
    distances = {node: float('inf') for node in graph}
    distances[start] = 0
    pq = [(0, start)] # priority queue of (distance, node)

    while pq:
        current_distance, current_node = heapq.heappop(pq)
        if current_distance > distances[current_node]:
            continue

        for neighbor, weight in graph[current_node].items():
            distance = current_distance + weight
            if distance < distances[neighbor]:
                distances[neighbor] = distance
                heapq.heappush(pq, (distance, neighbor))

    return distances
```

Gambar 5. 10. Algoritma Dijkstra

Implementasi ini, berdasarkan diskusi dalam "Algoritma" (Dasgupta et al., 2008), menggunakan antrian prioritas (sering diimplementasikan sebagai heap) untuk menemukan node terdekat berikutnya secara efisien. Grafik ADT mengabstraksikan detail tentang bagaimana node dan edge disimpan.

Untuk meringkas karakteristik dan kasus penggunaan ADT ini (Tabel 5.4).

Tabel 5. 4. The Characteristics and Use Cases of These ADTs

ADT	Key Operations	Typical Implementations	Use Cases
List	insert, remove, get, set	Array, Linked List	Dynamic data, editor buffers
Queue	enqueue, dequeue	Array, Linked List	Task scheduling, message passing
Stack	push, pop	Array, Linked List	Function calls, expression

ADT	Key Operations	Typical Implementations	Use Cases
			evaluation
Tree	insert, remove, search	Linked Structure, Array	File systems, decision trees
Graph	addEdge, removeEdge, findPath	Adjacency List, Matrix	Social networks, routing algorithms

Tabel (Tabel 5.4) ini, terinspirasi oleh "Pengantar Algoritma" (Cormen et al., 2009), menunjukkan bagaimana operasi mendefinisikan ADT, sementara implementasi yang berbeda dapat menyediakan operasi ini.

Kesimpulannya, Tipe Data Abstrak adalah alat yang ampuh dalam gudang senjata programmer. Mereka memungkinkan kita untuk berpikir pada tingkat abstraksi yang lebih tinggi, dengan fokus pada apa yang dapat dilakukan oleh data, bukan pada bagaimana data disimpan. Pemisahan masalah ini menghasilkan kode yang lebih modular, dapat dipelihara, dan dapat digunakan kembali. Seperti yang dinyatakan oleh Barbara Liskov, yang memenangkan Turing Award atas karyanya pada abstraksi data, "Ide kuncinya adalah memisahkan spesifikasi perilaku suatu objek dari implementasinya" (Liskov, 1991). Dengan menguasai ADT, programmer dapat menulis kode yang tidak hanya benar dan efisien tetapi juga jelas dan mudah beradaptasi.

5.5 Tipe Data yang Didefinisikan oleh Pengguna

Tipe data yang ditentukan pengguna mewakili kemajuan signifikan dalam desain bahasa pemrograman. Mereka mengizinkan pemrogram untuk memperluas sistem tipe bahasa dengan mendefinisikan tipenya sendiri, disesuaikan dengan domain masalah spesifiknya. Kemampuan ini sangat penting untuk mengelola kompleksitas dalam sistem perangkat lunak yang besar dan untuk menciptakan bahasa domain spesifik dalam bahasa tujuan umum (Meyer, 1997).

Konsep tipe yang ditentukan pengguna muncul dengan munculnya pemrograman modular dan desain berorientasi objek. Dalam makalahnya yang berpengaruh "The Programming Language Pascal," Niklaus Wirth (1971) memperkenalkan kemampuan untuk mendefinisikan tipe kustom, dengan alasan bahwa "Eksplorasi struktur adalah inti dari pemrograman terstruktur." Ide ini dikembangkan lebih lanjut dalam bahasa berorientasi objek, di mana tipe (kelas) yang ditentukan pengguna adalah landasannya.

Bentuk umum dari tipe yang ditentukan pengguna meliputi:

1. Pencacahan (enum): Mendefinisikan tipe dengan sekumpulan konstanta bernama tetap.
2. Kelas: Menggabungkan data (atribut) dengan operasi (metode) dalam pemrograman berorientasi objek.

Mari kita visualisasikan tipe-tipe ini (Gambar 5.11).

```
Enumeration (TrafficLight):
+-----+
|  TrafficLight  |
+-----+
| RED    = 0     |
| YELLOW = 1     |
| GREEN  = 2     |
+-----+

Class (BankAccount):
+-----+
|  BankAccount  |
+-----+
| - accountNumber |
| - balance       |
| - owner         |
+-----+
| + deposit()     |
| + withdraw()    |
| + getBalance()  |
+-----+
```

Gambar 5. 11. Tipe Yang Ditentukan Pengguna

Diagram enumerasi, yang terinspirasi oleh "C++ Primer" (Lippman et al., 2012), menunjukkan bagaimana enum menyediakan cara untuk bekerja dengan sekumpulan konstanta terkait. Hal ini membuat kode lebih mudah dibaca dan aman untuk diketik dibandingkan menggunakan bilangan bulat atau string mentah.

Diagram kelas, menggunakan notasi UML seperti yang dijelaskan dalam "UML Distilled" (Fowler, 2003), menunjukkan bagaimana kelas merangkul data (accountNumber, balance, owner) dan operasi (deposit, penarikan, getBalance). Tanda - dan + masing-masing menunjukkan anggota swasta dan publik.

Pencacahan sangat berguna pada state machines. Pertimbangkan mesin penjual otomatis sederhana (Gambar 5.10

```
enum class VendingState { IDLE, COIN_INSERTED, DISPENSING, OUT_OF_STOCK };

class VendingMachine {
    VendingState state = VendingState::IDLE;
public:
    void insertCoin() {
        if (state == VendingState::IDLE)
            state = VendingState::COIN_INSERTED;
    }
    void dispense() {
        if (state == VendingState::COIN_INSERTED)
            state = VendingState::DISPENSING;
    }
    // other methods...
};
```

Gambar 5. 12. Vending Machine

Kode ini, mengingatkan pada contoh dalam "Pola Desain" (Gamma et al., 1994), menggunakan enum untuk membuat transisi keadaan menjadi eksplisit dan mengurangi kemungkinan kesalahan dibandingkan dengan menggunakan angka atau string ajaib.

Kelas bersinar dalam memodelkan entitas dunia nyata. Pertimbangkan sistem perpustakaan (Gambar 5.13).

```
class Book:
    def __init__(self, title, author, isbn):
        self.title = title
        self.author = author
        self.isbn = isbn
        self.is_checked_out = False

    def checkout(self):
        if not self.is_checked_out:
            self.is_checked_out = True
            return True
        return False

    def return_book(self):
        self.is_checked_out = False

class Library:
    def __init__(self):
        self.books = []

    def add_book(self, book):
        self.books.append(book)

    def find_book_by_isbn(self, isbn):
        return next((book for book in self.books if book.isbn == isbn), None)

    # other methods...
```

Gambar 5. 13. Sistem Perpustakaan

Desain (Gambar 5.13) ini, terinspirasi oleh “Clean Code” (Martin, 2008), menunjukkan bagaimana kelas (Buku dan Perpustakaan) dapat memodelkan sistem dengan cara yang mendekati cara kita memikirkannya di dunia nyata. Kelas Library mendemonstrasikan komposisi, berisi daftar objek Buku.

Untuk meringkas karakteristik dan kasus penggunaan tipe yang ditentukan pengguna ini (Tabel 5.5).

Tabel 5. 5. The Characteristics and Use Cases of These User-Defined Types

Type	Key Features	Common Use Cases	Examples
Enum	Named constants,	State machines,	{RED, GREEN, BLUE}, {MONDAY, ...,

Type	Key Features	Common Use Cases	Examples
	type safety	configuration options	SUNDAY}
Class	Data + Operations, Encapsulation	Modeling real-world entities, Creating APIs	BankAccount, Player, Document

Tabel (Tabel 5.5) ini, yang mengambil ide dari "Jawa Efektif" (Bloch, 2018), menyoroti bagaimana enum memberikan kejelasan dan keamanan untuk sekumpulan nilai yang tetap, sementara kelas menawarkan cara yang ampuh untuk memodelkan entitas dan perilaku yang kompleks.

Tipe yang ditentukan pengguna juga mendukung fitur lanjutan seperti pewarisan, polimorfisme, dan generik. Contohnya (Gambar 5.14).

```
class Shape {
    public double area() { return 0; }
}

class Circle extends Shape {
    private double radius;
    public Circle(double r) { radius = r; }
    @Override
    public double area() { return Math.PI * radius * radius; }
}

class Rectangle extends Shape {
    private double width, height;
    public Rectangle(double w, double h) { width = w; height = h; }
    @Override
    public double area() { return width * height; }
}
```

Gambar 5. 14. Inheritance, Polymorphism, And Generics

Hierarki ini, mengingatkan pada contoh dalam "Pola Desain Kepala Pertama" (Freeman & Robson, 2004), menunjukkan bagaimana pewarisan memungkinkan penggunaan kembali kode (area() dalam Bentuk) dan perilaku polimorfik (implementasi berbeda dari area()).

Kesimpulannya, tipe yang ditentukan pengguna adalah bukti kekuatan ekspresif bahasa pemrograman modern. Mereka memungkinkan pemrogram untuk memperluas bahasa agar sesuai dengan domain masalahnya, sehingga menghasilkan kode yang lebih mudah dibaca, dipelihara, dan tidak terlalu rawan kesalahan. Seperti yang dikatakan Bjarne Stroustrup, pencipta C++, "Tipe adalah entitas sintaksis dan semantik. Tipe mendefinisikan makna data, bukan hanya bagaimana data tersebut direpresentasikan" (Stroustrup, 2013). Dengan menguasai tipe yang ditentukan pengguna, pemrogram dapat membuat abstraksi yang sesuai dengan bahasa masalahnya, membuat pengembangan perangkat lunak yang kompleks menjadi lebih mudah dikelola dan efektif.

5.6 Konversi dan Casting Tipe Data

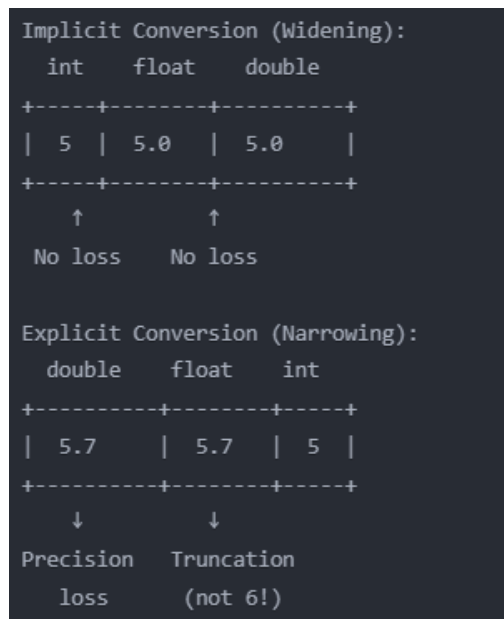
Konversi tipe dan casting adalah konsep dasar dalam pemrograman yang berhubungan dengan perubahan nilai dari satu tipe data ke tipe data lainnya. Operasi ini sangat penting karena tipe yang berbeda memiliki rentang, presisi, dan perilaku yang berbeda. Memahami konversi tipe sangat penting untuk menulis kode yang kuat, efisien, dan bebas bug, terutama ketika bekerja dengan bahasa yang memiliki pengetikan statis (Sebesta, 2015).

Perbedaan antara konversi tipe implisit dan eksplisit sudah ada sejak bahasa pemrograman awal. Dalam makalahnya "Laporan Revisi ALGOL 60," Naur (1963) menjelaskan bagaimana ALGOL 60 menangani konversi otomatis (implisit) dan paksa (eksplisit). Perbedaan ini juga diterapkan pada bahasa-bahasa modern, dengan konversi implisit sering disebut "paksaan" dan konversi eksplisit disebut "pemeran".

Ada dua tipe utama konversi tipe:

1. Konversi Tipe Implisit (Pemaksaan): Secara otomatis dilakukan oleh kompiler ketika tipenya kompatibel tetapi tidak identik.
2. Konversi Tipe Eksplisit (Casting): Ditentukan secara manual oleh pemrogram, sering kali untuk mengesampingkan aturan kompiler.

Mari kita visualisasikan konversi ini (Gambar 5.15).



Gambar 5. 15. Visualisasi Konversi

Diagram Gambar 5.15, yang terinspirasi oleh diskusi dalam "C++ How to Program" (Deitel & Deitel, 2016), menunjukkan bagaimana konversi pelebaran (int ke float menjadi double) aman dan implisit, sementara konversi yang mempersempit (double to float to int) dapat kehilangan informasi dan membutuhkan pemeran eksplisit.

Konversi implisit sering terjadi dalam ekspresi tipe campuran (Gambar 5.16).

```
int i = 10;
double d = 3.14;
double result = i + d; // i is implicitly converted to double
```

Gambar 5. 16. Mixed-Type Expression

Kode Gambar 5.16 ini, mirip dengan contoh dalam "Bahasa Pemrograman C" (Kernighan & Ritchie, 1988), menunjukkan bagaimana nilai int i secara otomatis dipromosikan menjadi dua kali lipat untuk mencocokkan d. Ini umumnya aman karena double dapat mewakili semua nilai int dengan tepat.

Namun, konversi implisit dapat menyebabkan bug yang tidak kentara (Gambar 5.17)

```
unsigned char byte = 200;
int value = byte * 2; // value is 144, not 400!
```

Gambar 5. 17. Implicit Conversions Can Lead to Subtle Bugs

Masalah (Gambar 5.17) ini, yang disorot dalam "C++ Efektif" (Meyers, 2005), terjadi karena byte pertama kali dipromosikan menjadi int, tetapi sebagai karakter yang tidak ditandatangani, ia membungkus ($200 * 2 = 400$, tetapi dalam 8 bit, ini adalah 144).

Pemeran eksplisit digunakan ketika pemrogram mengetahui lebih banyak tentang data dibandingkan kompiler (Gambar 5.18).

```
double pi = 3.14159;
int approximation = (int)pi; // C-style cast
int better = static_cast<int>(pi); // C++ style
```

Gambar 5. 18. Explicit casts

Static_cast gaya C++, diperkenalkan dalam "Desain dan Evolusi C++" (Stroustrup, 1994), lebih disukai karena lebih terlihat dan tidak dapat melakukan konversi berbahaya tertentu yang dapat dilakukan oleh pemeran gaya C.

Untuk meringkas aturan dan bahaya konversi (Tabel 5.6).

Tabel 5. 6. Conversion Rules And Dangers

Conversion Type	From	To	Safety	Possible Issues
Widening	int	float	Safe	Minor: Might lose some precision for large ints
Widening	float	double	Safe	None
Narrowing	double	float	Unsafe	Loss of precision
Narrowing	float/ double	int	Unsafe	Truncation of fractional part
Mixed	signed	unsigned	Unsafe	Negative values wrap around
Pointer	void*	T*	Varies	Negative values wrap around

Tabel (Tabel 5.6) ini, terinspirasi oleh "C++ Templates: The Complete Guide" (Vandevoorde & Josuttis, 2017), menunjukkan mengapa memahami rentang dan semantik tipe target sangat penting.

Dalam bahasa berorientasi objek, konversi menjadi lebih kompleks dengan pewarisan (Gambar 5.179).

```
class Shape { public: virtual ~Shape() {} };  
class Circle : public Shape { /* ... */ };  
  
Shape* shape = new Circle();  
Circle* circle = static_cast<Circle*>(shape); // OK if shape points to a Circle  
Circle* bad = static_cast<Circle*>(new Shape()); // Undefined behavior!
```

Gambar 5. 19. Circle

Kode (Gambar 5.19) ini, mengingatkan pada contoh dalam "C++ Modern Efektif" (Meyers, 2014), menunjukkan bagaimana `static_cast` dapat digunakan untuk downcasting dalam hierarki warisan. Namun, lebih aman menggunakan `Dynamic_cast`, yang memeriksa saat runtime apakah konversinya valid (Gambar 5.20).

```
Circle* safe = dynamic_cast<Circle*>(shape); // nullptr if shape is not a Circle
```

Gambar 5. 20. Dynamic Cast

Kesimpulannya, konversi tipe adalah pedang bermata dua. Konversi implisit dapat membuat kode lebih ringkas dan bekerja secara lancar dengan ekspresi tipe campuran. Namun, mereka juga dapat menyembunyikan bug. Pemeran eksplisit memberikan lebih banyak kontrol tetapi juga dapat melewati pemeriksaan keamanan sistem tipe. Seperti yang diperingatkan oleh Bjarne Stroustrup, "Pemeran adalah cara memberi tahu kompiler untuk berhenti memeriksa dan membiarkan pemrogram mengambil alih" (Stroustrup, 2013). Oleh karena itu, gips harus digunakan secara bijaksana, dengan pemahaman yang jelas tentang jenis gips dan potensi konsekuensinya.

5.7 Tipe Data dalam Berbagai Bahasa Pemrograman

Konsep tipe data bersifat universal dalam pemrograman, namun secara spesifik bagaimana tipe didefinisikan, diperiksa, dan dimanipulasi bervariasi secara signifikan antar bahasa. Variasi ini mencerminkan filosofi desain, domain aplikasi, dan

konteks sejarah yang berbeda. Memahami perbedaan-perbedaan ini sangat penting bagi pemrogram poliglوت dan untuk memilih bahasa yang tepat untuk tugas tertentu (Van Roy & Haridi, 2004).

Evolusi tipe data terkait dengan evolusi bahasa pemrograman. Dalam "History of Programming Languages," Wexelblat (1981) menelusuri bagaimana bahasa awal seperti FORTRAN hanya memiliki beberapa tipe dasar (integer, real, logic), sedangkan bahasa selanjutnya menambahkan lebih banyak tipe dan fitur terkait tipe untuk meningkatkan ekspresi dan keamanan.

Berikut perbandingan pengetikan data tingkat tinggi di seluruh paradigma bahasa (Gambar 5.21).

Procedural	Object-Oriented	Functional	Dynamic
+-----+	+-----+	+-----+	+-----+
int x = 5;	class Point	let x = 5	x = 5
float y=3.14	{	let y = 3.14	y = 3.14
struct Pt {	int x;	type point =	p = {x:1,
int x, y;	int y;	{x:int, y:2}	
};	method m	y:float}	
+-----+	+-----+	+-----+	+-----+
FORTAN, C	Java, C++	Haskell, ML	Python, Ruby

Gambar 5. 21. High-Level Comparison Of Data Typing Across Language Paradigms

Diagram (Gambar 5.18) ini, yang terinspirasi oleh “Konsep, Teknik, dan Model Pemrograman Komputer” (Van Roy & Haridi, 2004), menunjukkan bagaimana paradigma yang berbeda mewakili konsep yang serupa. Bahasa prosedural menggunakan tipe dan struct dasar, bahasa berorientasi objek merangkum data dan metode, bahasa fungsional menekankan kekekalan dan inferensi tipe, dan bahasa dinamis sering kali memiliki variabel tipe bebek yang fleksibel.

Pengetikan statis vs. dinamis adalah poros variasi utama:

- 1. Bahasa yang diketik secara statis (C++, Java, Rust) memeriksa tipe pada waktu kompilasi (Gambar 5.22).

```
int x = 10;
x = "hello"; // Error: cannot convert string to int
```

Gambar 5. 22. Statically-typed languages (C++, Java, Rust)

- 2. Bahasa yang diketik secara dinamis (Python, JavaScript, Ruby) memeriksa tipe saat runtime (GAMABR 5.20).

```
x = 10
x = "hello" # OK: x can hold any type
```

Gambar 5. 23. Dynamically-typed languages (Python, JavaScript, Ruby)

Perbedaan ini, yang dibahas dalam "Pragmatika Bahasa Pemrograman" (Scott, 2015), berdampak pada kecepatan pengembangan, deteksi kesalahan, dan kinerja. Pengetikan statis menangkap kesalahan lebih awal tetapi memerlukan perencanaan lebih awal.

Perbedaan utama lainnya adalah cara bahasa menangani konversi tipe (Gambar 5.24).

C/C++	Java	Python
+-----+	+-----+	+-----+
int x = 5;	int x = 5;	x = 5
double y = x	double y =x	y = float(x)
// y is 5.0	// y is 5.0	# y is 5.0
+-----+	+-----+	+-----+

Gambar 5. 24. Key Difference Is How Languages Handle Type Conversion

Di C/C++, konversi implisit adalah hal biasa. Java lebih ketat, sering kali membutuhkan pemeran yang eksplisit. Python, yang diketik secara dinamis, menggunakan fungsi konversi eksplisit. Pilihan-pilihan ini mempengaruhi kenyamanan dan potensi bug halus, seperti yang dibahas dalam "The Art of Computer Programming" (Knuth, 1997).

Bahasa berorientasi objek memperkenalkan pewarisan, yang mempersulit pengetikan (Gambar 5.25)

```
// Java
class Animal { void speak() { /* ... */ } }
class Dog extends Animal { void speak() { /* ... */ } }

Animal a = new Dog(); // OK: Dog is-a Animal
Dog d = (Dog) a;      // OK if a refers to a Dog
```

Gambar 5. 25. Object-oriented languages introduce inheritance

"Prinsip Substitusi Liskov" ini, dinamai menurut Barbara Liskov dan dijelaskan dalam "Pemrograman Berorientasi Objek: Rasa Gaya yang Objektif" (Liskov & Zilles, 1974), memungkinkan objek subtype digunakan di mana objek supertype diharapkan.

Bahasa fungsional sering kali memiliki fitur tipe lanjutan (Gambar 5.26).

```
-- Haskell
data Maybe a = Just a | Nothing

safeDiv :: Int -> Int -> Maybe Int
safeDiv _ 0 = Nothing
safeDiv a b = Just (a `div` b)
```

Gambar 5. 26. Functional languages often have advanced type features

Kode Haskell ini, mirip dengan contoh di "Pelajari Haskell untuk Kebaikan Besar!" (Lipovača, 2011), menggunakan tipe data aljabar dan polimorfisme parametrik untuk menyatakan kemungkinan kegagalan dalam pembagian tanpa menggunakan pengecualian atau nilai nol.

Untuk meringkas perbedaan utama (Table 5.7).

Tabel 5. 7. Key Differences

Feature	C/C++	Java	Python	Haskell	Erlang
Typing	Static	Static	Dynamic	Static, Strong	Dynamic, Strong
Null/None	Yes (unsafe)	Yes (checked)	Yes (None)	No (Maybe)	No
Generics/Templates	Yes	Yes	No (duck typing)	Yes (parametric)	No
Exceptions	Optional	Checked	Yes	No (monads)	No (let-it-crash)

Tabel (Table 5.7) ini, terinspirasi oleh "Tujuh Bahasa dalam Tujuh Minggu" (Tate, 2010), menunjukkan bagaimana pilihan desain bahasa memengaruhi penanganan kesalahan, penggunaan kembali kode, dan gaya pemrograman secara keseluruhan.

Kesimpulannya, tipe data adalah sebuah lensa yang melaluinya kita dapat melihat beragam lanskap bahasa pemrograman. Dari tipe C yang kaku dan berorientasi pada kinerja hingga tipe Python yang fleksibel dan berfokus pada produktivitas, hingga tipe Haskell yang ketat secara matematis, setiap sistem tipe bahasa mencerminkan prioritas dan filosofinya. Seperti yang dikatakan Tony Hoare, penemu referensi nol, "Saya menyebutnya kesalahan miliaran dolar" (Hoare, 2009), menyoroti bagaimana keputusan sistem yang kecil sekalipun dapat memiliki dampak yang sangat besar.

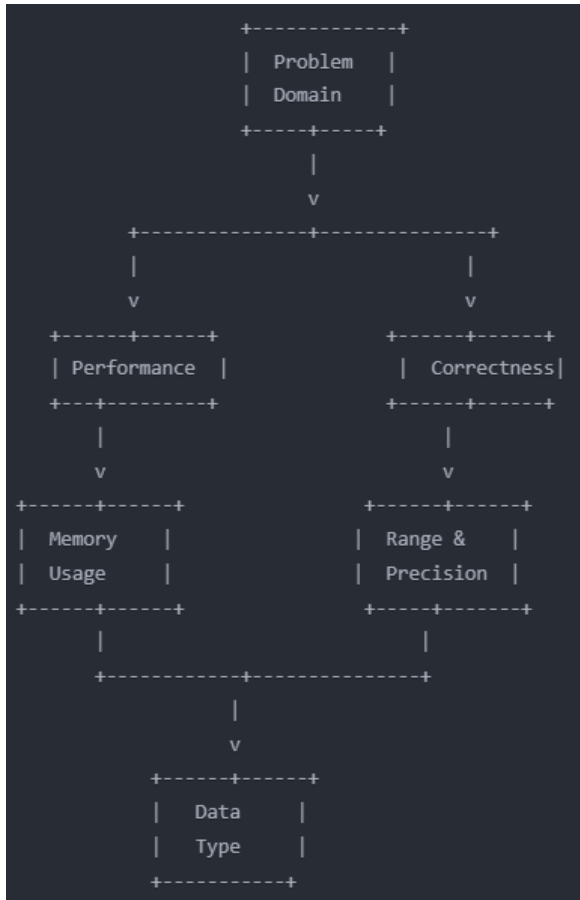
Memahami perbedaan-perbedaan ini memberdayakan pemrogram untuk memilih bahasa yang tepat untuk tugas tersebut dan menulis kode yang tidak hanya fungsional, namun juga aman, efisien, dan dapat dipelihara.

5.8 Memilih Tipe Data yang Tepat

Memilih tipe data yang sesuai adalah keputusan penting yang mempengaruhi setiap aspek program, mulai dari kebenaran dan efisiensi hingga kejelasan dan pemeliharaannya. Ini adalah tugas yang memerlukan pemahaman tidak hanya tipe yang tersedia tetapi juga domain masalah, persyaratan kinerja, dan seluk-beluk bagaimana tipe berinteraksi. Seperti yang dikatakan Linus Torvalds, pencipta Linux, "Pemrogram yang buruk mengkhawatirkan kodenya. Pemrogram yang baik mengkhawatirkan struktur data dan hubungannya" (Raymond, 2003).

Pentingnya memilih tipe data yang tepat digarisbawahi dalam teks dasar seperti "Praktik Pemrograman" (Kernighan & Pike, 1999). Mereka berpendapat bahwa struktur data adalah kerangka kerja di mana program dibangun, dan kejelasan tipe data mengarah pada kejelasan algoritma. Hal ini digaungkan dalam karya-karya yang lebih baru seperti "Clean Code" (Martin, 2008), yang menekankan bahwa tipe yang dipilih dengan baik membuat kode terdokumentasi sendiri.

Mari kita visualisasikan faktor-faktor yang mempengaruhi pilihan jenis (Gambar 5.24).



Gambar 5. 27. Factors That Influence Type Choice

Diagram ini, yang terinspirasi dari diskusi dalam "Code Complete" (McConnell, 2004), menunjukkan bahwa pilihan tipe data adalah keseimbangan antara persyaratan masalah, kebutuhan kinerja, dan batasan kebenaran.

Pertimbangkan tipe data numerik. Dalam "Computer Systems: A Programmer's Perspective" (Bryant & O'Hallaron, 2015), penulis memberikan tabel berikut (Tabel 5.8).

Tabel 5. 8. Key Differences

Type	Size (bits)	Range (approx.)	Precision
int8	8	-128 to 127	Perfect
int32	32	-2^{31} to $2^{31}-1$	Perfect
float32	32	$\pm 3.4\text{e}38$	~7 digits
float64	64	$\pm 1.8\text{e}308$	~15 digits

Untuk menghitung barang di toko kecil, int8 mungkin cukup. Untuk penghitungan populasi, int32 atau bahkan int64 mungkin diperlukan. Untuk simulasi ilmiah, jangkauan dan ketepatan float64 (ganda) bisa jadi sangat penting. Menggunakan tipe besar yang tidak perlu akan membuang-buang memori; menggunakan tipe yang terlalu kecil berisiko meluap atau meluap.

Dalam "Java Efektif" (Bloch, 2018), penulis memperingatkan agar tidak menggunakan tipe floating-point untuk perhitungan moneter (Gambar 5.28).

```
// DON'T DO THIS
double funds = 1.00;
funds -= 0.10;
funds -= 0.20;
System.out.println(funds == 0.70); // Prints false!
```

Gambar 5. 28. Warns against using floating-point types

Karena masalah representasi biner, cetakan ini salah. Untuk uang, gunakan BigDecimal, int, atau long (menyimpan sen).

Untuk data tekstual, pilihannya sering kali bergantung pada rangkaian karakter yang diperlukan (Gambar 5.29).

```
// C++
char ascii_char = 'A'; // 1 byte, ASCII
wchar_t wide_char = L'A'; // 2 or 4 bytes, Unicode
```

Gambar 5. 29. The choice often depends on the character set needed

Seperti yang dibahas dalam "Bahasa Pemrograman C++" (Stroustrup, 2013), `char` cocok untuk teks bahasa Inggris, tetapi teks internasional memerlukan `wchar_t` atau tipe string yang mendukung Unicode.

Dalam aplikasi yang banyak data, pilihan jenis kontainer juga penting (Gambar 5.30).

```
# Python
books = ["Dune", "Ender's Game"] # List: ordered, mutable
authors = {"Herbert", "Card"} # Set: unordered, no duplicates
ratings = {"Dune": 4.5, "Ender's Game": 4.7} # Dict: key-value pairs
```

Gambar 5. 30. The choice of container type

"Python Cookbook" (Beazley & Jones, 2013) menjelaskan bagaimana penggunaan wadah yang tepat dapat meningkatkan kinerja secara drastis. Misalnya, pengujian keanggotaan adalah $O(n)$ dalam daftar tetapi $O(1)$ dalam satu set.

Dalam pemrograman sistem, tipe data tingkat bit sering digunakan (Gambar 5.31).

```
// C
#include <stdint.h>
uint8_t status_flags;
#define IS_ACTIVE (1 << 0)
#define IS_ADMIN (1 << 1)
#define IS_VERIFIED (1 << 2)

status_flags |= IS_ACTIVE; // Set the flag
if (status_flags & IS_ADMIN) { /* ... */ } // Check the flag
```

Gambar 5. 31. Bit-level data types

"Hacker's Delight" (Warren, 2012) penuh dengan teknik memutar-mutar, yang penting dalam pemrograman tingkat rendah untuk efisiensi dan kekompakan.

Kesimpulannya, memilih tipe data yang tepat adalah keputusan yang memiliki banyak aspek. Hal ini memerlukan pemahaman domain masalah (Apakah ini untuk komputasi ilmiah? Transaksi keuangan?), kebutuhan kinerja (Apakah kecepatan penting? Memori ketat?), dan nuansa tipe (Apakah int ini akan meluap? Apakah presisi floating-point memadai?). Seperti yang dikatakan Knuth (1974) yang terkenal, "Optimasi dini adalah akar segala kejahatan," namun dia juga mencatat bahwa kita "tidak boleh melewatkan peluang kita dalam kondisi kritis 3% tersebut." Seringkali, 3% kritis tersebut melibatkan pemilihan tipe data yang tepat. Keseimbangan antara kejelasan, kebenaran, dan kinerja merupakan inti dari desain perangkat lunak yang baik.

DAFTAR PUSTAKA

- Aho, A. V., Hopcroft, J. E., & Ullman, J. D. 1983. Data Structures and Algorithms. Addison-Wesley.
- Bloch, J. 2018. Effective Java (3rd ed.). Addison-Wesley.
- Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. 2007. Object-Oriented Analysis and Design with Applications (3rd ed.). Addison-Wesley.
- Bryant, R. E., & O'Hallaron, D. R. 2015. Computer Systems: A Programmer's Perspective (3rd ed.). Pearson.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. 2009. Introduction to Algorithms (3rd ed.). MIT Press.
- Crochemore, M., & Lecroq, T. 2004. Algorithms on Strings. Cambridge University Press.
- Dasgupta, S., Papadimitriou, C. H., & Vazirani, U. V. 2008. Algorithms. McGraw-Hill.
- Deitel, P. J., & Deitel, H. M. 2016. C++ How to Program (10th ed.). Pearson.
- Fowler, M. 2003. UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley.
- Freeman, E., & Robson, E. 2004. Head First Design Patterns. O'Reilly Media.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. 1994. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- Guttag, J., & Horning, J. 1978. On Abstract Data Types. Communications of the ACM, 21(8), 560-564.
- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. 2014. Data Structures and Algorithms in Java (6th ed.). Wiley.
- Gregory, J. 2018. Game Engine Architecture (3rd ed.). CRC Press.

- Hoare, C. A. R. 2009. Null References: The Billion Dollar Mistake. QCon London.
- Kernighan, B. W., & Ritchie, D. M. 1988. The C Programming Language (2nd ed.). Prentice Hall.
- Knuth, D. E. 1997. The Art of Computer Programming, Volume 2: Seminumerical Algorithms (3rd ed.). Addison-Wesley.
- Lipovača, M. 2011. Learn You a Haskell for Great Good!: A Beginner's Guide. No Starch Press.
- Lippman, S., Lajoie, J., & Moo, B. E. 2012. C++ Primer (5th ed.). Addison-Wesley.
- Liskov, B. 1991. Data Abstraction and Hierarchy. ACM SIGPLAN Notices, 23(5), 17-34.
- Liskov, B., & Zilles, S. 1974. Programming with Abstract Data Types. ACM SIGPLAN Notices, 9(4), 50-59.
- Martin, R. C. 2008. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
- McCarthy, J. 1960. Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I. Communications of the ACM, 3(4), 184-195.
- McDowell, G. L. 2015. Cracking the Coding Interview (6th ed.). CareerCup.
- Meyer, B. 1997. Object-Oriented Software Construction (2nd ed.). Prentice Hall.
- Meyers, S. 2005. Effective C++: 55 Specific Ways to Improve Your Programs and Designs (3rd ed.). Addison-Wesley.
- Meyers, S. 2014. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14. O'Reilly Media.
- Naur, P. (Ed.). 1963. Revised Report on the Algorithmic Language ALGOL 60. The Computer Journal, 5(4), 349-367.

- Sebesta, R. W. 2015. Concepts of Programming Languages (11th ed.). Pearson.
- Scott, M. L. 2015. Programming Language Pragmatics (4th ed.). Morgan Kaufmann.
- Stroustrup, B. 2014. Programming: Principles and Practice Using C++ (2nd ed.). Addison-Wesley.
- Stroustrup, B. 2013. The C++ Programming Language (4th ed.). Addison-Wesley.
- Stroustrup, B. 1994. The Design and Evolution of C++. Addison-Wesley.
- Tate, B. A. 2010. Seven Languages in Seven Weeks: A Pragmatic Guide to Learning Programming Languages. Pragmatic Bookshelf.
- Vandevorde, D., & Josuttis, N. M. 2017. C++ Templates: The Complete Guide (2nd ed.). Addison-Wesley.
- Van Roy, P., & Haridi, S. 2004. Concepts, Techniques, and Models of Computer Programming. MIT Press.
- Weiss, M. A. 2013. Data Structures and Algorithm Analysis in C++ (4th ed.). Pearson.
- Wexelblat, R. L. (Ed.). 1981. History of Programming Languages. Academic Press.
- Wirth, N. 1976. Algorithms + Data Structures = Programs. Prentice Hall.
- Wirth, N. 1971. The Programming Language Pascal. Acta Informatica, 1(1), 35-63.

BAB 6

TEORI TENTANG PEMROGRAMAN

Oleh Dahlan Susilo

6.1 Pendahuluan

Kemajuan teknologi informasi dan komunikasi yang sangat pesat telah menyumbangkan peran besar dalam perubahan peradaban manusia dari masa ke masa. Salah satu penyumbang kemajuan tersebut merupakan peran besar dari perangkat lunak (*software*) yang telah berkembang sangat pesat mengikuti kemajuan teknologi perangkat keras (*hardware*). Perangkat keras teknologi informasi hanya akan berfungsi apabila terpasang program (*software*) di dalamnya sesuai dengan kebutuhan penggunaannya. Perangkat lunak tersusun atas kumpulan instruksi berupa kode-kode tertentu menggunakan bahasa pemrograman dan dikompilasi dengan menggunakan *compiler* tertentu hingga dapat dieksekusi untuk melaksanakan tugas tertentu (Zaid Lee, 2022).

Keberadaan program tidak lepas dari peran para penyusunnya, yakni *programmer*. Program yang telah dibuat dapat melaksanakan melaksanakan tugas-tugas yang diberikan oleh *programmer* sesuai kebutuhan sistem yang diperoleh dari kebutuhan operasi masing-masing pengguna. Seorang *programmer* bekerja sesuai dengan keahliannya di dalam menyusun kode program yang dibuatnya sesuai dengan desain sistem yang telah dirancang oleh analis sistem. Keberhasilan seorang *programmer* dalam menerjemahkan sebuah desain sistem ke dalam sebuah program sangat tergantung pada

penguasaannya dalam bahasa pemrograman yang digunakan untuk menyusun program tertentu.

6.2 Pengertian Pemrograman

Pemrograman merupakan aktivitas menyusun perintah kepada komputer untuk melakukan tugas-tugas tertentu dengan menggunakan suatu bahasa pemrograman (Zaid Lee, 2022). Program merupakan hasil pengkodean yang dilakukan oleh *programmer* yang berisi instruksi-instruksi yang akan dimengerti komputer. Komputer akan mengeksekusi program tersebut untuk menyelesaikan tugas-tugas tertentu. Kumpulan tugas yang ada di dalam program akan dilaksanakan oleh komputer satu per satu sesuai dengan instruksi yang sudah disusun oleh *programmer*.

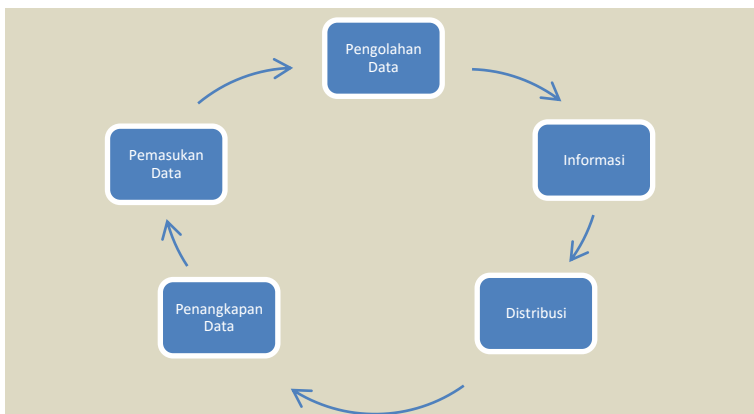
Pemrograman dasar merupakan pembelajaran yang menitikberatkan pada pembelajaran dasar-dasar logika tujuan umum dan pembentukan cara berpikir untuk menciptakan program yang efektif dan efisien. Dasar-dasar yang bersifat universal tersebut dapat diterapkan pada bahasa pemrograman apapun, kapanpun dan di manapun. Penggunaan bahasa pemrograman sangat bervariasi tergantung kebutuhan sistem dan penguasaan *programmer* yang menyusunnya.

Industri pengembangan perangkat lunak terus berkembang pesat, memberikan tuntutan yang berbeda pada bahasa pemrograman. Persyaratan bahasa akan berubah secara tiba-tiba disebabkan proyek baru memerlukan teknologi berbeda. Perintah yang akan diterima komputer mungkin memiliki sintaksis (aturan pengaturan kode) yang berbeda tergantung pada bahasanya. Logika dan algoritma yang disusun bersifat umum, sehingga akan diterjemahkan dengan bahasa pemrograman apapun. Oleh sebab itu, penting bagi *programmer* untuk menguasai logika, algoritma, dan konstruksi bahasa

pemrograman sehingga, apa pun bahasa pemrograman yang digunakannya.

Programmer akan menginstruksikan komputer untuk melakukan tugas-tugas tertentu sesuai dengan sistem riil yang dihadapi. Aktivitas menyusun program tentu tidak lepas dari aktivitas riil dari prosedur-prosedur dalam sebuah sistem yang akan dibuatkan programnya. Oleh sebab itu, seorang *programmer* harus memahami siklus pengolahan data dengan baik agar program yang dihasilkan berkualitas prima.

Siklus pengolahan data pada komputer dimulai dari pengumpulan data (*origination*), pemasukan data (*input*), pengolahan data (*process*), penyajian informasi (*output*), distribusi informasi (*distribution*), dan didukung dengan penyimpanan data/informasi (*storage*). Siklus tersebut dapat dilihat pada Gambar 6.1. Kompilasi instruksi-instruksi yang sudah ditulis dalam bahasa pemrograman menggunakan pola berpikir yang sama dengan pola berpikir manusia ketika akan menyelesaikan suatu masalah. Dengan kata lain, pola berpikir yang digunakan dalam bahasa pemrograman tidak lain mengadopsi pola berpikir manusia (Zaid Lee, 2022).



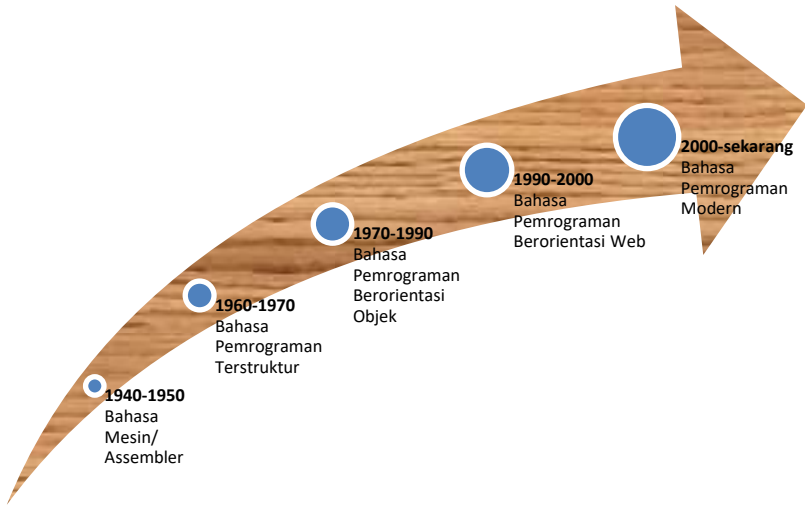
Gambar 6. 1. Siklus Pengolahan Data

(Sumber: Zaid Lee, 2022)

Pada tahap **Penangkapan Data** dilakukan pengumpulan data yang biasanya merupakan proses pencatatan (*recording*) data ke dalam dokumen dasar. Berdasar dokumen dasar tersebut kemudian dilakukan **Pemasukan Data**, yakni proses yang dilakukan *operator* untuk memasukan data yang diperoleh dari penangkatan data ke dalam perangkat komputer menggunakan alat input (*input device*). Berikutnya, masuk tahap **Pengolahan Data** dilakukan pemroses berupa proses menghitung, membandingkan, mengurutkan, mengklasifikasikan, mengendalikan, membaca/menyimpan data dari/ke dalam **Simpanan Luar** (*storage*). Dari hasil **Pengolahan Data** akan menghasilkan informasi sebagai luaran (*output*) yang disajikan melalui alat luaran (*output device*). **Distribusi** merupakan tahapan untuk mendistribusikan luaran (*output*) kepada pihak yang membutuhkan informasi. Untuk mendukung pengelolaan data/informasi jangka panjang mutlak membutuhkan **Simpanan Luar** (*storage*) sebagai alat penyimpan data/informasi. **Storage** akan menampung data primer maupun data sekunder yang akan digunakan pada masa-masa mendatang. Data dan informasi yang sudah tersimpan di dalam *storage*, sewaktu-waktu akan dibaca kembali untuk dilakukan pengolahan data lebih lanjut.

6.3 Sejarah Pemrograman

Pemrograman memiliki sejarah yang panjang sesuai dengan jamannya. Pemrograman sudah ada sejak komputer pertama dibuat, disebabkan komputer hanya bisa bekerja bila dipasang program di dalamnya. Setiap bahasa pemrograman memiliki sejarahnya masing-masing dan berperan penting dalam setiap perkembangan dunia teknologi informasi (Rendy Almaheri, 2023). Evolusi bahasa pemrograman dari masa ke masa dapat dilihat pada Gambar 6.2.



Gambar 6. 2. Evolusi Bahasa Pemrograman

(Sumber: Rendy Almaheri, 2023)

6.3.1 Periode Tahun 1940 — 1950: Bahasa Mesin (*Machine Language*) dan Bahasa Assembler (*Assembly Language*)

Sejarah pemrograman dimulai sejak terciptanya komputer pada tahun 1940-an. Pada awal dibuatnya komputer digunakan bahasa pemrograman berupa bahasa mesin. Kode program terdiri dari kode mesin yang akan langsung dieksekusi oleh komputer. Saat itu, kecepatan komputer terbatas dan memori yang sangat terbatas. Bahasa mesin merupakan bentuk paling dasar dari bahasa pemrograman, dan sebab memerlukan instruksi yang ditulis dalam kode numerik, tersebut bisa menjadi sangat kompleks dan rawan kesalahan.

Kompleksitas bahasa mesin menyebabkan pemrogram mengembangkan bahasa rakitan, sebuah bahasa simbolik untuk bahasa mesin. Setiap operasi bahasa mesin memiliki simbol masing-masing, seperti kata “MOVE” untuk pemindahan, “ADD” untuk penjumlahan, “MUL” untuk perkalian, dan “SUB” untuk pembagian. Dengan kata kunci dalam bahasa inggris tersebut

sangat membantu *programmer* dalam menyusun perintah yang ditujukan kepada komputer akan dilakukan dengan lebih mudah. Perkembangan bahasa assembly merupakan kemajuan besar dalam dunia pemrograman. Bahasa pemrograman assembly menjadi langkah pertama menuju pemrograman yang lebih manusiawi. Kemudahan tersebut masih dirasakan sangat sulit bagi manusia pada umumnya untuk mempelajari dikarenakan sangat bergantung pada arsitektur perangkat keras yang akan digunakan.

Pada tahun 1948, Konrad Zuse menerbitkan esai tentang bahasa pemrograman Plankalkül miliknya kepada publik. Namun karyanya tidak pernah digunakan sebab terasing dari bahasa pemrograman lain. Selain Plankalkül, bahasa pemrograman yang dikembangkan pada tahun 1940-an merupakan sistem pengkodean ENIAC pada tahun 1943 dan C-10 pada tahun 1949 (Appkey, 2023).

6.3.2 Periode Tahun 1950 — 1970: Bahasa Pemrograman Terstruktur (*Structured Programming Language*)

Bahasa pemrograman yang dikembangkan antara tahun 1950-an dan 1960-an antara lain FORTRAN pada tahun 1955 dikembangkan oleh John W. Buckuss. FORTRAN merupakan bahasa pemrograman pertama yang dikembangkan untuk mendukung tujuan ilmiah dan teknis. Hal tersebut memudahkan para ilmuwan dan insinyur untuk mengembangkan perangkat lunak numerik, dan masih digunakan dalam komputasi ilmiah hingga saat ini (Rendy Almaheri, 2023).

LISP (*LIS*t *Program*ming) dikembangkan oleh John McCarthy pada tahun 1958 dan COBOL (*Common Business-Oriented Language*) dikembangkan oleh *Short Range Committee* pada tahun 1959, dengan Grace Hopper memainkan peran utamanya. COBOL memungkinkan *programmer* untuk membuat perangkat lunak yang lebih mudah dibaca dan dipahami oleh para pengguna bisnis.

Bahasa BASIC (*Beginners' All-purpose Symbolic Instruction Code*) dikembangkan pada tahun 1964. BASIC merupakan bahasa pemrograman pertama yang dirancang untuk pemula. Bahasa BASIC memberikan pengaruh yang besar terhadap perkembangan bahasa pemrograman pada tahun 1960-an hingga tahun 1970-an.

Beberapa bahasa pemrograman lainnya antara lain seperti SIMULA (*Simulation Programming Languages*) dan APL (*Array-based Programming Language*) dibuat pada tahun 1962. SIMULA merupakan bahasa pemrograman pertama yang dirancang secara khusus untuk mendukung pemrograman berorientasi objek.

6.3.3 Periode Tahun 1970 — 1990: Bahasa Pemrograman Berorientasi Objek (*Object Oriented Language*)

Pada periode tahun 1970 - 1990 berkembang banyak bahasa pemrograman antara lain Pascal, Forth, dan C pada tahun 1970. Bahasa Prolog (*Programming in Logic*), bahasa pemrograman logika pertama, yang dibuat oleh tiga programmer berbeda yaitu Colmerauer, Roussel, dan Kowalski pada tahun 1972. Smalltalk dibuat tahun 1972, ML (*Machine Learning*) tahun 1973, dan SQL (*Structured Query Language*) tahun 1978. Pengembangan bahasa tingkat tinggi ini menggunakan berbagai teknik pemrograman seperti pemrograman prosedural dan pemrograman berorientasi objek/*Object Oriented Programming* (Appkey, 2023).

Letak perbedaan pada pemrograman prosedural dan pemrograman berorientasi objek terletak pada data yang dikelola. Pada pemrograman prosedural, program dijalankan berdasarkan kombinasi variabel terkait, sedangkan pada OOP, program dijalankan berdasarkan objek. Bahasa C memiliki sintaksis yang lebih modern dan fleksibel dan digunakan untuk mengembangkan sistem operasi UNIX yang sangat berpengaruh. Meskipun bahasa pemrograman mengalami kemajuan selama

periode tersebut, tahun 1980-an merupakan periode konsolidasi yang relatif. Pada masa tersebut terjadi perkembangan C++, bahasa pemrograman yang berjalan pada sistem pemrograman berorientasi objek.

Pada tahun 1980-an berkembang beberapa bahasa antara lain Ada tahun 1983, Eiffel tahun 1985, Perl tahun 1987, dan FL tahun 1989. Pada tahun 1983 bahasa C berevolusi menjadi C++ dengan menambahkan fitur pemrograman berorientasi objek. Bahasa tersebut memungkinkan *programmer* membuat program yang lebih kompleks dan struktur yang lebih terorganisir (Rizka Lailatul Amalia, 2020). Bahasa pemrograman Python lahir tahun 1989 dengan keunggulan memiliki sintaks yang mudah dibaca dan dipahami, sehingga sangat populer bagi pemula maupun pengembangan web, analisis data, dan kecerdasan buatan.

Pada tahun 1995, Java dipopulerkan dengan slogan “write once, run anywhere”. Kode program yang ditulis dalam bahasa tersebut dapat berjalan di berbagai platform. Dengan demikian, pengembangan aplikasi menjadi lebih portabel dan dapat digunakan dalam pengembangan aplikasi Android.

6.3.4 Periode Tahun 1990 — 2000: Bahasa Pemrograman Berorientasi Web (*Web Oriented Language*)

Untuk memudahkan *programmer* dalam membuat aplikasi, pada periode tersebut telah dilakukan pengembangan lebih lanjut pada bahasa pemrograman berorientasi objek dan berbasis visual. Gerakan tersebut dimulai pada tahun 1991 dengan Python dan Microsoft Visual Basic 1, dan disusul Delphi yang dikembangkan dari *Pascal for Windows*. Puncaknya terjadi pada tahun 1997 dengan dirilisnya Visual Basic 5 yang memudahkan koneksi ke database dan menggunakan OO-COBOL pada versi Windows. Tidak dapat disangkal lagi, bahwa mayoritas *programmer* percaya masa ini menjadi tahun yang sangat produktif bagi dunia pemrograman.

Bahasa pemrograman yang juga dikembangkan selama periode tersebut antara lain Haskell tahun 1990, Java tahun 1991, Ruby tahun 1993, PHP tahun 1995, dan JavaScript tahun 1995. JavaScript merupakan bahasa pemrograman sisi klien (*browser*) yang digunakan untuk mengembangkan aplikasi web interaktif. Kerangka kerja seperti React, Angular, dan Vue.js memperluas kemampuan JavaScript dalam pengembangan *front-end*.

Ruby dikenal sebagai kerangka *Ruby on Rails* yang memungkinkan pengembangan aplikasi web dapat dilakukan dengan cepat dan efisien. Ruby juga dikenal dengan sintaksisnya yang elegan dan mudah dipahami oleh *programmer*.

6.3.5 Periode Tahun 2000 — sekarang: Bahasa Pemrograman Modern (*Modern Programming Language*)

Selama periode tersebut, vendor bahasa pemrograman lebih cenderung menginstruksikan pemrogram untuk menulis program menggunakan produk baru. Ada dua konsep utama yang menekankan kenyamanan berbasis visual dan berbasis mobile dalam internet (internet untuk *platform mobile*).

Periode tersebut juga dikenal sebagai era CMS (*Content Management System*), dan perkembangan skrip PHP mengalami kemajuan pesat. Tentu saja hal tersebut sangat memudahkan *web programmer* dalam membuat website dengan beragam *template* yang sudah tersedia. Banyaknya modul siap pakai yang sangat memudahkan *programmer* untuk menyusun program yang bagus tanpa harus mempelajari semua scrip dan bahasa HTML.

Beberapa bahasa pemrograman yang dikembangkan pada tahun 2000-an dan masih populer hingga saat ini antara lain TCL/TK, O'Caml, Ruby, dan Python 3.1, Microsoft Visual Net, JavaScript Template, Java (Java 6 JDK, JED, Java Beans, J2ME).

Pada tahun 2009, Google merilis bahasa Golang, yang biasa dikenal dengan bahasa Go. Go merupakan bahasa pemrograman yang dikembangkan oleh Google. Golang fokus pada kinerja cepat dan efisiensi dalam pengembangan aplikasi berbasis server dan sistem terdistribusi.

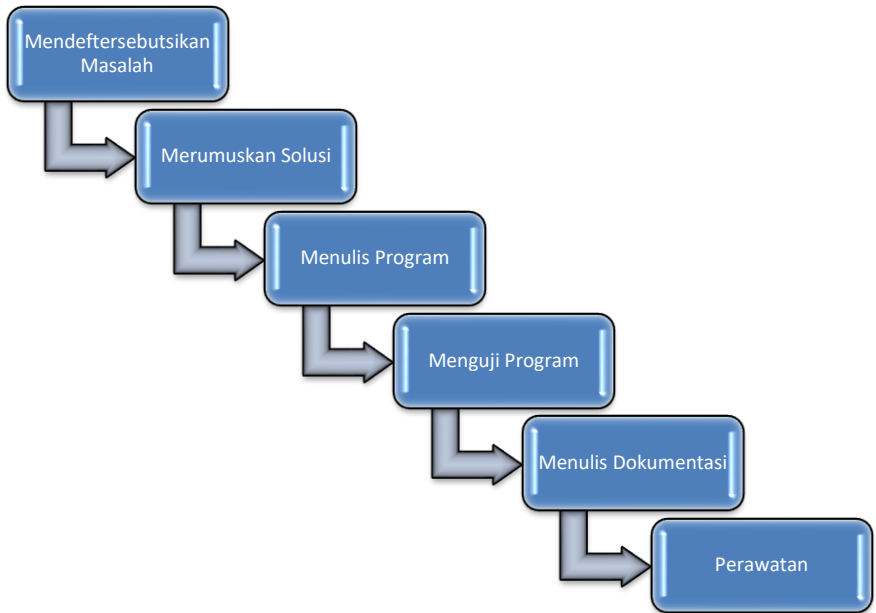
Kotlin yang dirilis tahun 2011 merupakan bahasa pemrograman modern yang digunakan untuk mengembangkan aplikasi Android. Bahasa tersebut lebih ekspresif dan aman dibandingkan Java. Swift dirilis tahun 2014 merupakan bahasa pemrograman yang dikembangkan oleh Apple untuk mengembangkan aplikasi iOS dan macOS. Bahasa tersebut dirancang untuk kinerja tinggi dan kemudahan pengkodean (Rendi Juliarto. 2020).

6.4 Tahapan Pemrograman

Seorang *programmer* harus selalu meng-*update* pengetahuan dan keterampilannya dalam pemrograman sesuai dengan perkembangan jaman. Setiap ada bahasa pemrograman yang baru, harus segera dipelajari oleh *programmer* di industri pengembangan perangkat lunak. Pada saat ini ada beberapa bahasa yang paling banyak digunakan untuk membangun aplikasi berbasis website dan mobile dalam sistem operasi Android atau iOS.

Dalam membuat program yang dibutuhkan klien, diperlukan langkah-langkah agar program akan disusun dengan baik dan benar. Untuk membuat program aplikasi yang baik dan benar, maka langkah-langkah dalam menghasilkan sebuah program dari dicermati dan dijalani dengan sebaik-baiknya. Seorang programmer terkadang harus mengulang-ulang salah satu atau beberapa langkah dari serangkaian langkah untuk menghasilkan sebuah aplikasi yang baik dan benar sesuai kebutuhan para pengguna (Appkey, 2023).

Tahapan pemrograman yang harus dilakukan dalam menyusun program komputer akan dilihat pada Gambar 8.3.



Gambar 6. 3. Tahapan Pemrograman

(Sumber: Appkey, 2023)

6.4.1 Mendefersebutasikan Masalah

Program aplikasi dibuat sesuai dengan masalah yang akan ditangani dengan teknologi informasi. Tahapan awal sebelum menyusun program merupakan mendefersebutasikan masalah dengan tepat. Langkah tersebut sering diabaikan oleh *programmer* terutama ketika bekerja secara individu. *Programmer* membuat program begitu saja saat menekan proyek untuk mengembangkan sebuah program/*software* tanpa mendefersebutasikan masalah terlebih dahulu. Akibat dari hal tersebut merupakan ketidaksesuaiannya spesifikasi program yang dibuat dengan kebutuhan yang diinginkan oleh pengguna (Dimas Setiawan, 2018).

Tahap tersebut sangat penting dilakukan agar program aplikasi akan disusun sesuai dengan permasalahan yang dihadapi oleh pengguna. Tahapan tersebut dilakukan oleh analis sistem yang mampu mengurai sistem yang rumit ke dalam bagian-bagian sistem yang lebih sederhana, sehingga mudah ditemukan permasalahan riil yang akan diselesaikan. Masalah akan mudah dilihat dan diidentifikasi, apabila setiap bagian terkait akan membantu mengungkapkan kebutuhan sistem yang akan dibuatkan programnya.

6.4.2 Merumuskan Solusi

Dari definisi masalah yang jelas untuk diselesaikan, maka akan disusun solusi yang tepat untuk mengatasi setiap permasalahan yang dihadapi oleh pengguna. Permasalahan yang jelas akan menunjukkan *input* dan *output*-nya pun sudah jelas. Langkah selanjutnya merupakan mencari solusi bagaimana sebuah masalah akan diselesaikan. Solusi disusun oleh seorang analis sistem yang harus akan menyusun serangkaian urutan langkah untuk mengolah data input menjadi output yang diharapkan.

Urutan langkah tersebut biasa disebut dengan algoritma. Algoritma disusun atas dasar analisis dan desain sistem yang dibuat dengan cermat dan sungguh-sungguh oleh tim analisis sistem. Dengan algoritma yang baik, akan menghasilkan program aplikasi yang baik pula. Seorang *programmer* tinggal memindahkan logika di dalam algoritma ke dalam kode program yang disusun dengan bahasa pemrograman tertentu.

6.4.3 Menulis Program

Setelah analis sistem menyusun seluruh algoritma yang diperlukan oleh pengguna, tugas *programmer* merupakan mengimplementasikan algoritma tersebut ke dalam kode program. *Programmer* akan menggunakan bahasa pemrograman apa pun yang dikuasai untuk membuat sebuah aplikasi. Namun

ada hal yang perlu diperhatikan yaitu kekompakan dan kecakapan tim pengembang sistem informasi berbasis teknologi informasi.

Setiap bahasa pemrograman mempunyai aturan sintaksis yang berbeda-beda, sehingga *programmer* akan memilih bahasa pemrograman mana yang paling dikuasai atau membuat aplikasi yang sesuai dengan kebutuhan perangkat dan penggunaanya.

6.4.4 Menguji Program

Setelah *programmer* selesai menulis kode program, selanjutnya perlu dilakukan pengujian. Tujuan pengujian program merupakan untuk memastikan bahwa program bebas dari kesalahan sintaksis dan algoritmik. Kesalahan algoritma pada suatu program akan menimbulkan kesalahan (*bug*) dan menghasilkan keluaran yang tidak sesuai harapan. Proses pengujian untuk menemukan kesalahan disebut *debugging*. Proses pengujian dilakukan oleh seorang penguji program aplikasi yang kompeten untuk mencari kelemahan atau kesalahan-kesalahan yang mungkin timbul pada sebuah aplikasi.

6.4.5 Menulis Dokumentasi

Dokumentasi program diperlukan untuk memudahkan perawatan program dan sebagai sarana komunikasi dengan semua pihak yang terkait dengan program aplikasi tersebut. Pemrogram akan menulis komentar singkat di bagian mana pun dari programnya yang menjelaskan tujuan atau penggunaan modul atau bagian kode program tertentu. Proses tersebut akan dilakukan bersamaan dengan pembuatan program. Namun, lebih baik menuliskannya dalam file terpisah dan mencetaknya di kertas agar mudah dilihat saat dibutuhkan.

Dokumentasi program aplikasi yang lengkap berisi spesifikasi sistem, hasil analisis dan desain sistem, kode program, cara melakukan instalasi program aplikasi, cara menjalankan program, dan perawatan program aplikasi.

6.4.6 Perawatan

Prosedur pemeliharaan dilakukan setelah program dibuat dan digunakan oleh pengguna. Kesalahan yang sebelumnya tidak terdeteksi oleh prosedur pengujian program mungkin terjadi setelah digunakan oleh pengguna. Proses pemeliharaan tersebut juga akan menambah fungsionalitas baru pada program yang dibuat, sehingga mengharuskan pemrogram untuk memodifikasi kode program. Oleh sebab itu, pada saat membuat rilis software biasanya terakan perubahan versi seperti versi beta, versi 1.0, versi 1.1, versi 1.2, versi 2.0, dan seterusnya.

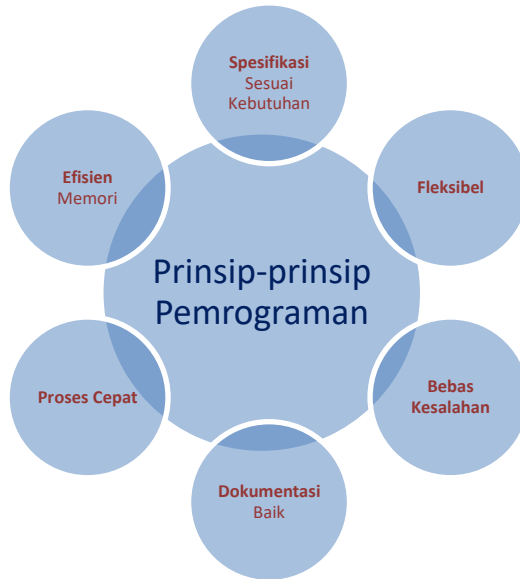
6.5 Prinsip Pemrograman

Seorang programmer dituntut untuk akan menyusun program dengan baik dan benar. Sebuah program dikatakan baik, jika memenuhi 6 prinsip pemrograman (Gamal Thabron, 2022). Gambar 6.4 menggambarkan prinsip-prinsip pemrograman yang baik.

6.5.1 Spesifikasi Sesuai Kebutuhan

Sebuah program yang dibuat harus bisa menangani kebutuhan yang diinginkan penggunanya, mulai dari data yang dimasukkan ke dalam aplikasi (*input*), proses yang dilakukan, hingga ketepatan hasil pemrosesan yang diharapkan (*output*). Perlu diperhatikan dengan seksama apa saja data yang tersedia dan apa saja luaran yang dibutuhkan oleh pengguna, bukan disesuaikan dengan keinginan programmer. Sebaik apapun sebuah program aplikasi tidak memberi manfaat bagi penggunanya, maka akan ditinggalkan oleh pengguna. Sebaiknya,

sesederhana apapun sebuah aplikasi, apabila memberikan manfaat sesuai dengan kebutuhan penggunanya, maka program akan dipakai untuk membantu pekerjaannya.



Gambar 6. 4. Prinsip-prinsip Pemrograman

(Sumber: Gamal Thabron, 2022)

6.5.2 Fleksibel

Program yang baik harus fleksibel dalam pengembangannya sesuai dengan kebutuhan para penggunanya. Untuk menjamin fleksibilitas program, maka *programmer* harus mengikuti desain sistem dan algoritma yang sudah dibuat oleh tim analis sistem. Kesesuaian program dengan desain sistemnya, akan memiliki fleksibilitas yang tinggi bila sewaktu-waktu aplikasi dikembangkan menjadi lebih besar.

6.5.3 Bebas Kesalahan (*error/galat*)

Kesalahan dalam program aplikasi ada 2, yakni kesalahan sintaks (*syntax error*) dan kesalahan algoritma (*algorithmic error*). Kesalahan sintaks menjadi tanggung jawab *programmer*

untuk membetulkan selama program disusun, sedangkan kesalahan algoritma menjadi tanggung jawab analis sistem untuk melakukan pemilihan algoritma yang tepat pada saat menyusun algoritma. Kebenaran program aplikasi merupakan syarat mutlak, agar sebuah program akan diterima oleh pengguna.

6.5.4 Dokumentasi Baik

Dokumentasi sebuah program aplikasi sangat dibutuhkan untuk kebutuhan berbagai pihak terkait baik pada saat program aplikasi baru dirilis maupun masa-masa mendatang. Dokumentasi tersebut sangat berguna pada saat implementasi sistem, sebagai bahan untuk melakukan pelatihan kepada para pengguna, agar setiap pengguna akan menggunakan aplikasi dengan baik dan benar. Dokumen tersebut juga akan menjadi dasar dalam penelusuran kesalahan apabila terjadi kesalahan dalam eksekusi program. Dokumentasi program yang baik juga akan memudahkan *programmer* dalam melakukan pengembangan aplikasi di masa mendatang, bahkan ketika aplikasi akan dikembangkan oleh tim pengembang lain selanjutnya.

6.5.5 Proses Cepat

Program yang baik disusun sesuai dengan algoritma yang sudah disusun oleh tim analis dan menggunakan struktur data yang tepat, sehingga penggunaan memori sangat efisien. Program seperti tersebut tidak hanya bisa memberikan *output* yang tepat, namun seorang pengguna program juga menghendaki proses yang cepat dalam melakukan pemasukan data, pemrosesan data, maupun pada saat menghasilkan *output*. Program yang memiliki waktu akses yang singkat akan sangat mempengaruhi kinerja dari para penggunanya. Prosedur atau fungsi yang memiliki struktur ramping dengan variabel yang efisien akan berjalan dengan waktu yang singkat.

6.5.6 Efesien Memori

Semakin banyak memori komputer yang dibutuhkan suatu program agar akan berfungsi, semakin lama waktu yang dibutuhkan untuk mengubah *input* menjadi *output*. Oleh sebab itu, program yang baik merupakan program yang menggunakan sumber daya memori/random access memory (RAM) sesedikit mungkin. Hal tersebut akan diusahakan oleh programmer dalam memilih tipe data yang tepat dan memerlukan memori terkecil. Dalam hal tersebut kemampuan dan pengalaman *programmer* sangat menentukan untuk menghasilkan program yang akan mengakses memori sehemat mungkin.

6.6 Fungsi Bahasa Pemrograman

Bahasa pemrograman tersusun dari sekumpulan sintaks yang merupakan perintah komputer untuk disusun menjadi suatu program aplikasi. Perintah tersebut akan diterjemahkan menjadi logika yang dimengerti oleh komputer (Yuyun Khairunisa, dkk., 2023).

6.6.1 Media Komunikasi

Bahasa pemrograman berfungsi sebagai jembatan komunikasi antara *programmer* dengan komputer. Komputer hanyalah mesin yang hanya mengerti sinyal digital, yakni on dan off. Sinyal tersebut biasa disajikan dengan nilai “1” untuk sinyal “*on*” dan “0” untuk sinyal “*off*”. Untuk akan memahami sinyal tersebut, maka *programmer* memerlukan alat komunikasi berupa bahasa pemrograman. Bahasa tersebutlah yang akan menyatukan keinginan programmer dengan mesin komputer dalam sebuah komunikasi yang terhubung sesuai keinginan programmer.

Bahasa yang dipahami oleh komputer tentu berbeda dengan bahasa manusia. Oleh sebab itu, perintah bahasa pemrograman yang dibuat oleh *programmer* harus diterjemahkan terlebih dahulu ke dalam bentuk lain agar komputer akan memahaminya. Ada dua metode yang umum digunakan untuk menerjemahkan bahasa pemrograman ke bahasa komputer yaitu metode *compiler* dan metode *interpreter*.

Compiler merupakan alat untuk mengubah semua perintah bahasa pemrograman menjadi paket file baru. Paket file tersebut berisi kode mesin yang akan dibaca oleh komputer atau perangkat lain untuk dilaksanakan. Setiap kali pengguna menjalankan paket file, komputer diaktifkan untuk menjalankan perintah bahasa pemrograman. Ekstensi paket file tersebut mungkin berbeda-beda, bergantung pada perangkat yang menjalankan program. Ekstensi *.EXE untuk perangkat berbasis Windows, *.APK untuk Android, dan *.APP untuk iOS. Bahasa pemrograman termasuk katagori ini antara lain C, C++, C#, Java, Kotlin, dan Swift.

Interpreter merupakan alat yang mengubah perintah bahasa pemrograman menjadi kode mesin tanpa membuat paket file baru. *Interpreter* menerjemahkan perintah tersebut baris demi baris saat program berjalan. Apabila perintah lengkap dan benar, maka interpreter kan dieksekusi oleh komputer, tetapi kalau ada kesalahan perintah, maka akan diberitahukan kesalahannya sesuai kode error-nya. *Interpreter* yang sering digunakan oleh bahasa pemrograman spesialis website antara lain: *web browser*., JavaScript, Ruby, dan PHP.

6.6.2 Alat Pengembangan (*Tools*)

Fungsi bahasa pemrograman yang kedua merupakan sebagai alat untuk mengembangkan sistem dan program. Sebuah program yang ditulis oleh seorang *programmer* diterjemahkan ke dalam kode mesin oleh *compiler* atau *interpreter*. Hasil akhir dari kode mesin tersebut merupakan

jutaan sistem operasi, *website*, *game*, dan aplikasi yang ada saat tersebut. Faktanya, perangkat yang disebut "*smart*" berfungsi berkat kode program yang ada di dalamnya.

Hasil akhir dari penggunaan bahasa pemrograman adalah sistem operasi, aplikasi *desktop*, *website*, aplikasi *mobile*, bahkan berbagai perangkat teknologi yang biasa digunakan dalam kehidupan sehari-hari. Dengan bahasa pemrograman yang sudah dikuasai oleh *programmer* dengan baik, maka sebuah program aplikasi bisa dikembangkan fiturnya sesuai dengan kebutuhan pengguna.

6.7 Jenis Bahasa Pemrograman

Bahasa pemrograman akan ditinjau berdasarkan tingkat kedekatannya dengan logika komputer. Ada tiga jenis bahasa pemrograman yang akan dipahami, yakni bahasa pemrograman tingkat rendah, menengah dan tinggi (Ariffud Muhammad, 2022).

6.7.1 Bahasa Pemrograman Tingkat Rendah

Bahasa tingkat rendah merupakan jenis bahasa pemrograman yang paling erat hubungannya dengan logika komputer. Hanya ada dua bahasa yang masuk dalam kategori ini, yakni bahasa mesin dan bahasa *assembly*.

Bahasa mesin merupakan bahasa asli yang dipahami oleh komputer. Bahasa tersebut disebut juga kode mesin. Bahasa mesin umumnya dinyatakan sebagai kode biner (0 dan 1). Meski bahasa mesin cenderung sulit dipahami manusia, namun program yang ditulis dalam bahasa mesin tidak memerlukan proses penerjemahan untuk dijalankan di komputer sehingga menghasilkan kinerja yang sangat cepat.

Bahasa rakitan atau bahasa assembly merupakan salah satu jenis bahasa pemrograman tingkat rendah untuk membangun sistem yang bekerja langsung dengan perangkat keras. Misalnya dalam menulis sistem operasi dan driver. Proses penerjemahan tersebut dilakukan oleh alat yang disebut *assembler*. Meskipun proses penerjemahan sebelum eksekusi, kinerja bahasa *assembly* sangat cepat. Hal tersebut disebabkan program tersebut tidak membutuhkan banyak memori saat dijalankan.

6.7.2 Bahasa Pemrograman Tingkat Menengah

Bahasa Tingkat Menengah sering disebut bahasa semu atau pseudo-language. Bahasa tingkat menengah tersebut memiliki perintah seperti bahasa tingkat tinggi yang mudah dipahami oleh *programmer* disebabkan menggunakan perintah yang mirip dengan bahasa manusia, namun bahasa ini juga sebagai bahasa penghubung antara bahasa tingkat rendah dengan bahasa tingkat tinggi.

Kompleksitas bahasa tingkat menengah sangat tinggi, mendekati bahasa *assembly*. Bahasa tingkat menengah juga sering digunakan untuk membangun sistem operasi atau program yang digunakan untuk mengelola program lain. Contoh bahasa tingkat menengah antara lain C, C++, dan C#.

6.7.3 Bahasa Pemrograman Tingkat Tinggi

Bahasa tingkat tinggi yang menyerupai bahasa manusia ini, sangat direkomendasikan bagi semua orang yang ingin belajar pemrograman. Sedemikian dekatnya dengan bahasa manusia, sehingga tingkat kesulitannya relatif sangat rendah.

Bahasa tingkat tinggi sering dimanfaatkan untuk membangun website dan program aplikasi lain dengan sangat mudah. Hampir semua website dan program aplikasi yang

terpasang di setiap komputer dibuat menggunakan bahasa tingkat tinggi (Andri Permana Wicaksono, 2020).

Bahasa tingkat tinggi masih diklasifikasikan menjadi dua jenis: pemrograman prosedural dan pemrograman berorientasi objek. Pemrograman prosedural merupakan jenis bahasa pemrograman yang memecah instruksi program menjadi bagian-bagian yang lebih kecil. Bagian-bagian program disusun secara sistematis sehingga akan dijalankan dalam aplikasi sesuai kebutuhan operasional yang diinginkan.

Komputer mengeksekusi instruksi dalam urutan yang telah ditentukan dengan mengeksekusi setiap bagian dari program yang dipanggil. Kelebihan bahasa prosedural merupakan programmer lebih mudah memahami alur program. Contoh bahasa pemrograman prosedural antara lain Pascal, BASIC, dan FORTRAN.

Pemrograman berorientasi objek merupakan jenis bahasa pemrograman yang mengatur instruksi program ke dalam objek tertentu. Setiap objek memiliki propertinya masing-masing. Komputer menjalankan perintah yang ditentukan sesuai dengan properti masing-masing objek.

Keuntungan dari pemrograman berorientasi objek merupakan perubahan akan dilakukan dengan mudah tanpa menimbulkan kesalahan pada bagian lain dari program. Disamping kelebihan tersebut, OOP memiliki kelemahan yaitu sulitnya memahami alur program yang telah selesai. Banyak sistem dan program aplikasi yang ada saat tersebut tercipta berkat bahasa pemrograman tingkat tinggi.

DAFTAR PUSTAKA

- Andri Permana Wicaksono. 2020. *Pemrograman Berbasis Web*. Jember: Politeknik Negeri Jember.
- Appkey. 2023. *Perkembangan Bahasa Pemrograman Dulu dan Sekarang*. <https://markey.id/blog/development/bahasa-pemrograman#:~:text=Perkembangan%20Bahasa%20Pemrograman%20Dulu%20dan%20Sekarang%201%201.,Pemrograman%205%205.%20Bahasa%20Pemrograman%20yang%20Lazim%20Digunakan>
- Ariffud Muhammad. 2022. *Bahasa Pemrograman: Pengertian, Fungsi, Jenis, dan 15+ Contohnya!* <https://www.niagahoster.co.id/blog/bahasa-pemrograman/>
- Dimas Setiawan. 2018. *Dasar Pemrograman Komputer*. Madiun: Unipma Press.
- Gamal Thabron. 2022. *Pemrograman Dasar – Konsep, Logika, Prinsip & Langkah*. https://serupa.id/pemrograman-dasar-konsep-logika-prinsip-langkah/#Konsep_Dasar_Pemrograman
- Rendy Almaheri. 2023. *Sejarah dan Perkembangan Bahasa Pemrograman Jejak Evolusi Teknologi Komputer*. https://medium.com/@rendy_almaheri/sejarah-dan-perkembangan-bahasa-pemrograman-jejak-evolusi-teknologi-komputer-6d9e83417abf
- Rendi Juliarto. 2020. *Apa Itu Kotlin? Kenapa Kita Harus Mempelajari Kotlin?* <https://www.dicoding.com/blog/apa-itu-kotlin-kenapa-kita-harus-mempelajari-kotlin/>

- Rizka Lailatul Amalia. 2020. *Teori Dasar Bahasa Pemrograman C++*.
https://www.academia.edu/41589648/TEORI_DASAR_BAHASA_PEMROGRAMAN_C_
- Yuyun Khairunisa, Eri Mardiani, Nur Rahmansyah, Agus Ambarwari, Zuriati, Putri Ariatna Alia, Purwadi Budi Santoso, Nur Hayati, Erik Yohan Kartiko, Johan Suryo Prayogo, Yulifda Elin Yuspita, Agung Teguh Setyadi, Nurhafifah Matondang, Imanaji Hari Saye. 2023. *Dasar-Dasar Pemrograman*. Yogyakarta: Penamuda.
- Zaid Lee. 2022. *Pemrograman Dasar – Konsep, Logika, Prinsip & Langkah*.
<https://zaidlee-rabbaanii.blogspot.com/2022/09/pemrograman-dasar-konsep-logika-prinsip.html>

BAB 7

METODE ANALIS DATA

Oleh Mutia Delina

7.1 Pendahuluan

Analisis data memproses dan mengolah data menjadi informasi baru yang lebih mudah dipahami. Analisis ini sangat penting untuk memanfaatkan data sebagai dasar dalam pengambilan keputusan serta pengoptimalan proses. Dalam dunia bisnis, analisis data biasanya digunakan untuk mengidentifikasi peluang, memetakan risiko, serta meningkatkan performa bisnis. Sedangkan pada penelitian, analisis data digunakan untuk menyederhanakan ribuan bahkan jutaan data yang diperoleh. Misalnya sebuah penelitian komputasi fisika memperoleh satu juta data. Data-data ini tidak dapat langsung disajikan karena memerlukan ruang yang sangat besar dan serta sulit dipahami. Data tersebut akan memerlukan kertas yang sangat banyak jika langsung dicetak. Data-data sebaiknya diolah dan hasilnya disajikan dalam bentuk grafik.

Grafik merupakan alat bantu visual untuk menyajikan sekumpulan data sehingga mudah dipahami dengan cepat. Grafik yang dapat digunakan adalah histogram, diagram titik, diagram pie, diagram batang yang disesuaikan dengan kebutuhan.

Contoh:

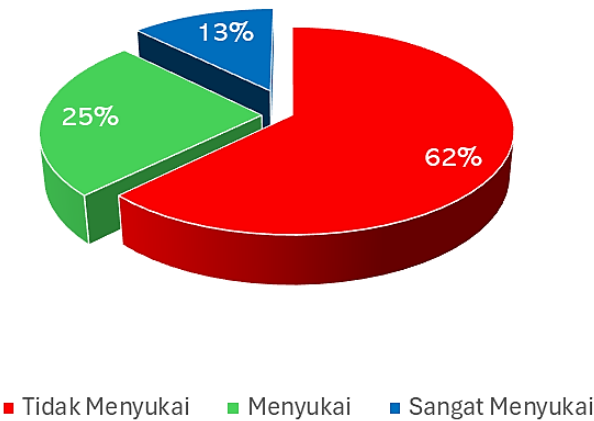
Sebagian besar mahasiswa tidak menyukai mata kuliah pemograman karena kesulitan memahami algoritma. Setelah dilakukan survei kepada 400 orang mahasiswa tahun pertama di

sebuah Universitas Negeri, diperoleh data 250 orang tidak menyukai, 100 orang menyukai dan 50 orang sangat menyukai mata kuliah pemograman. Agar mudah dipahami, data dapat diolah ke dalam bentuk presentase.

Tabel 7. 1. Penyajian data dalam tabel

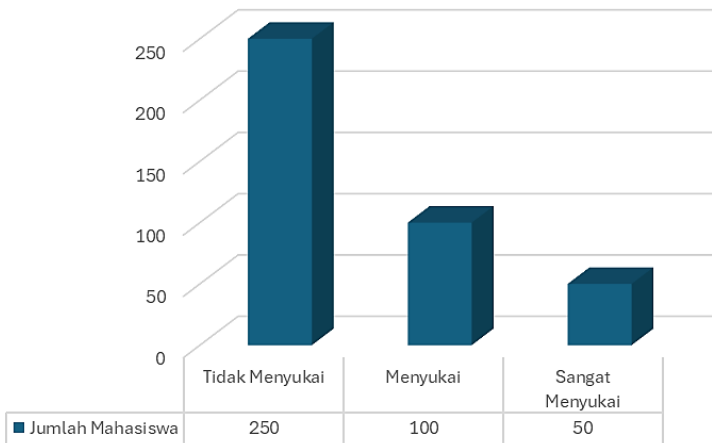
KATEGORI	JUMLAH MAHASISWA	PRESENTASE
Tidak Menyukai	250	62%
Menyukai	100	25%
Sangat Menyukai	50	13%
JUMLAH	400	100%

Data pada tabel 7.1 dapat disajikan dengan diagram pie untuk memudahkan pembaca melihat proporsi atau presentase terbesar dari hasil survei.



Gambar 7. 1. Contoh penyajian data dalam diagram Pie

Dengan bantuan diagram pie pada gambar 7.1., pembaca langsung dapat memahami bahwa sebagian besar mahasiswa responden tidak menyukai matakuliah pemograman. Data tersebut juga dapat di sajikan dalam diagram batang seperti yang ditunjukkan oleh gambar 9.2 berikut ini



Gambar 7. 2. Contoh penyajian data dengan diagram batang

Diagram batang pada gambar 7.2., menunjukkan bahwa mahasiswa yang tidak menyukai matakuliah pemograman adalah 250 orang dan menjadi jumlah tertinggi jika dibandingkan mahasiswa yang menyukai dan sangat menyukai matakuliah pemograman.

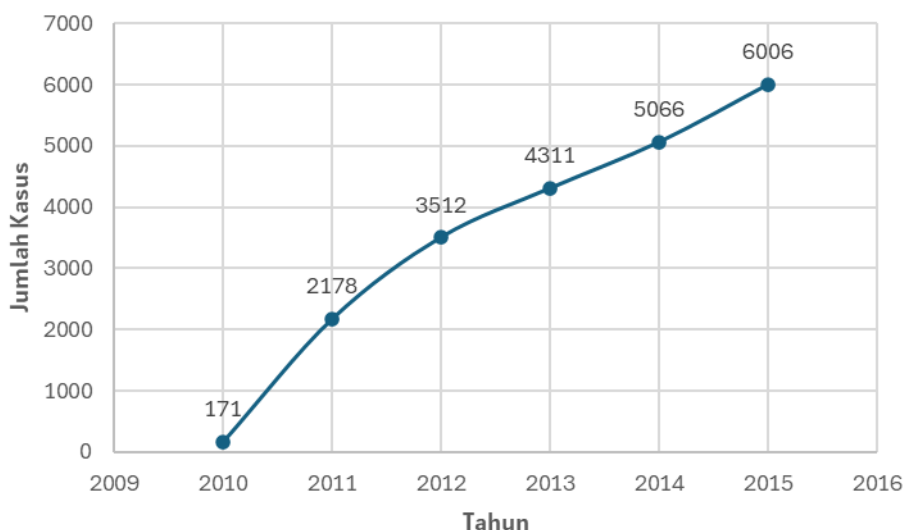
Selain digram pie dan digram batang, data juga dapat disajikan dalam diagram garis dan diagram titik. Perhatikan contoh berikut ini.

Menurut data KPAI P2TP2A Bukit Tinggi tahun 2010 sampai dengan 2015 kasus kekerasan terhadap anak mengalami peningkatan setiap tahunnya.

Tabel 7. 2. Jumlah kasus kerasan terhadap anak

Tahun	Jumlah Kasus
2010	171
2011	2178
2012	3512
2013	4311
2014	5066
2015	6006

Data pada tabel 7.2., dapat disajikan dalam diagram titik seperti yang ditunjukkan oleh gambar 7.3 berikut ini.



Gambar 7. 3. Contoh penyajian data dengan diagram titik atau scater plot

Sebelum disajikan dalam grafik data diolah dengan metode analisis data yang terdiri dari metode kualitatif dan metode kuantitatif (Mutia Delina, 2021).

7.2 Metode Kualitatif

Metode kualitatif merupakan teknik analisis untuk mengolah data kualitatif atau kategori (non-numerik) yang biasanya diperoleh dari hasil wawancara, literatur dan observasi. Data kualitatif merupakan informasi dalam berbentuk teks, kalimat verbal maupun narasi. Misalnya jenis kelamin (laki-laki dan perempuan) model rambut (lurus, ikal, keriting), jenis kelamin (laki-laki, perempuan). Metode kualitatif mengolah data-data tersebut secara mendalam serta berfokus pada analisis konseptual dengan memanfaatkan statistika deskriptif dan uraian yang lebih subjektif. Beberapa jenis teknik analisis data kualitatif antara lain analisis wacana, analisis naratif, dan analisis *Grounded Theory*.

Analisis wacana digunakan untuk memahami makna kontekstual dari kata, kalimat, atau bahasa. Analisis hanya fokus pada pemahaman responden pada topik penelitian. Sementara analisis Naratif digunakan untuk menjelaskan informasi yang diperoleh dari berbagai sumber, seperti wawancara, observasi dan survei. Sedangkan analisis *grounded theory* digunakan untuk mengembangkan teori baru dari data yang telah dikumpulkan. Pada analisis data kualitatif, data diklasifikasi, diinterpretasi dan dianalisa sehingga menghasilkan informasi yang dibutuhkan.

7.3 Metode Kuantitatif

Metode kuantitatif merupakan teknik analisis untuk mengolah data kuantitatif atau numerik (angka dan bilangan) yang dianalisis dengan statistika. Data kuantitatif biasanya diperoleh dari pengukuran misalnya panjang sebuah benda, kecepatan kendaraan, luas wilayah dan lain-lain. Metode kuantitatif membantu pengguna untuk memperoleh kesimpulan yang lebih terukur dan komprehensif. Hasil analisis kuantitatif disajikan dalam bentuk angka, diagram/grafik/gambar dan penjelasan yang lebih mudah dipahami. Metode kuantitatif dapat menginterpretasikan data kompleks yang jumlahnya sangat besar, seperti data raya. Hasil dari interpretasi data ini bergantung pada kepada proses mempersiapkan, mengolah, dan memvisualisasikan data dengan bantuan perangkat lunak atau *software* pendukung. Data yang jumlahnya sangat besar ini tidak dapat diolah secara manual karena akan membutuhkan waktu yang panjang. Dalam mengolah data, metode kuantitatif biasanya memanfaatkan statistika deskriptif dan statistik inferensia.

7.3.1 Statistika Deskriptif

Statistika deskriptif digunakan untuk mendeskripsikan penyebaran data tunggal serta melihat hubungan antar data dengan bantuan grafik, bagan, tabel dan diagram (Stewart, 2016). Metode yang sering digunakan dalam statistik deskriptif antara lain mean, median, modus, dan presentase.

Mean atau nilai rata-rata digunakan untuk mewakili sekumpulan data dengan sebuah nilai. Dengan demikian pembaca tidak harus melihat data satu persatu karena sudah diwakilkan oleh nilai rata-rata.

Contoh:

Sebuah sekolah taman kanak-kanak mengukur tinggi badan dari 10 orang siswanya. Hasil pengukuran ditunjukkan oleh tabel berikut ini

Tabel 7.3. Hasil pengukuran tinggi badan siswa

Siswa	Tinggi Badan (cm)
1	100
2	110
3	115
4	111
5	90
6	98
7	115
8	112
9	100
10	100
Jumlah	1051

Rata-rata tinggi siswa dapat dihitung dengan menjumlahkan total tinggi badan semua siswa kemudian dibagi dengan jumlah siswa. Dengan demikian rata-rata tinggi siswa sekolah taman kanak-kanak tersebut adalah 105,1 cm.

Sedangkan median merupakan nilai tengah dari sekumpulan data tunggal atau berkelompok yang sudah diurutkan dari nilai terkecil ke nilai terbesar. Mari kita gunakan data pada tabel 7.3. dengan mengurutkan data-data tersebut dari tinggi badan terendah ke tinggi badan tertinggi.

Tabel 7. 4. Data tinggi badan siswa yang telah diurutkan

Siswa	Tinggi Badan (cm)
1	90
2	98
3	100
4	100
5	100
6	110
7	111
8	112
9	115
10	115

Nilai tengah dari data diatas berada pada baris 5 dan 6. Dengan demikian nilai tengah dari data pada tabel 7.4. adalah 105.

Modus menunjukkan nilai paling sering muncul. Mari kita gunakan kembali data pada tabel 7.3.

Tabel 7. 5. Jumlah kemunculan (frekwensi) tinggi badan siswa

Tinggi Badan (cm)	Frekwensi
90	1
98	1
100	3
110	1

Tinggi Badan (cm)	Frekwensi
111	1
112	1
115	2

Dengan demikian nilai yang sering muncul pada tabel 9.5. adalah siswa dengan tinggi badan 100 cm. Selanjutnya presentase dapat memperlihatkan proporsi nilai dalam sebuah kelompok data.

Contoh:

Masyarakat Amerika mengalami peningkatan dalam penggunaan internet dari tahun ke tahun. Di tahun 2000, 53% masyarakat dewasa telah memanfaatkan internet. Uniknya data ini mengalami peningkatan di tahun 2014 menjadi 87% (Census., 2012).

7.3.1 Statistika Inferensia

Statistika inferensia menggunakan prinsip peluang untuk menarik kesimpulan, seperti analisis regresi, analisis faktor, dan analisis untuk data berkelompok (Healey, 2015).



Gambar 7. 4. Salah satu contoh statistika inferensia

Gambar 7.4. menunjukkan penelitian National Retail Federation kepada 8860 responden dewasa. Mereka ingin mengetahui apakah tv shopping mempengaruhi pemirsa untuk belanja produk perusahaan tertentu. Hasil penelitian menunjukkan 18% terpengaruh, 44% tidak terpengaruh dan 39% adalah pembeli tetap.

7.4 Prosedur Analisis Data

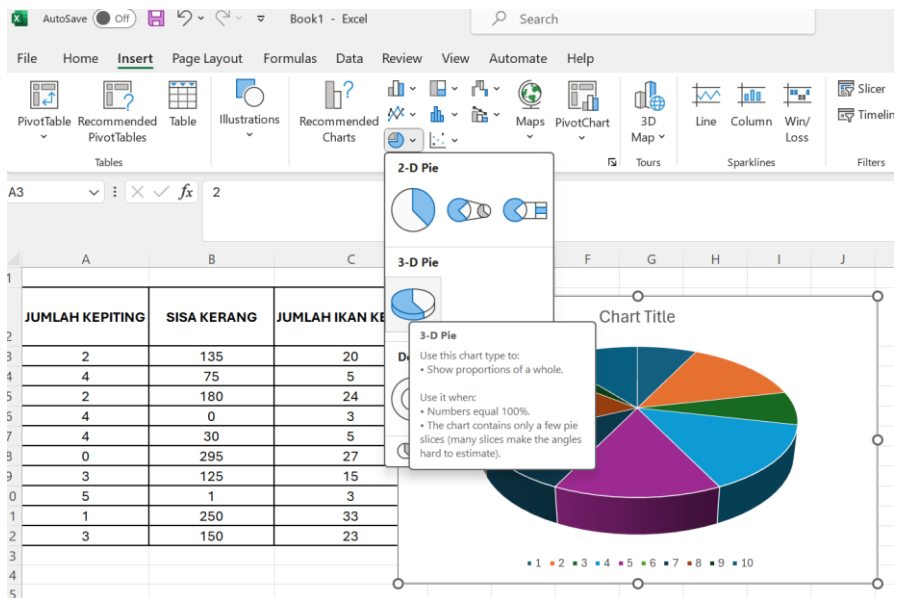
Analisis data merupakan proses yang sangat penting untuk mengolah data penelitian sehingga menjadi data yang mudah dipahami. Secara umum langkah-langkah yang digunakan dalam menganalisis data adalah sebagai berikut :

- Menentukan tujuan analisis data
- Menentukan cara pengumpulan data
- Mengumpulkan data
- Membersihkan atau membentuk data, atau tahap editing yaitu proses pemeriksaan

- Memeriksa kelengkapan data.
- Menganalisis data dengan metode kualitatif atau metode kuantitatif
- Interpretasi hasil analisis data dalam bentuk diagram, tabel atau grafik
- Menarik kesimpulan.

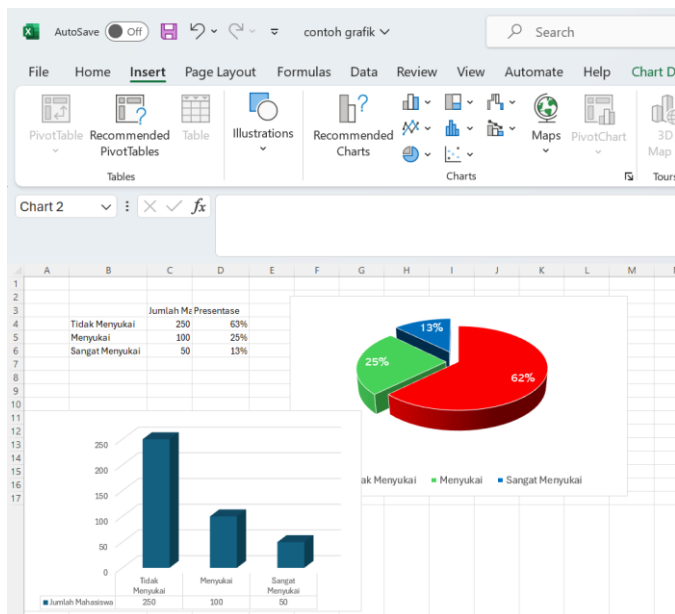
7.5 Pemanfaatan Tools untuk Analisis Data

Pada era digital, berbagai macam software atau perangkat lunak dapat dimanfaatkan untuk menganalisis data dan menyajikan hasil yang lebih mudah dipahami. Misalnya perangkat lunak Microsoft excel yang merupakan salah satu produk dari Microsoft office.



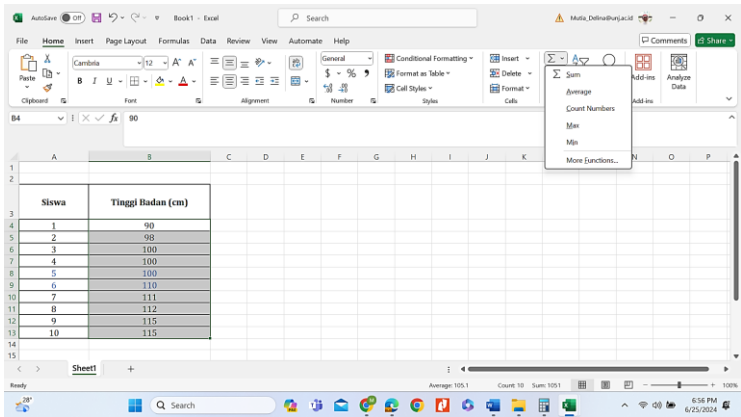
Gambar 7. 5. Membuat diagram dengan bantuan excel

Gambar 7.5. menunjukan salah satu keunggulan microsoft excel menyajikan data kedalam diagram.



Gambar 7. 6. Microsoft Excel menyediakan beberapa tipe diagram dan grafik.

Selain grafik, Microsoft Excel juga menyediakan beberapa *feature* untuk membantu mengolah data.



Gambar 7. 7. Feature pada Microsoft Excel untuk mengolah data.

DAFTAR PUSTAKA

- Census., U. B. (2012). *Statistical Abstract of the United States: 2012*.
- healey, J. f. (2015). *The essentials of statistics a tool for social research*. Boston: Cengage Learning.
- Mutia Delina, S. R. (2021). *Statistik untuk Bidang Sains*. Jakarta: Erlangga.
- Stewart, A. (2016). *Basic Statistics and Epidemiology A Practical Guide*. London: CRC Press Taylor and Francis Group.

BAB 8

MEMVISUALISASIKAN DATA

Oleh Elvi Rahmi

8.1 Pendahuluan

Visualisasi data adalah representasi grafis dari data atau informasi. Ini melibatkan penggunaan elemen visual seperti grafik, diagram, peta, dan tampilan visual lainnya untuk menggambarkan data dengan cara yang mudah dimengerti. Tujuannya adalah untuk mengungkapkan pola, tren, dan wawasan dari data yang mungkin sulit ditemukan dalam bentuk data mentah atau tabel angka. Visualisasi data memungkinkan kita untuk "melihat" data dan memahaminya dengan lebih baik.

Dalam data mining, visualisasi menjadi penting karena:

1. Pengungkapan wawasan
Visualisasi membantu mengungkapkan pola, tren, dan hubungan yang tersembunyi dalam data. Ini memudahkan pemahaman yang mendalam terhadap data.
2. Validasi model
Visualisasi digunakan untuk memvalidasi model-data mining dan mengidentifikasi anomali atau kesalahan dalam hasil prediksi.
3. Pengambilan Keputusan
Visualisasi membuat data lebih mudah dimengerti oleh pemangku kepentingan, membantu mereka membuat keputusan yang lebih baik dan lebih cepat.
4. Komunikasi efektif

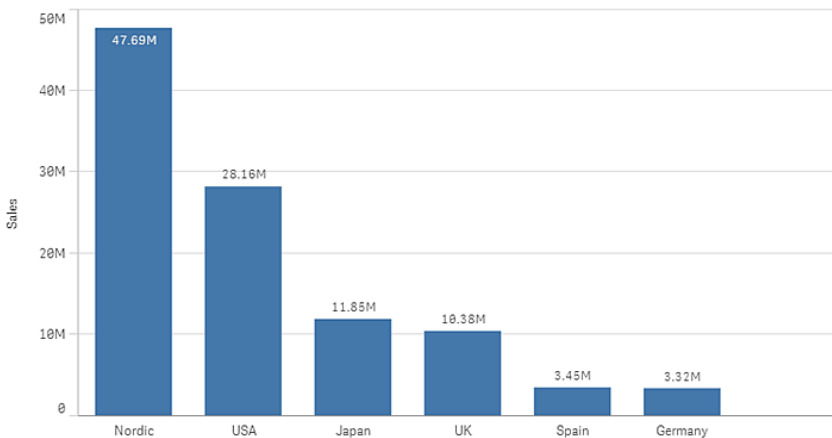
Dalam tim data mining, visualisasi memungkinkan komunikasi yang lebih efektif antara ahli statistik, analis data, dan pemangku kepentingan lainnya.

8.2 Jenis Chart/Grafik

Terdapat beberapa jenis chart/grafik untuk memvisualisasikan data, diantaranya:

8.2.1 Graphic Batang (*bar chart*)

Grafik batang digunakan untuk menampilkan data kategorikal atau diskrit dan merupakan pilihan yang baik untuk menunjukkan distribusi data antar kategori. Batang vertikal atau horizontal mewakili kategori, sementara panjang batang tersebut sesuai dengan nilai atau frekuensi data dalam kategori tersebut.

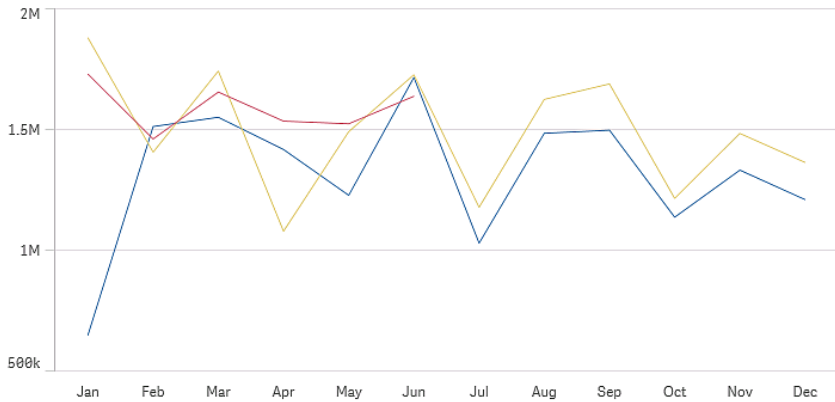


Gambar 8. 1. Contoh Bar Chart

Sumber: Data Olahan, 2024

8.2.2 Grafik garis (*line chart*)

Grafik garis digunakan untuk menunjukkan hubungan atau perubahan data seiring waktu atau dalam data berurutan. Ini menghubungkan titik data dalam urutan dengan garis yang halus, memudahkan pemahaman tren, fluktuasi, atau pola dalam data.

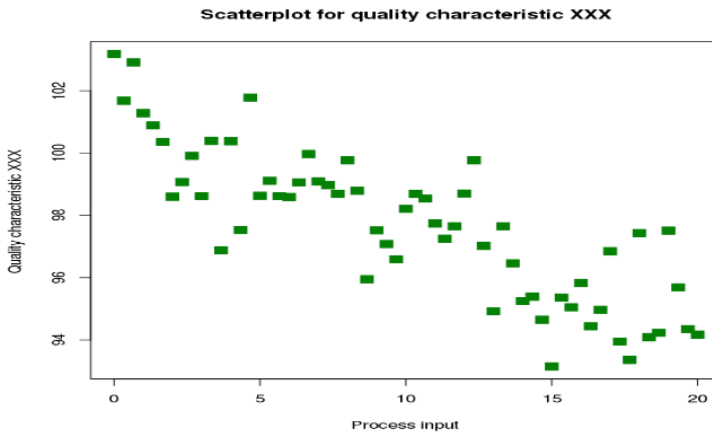


Gambar 8. 2. Contoh Line Chart

Sumber: Data Olahan, 2024

8.2.3 Grafik sebaran (*scatter plot*)

Grafik sebaran menggambarkan hubungan antara dua variabel numerik. Setiap titik dalam grafik mewakili satu data point dengan sumbu x dan y yang berbeda. Ini memungkinkan pengamatan tentang hubungan korelasi atau distribusi data.

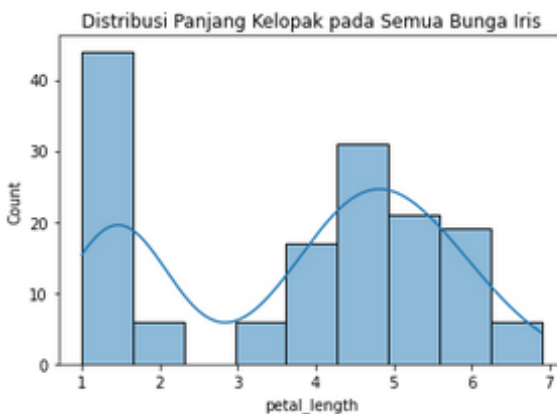


Gambar 8. 3. Contoh Scatter Plot

Sumber: Data Olahan, 2024

13.2.4 Histogram

Histogram digunakan untuk menggambarkan distribusi data numerik. Data dibagi menjadi interval atau bin, dan tinggi batang mewakili jumlah data yang jatuh ke dalam setiap interval. Ini membantu melihat pola distribusi data, seperti apakah data normal atau asimetris.

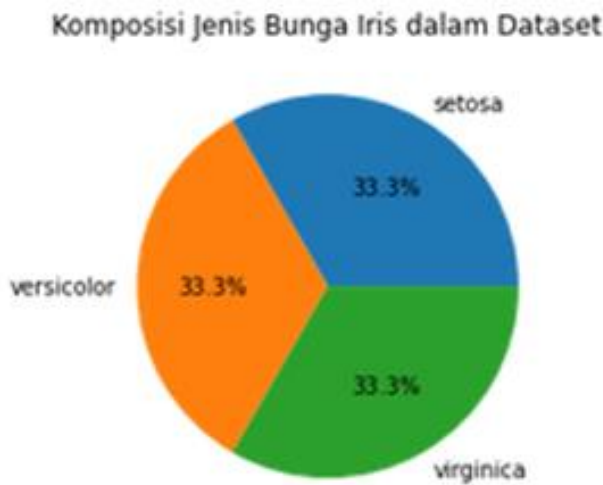


Gambar 8. 4. Contoh Histogram

Sumber: Data Olahan, 2024

8.2.5 Diagram lingkaran (*pie chart*)

Diagram lingkaran digunakan untuk menunjukkan bagian atau proporsi dari keseluruhan. Ini cocok untuk menunjukkan persentase komposisi dari data kategorikal. Setiap sektor melambangkan bagian dari keseluruhan, dan total sektor adalah 100%.

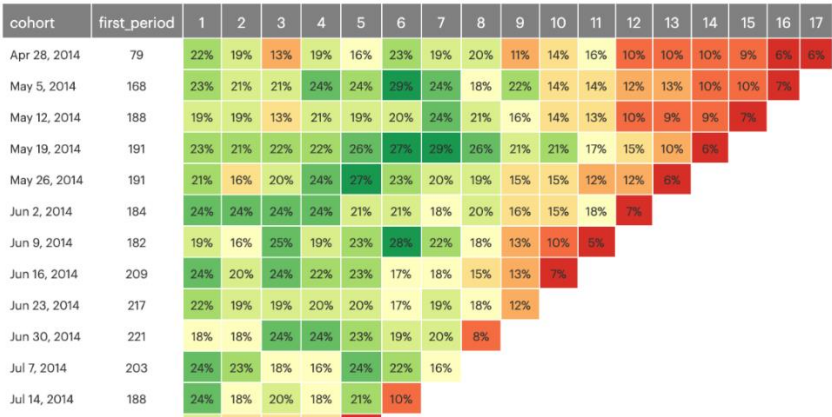


Gambar 8. 5. Contoh Pie Chart

Data Olahan, 2024

8.2.6 Heatmap

Grafik dua dimensi yang digunakan untuk mewakili data dalam bentuk matriks. Warna digunakan untuk mengindikasikan nilai dalam matriks, memudahkan identifikasi pola atau korelasi antara elemen matriks.

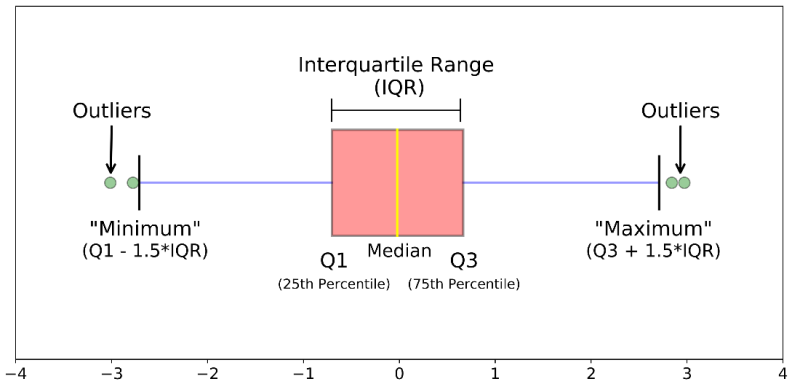


Gambar 8. 6. Contoh Heatmap

Sumber: <https://mode.com/example-gallery/heat-map/>, 2024

8.2.7 Diagram kotak (box plot)

Diagram kotak digunakan untuk menunjukkan statistik ringkasan dan distribusi data numerik. Ini mencakup kuartil, rentang interkuartil, pencilan (outliers), serta batas-batas data.

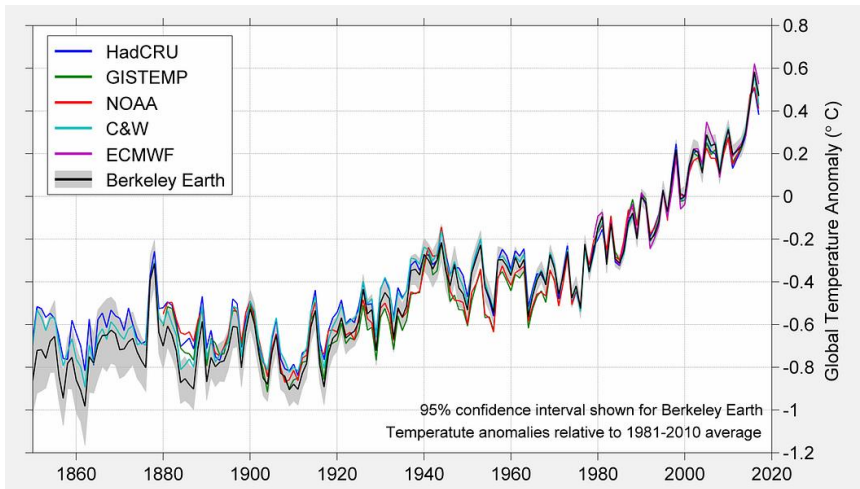


Gambar 8. 7. Contoh Box Plot

Sumber: Data Olahan, 2024

8.2.5 Diagram garis waktu (*time series plot*)

Diagram garis waktu digunakan untuk menggambarkan data dalam konteks waktu. Ini memudahkan pengamatan terhadap perubahan dalam data sepanjang waktu, seperti perubahan musiman atau tren jangka panjang.

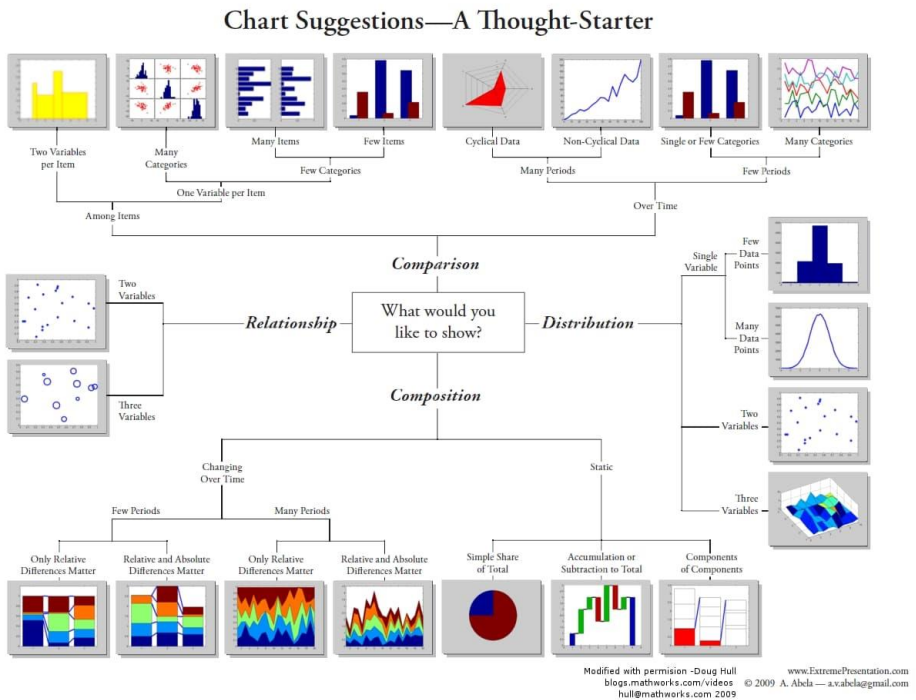


Gambar 8. 8. Contoh Time Series

Sumber: <https://towardsdatascience.com/>, 2024

8.3 Pemilihan Jenis Visualisasi

Selain jenis-jenis di atas, ada banyak jenis visualisasi data lainnya seperti scatter matrix, chord diagram, sankey diagram, dan banyak lagi. Pemilihan jenis visualisasi yang tepat tergantung pada karakteristik data yang kita miliki dan tujuan analisis kita. Salah satu panduan yang dapat membantu kita dalam memilih jenis chart/grafik adalah panduan bernama *Chart Selection Diagram* dari Dr. Andrew Abela.



Gambar 8. 9. Chart Selection Diagram dari Dr. Andrew Abela

Sumber: <https://apandre.wordpress.com/dataviews/choiceofchart/>,
2024

Panduan tersebut merupakan bagian dari metode Extreme Presentation. Chart Selection Diagram bisa membantu dalam memilih visualisasi data yang tepat berdasarkan tujuan, kesesuaian jenis chart/grafik, dan variabel data yang tersedia. Penggunaan Chart Selection Diagram bisa kita mulai dengan menjawab pertanyaan “Apa yang ingin kamu tunjukkan?”. Kemudian Chart Selection Diagram akan memberikan jalur cabang sesuai dengan tujuan dari informasi yang ingin disampaikan. Empat kategori tujuan yang terdapat dalam Chart Selection Diagram adalah: *Comparison* (perbandingan), *Relationship* (hubungan), *Composition* (komposisi), *Distribution* (distribusi).

8.3.1 Visualisasi Perbandingan

Comparison (perbandingan) berperan dalam membandingkan nilai antara variabel yang berbeda. Lebih dari itu, perbandingan juga dapat digunakan untuk mengidentifikasi nilai ekstrem dalam data (nilai tertinggi ataupun terendah dari data) dan untuk membandingkan berbagai kategori dalam suatu variabel. Beberapa pertanyaan yang dapat dijawab oleh diagram perbandingan adalah:

- a. Produk apa yang memiliki total penjualan tertinggi tahun ini?
- b. Bagaimana kondisi naik-turunnya tingkat penjualan produk selama 24 bulan terakhir?

Beberapa contoh pilihan jenis comparison chart antara lain bar chart, line chart, combo chart.

8.3.2 Visualisasi Hubungan

Relationship (hubungan) bermanfaat dalam menggambarkan cara variabel berinteraksi satu sama lain. *Relationship* Chart juga memungkinkan kita untuk mengidentifikasi korelasi, data yang berada di luar kebiasaan

(outliers), dan kelompok data (cluster). Beberapa pertanyaan yang dapat dijawab melalui diagram hubungan antara lain:

- a. Apakah terdapat korelasi/hubungan antara pengeluaran biaya iklan dan tingkat penjualan dari suatu produk?
- b. Bagaimana variasi dari pendapatan berdasarkan region?

Beberapa contoh pilihan *relationship chart* adalah *scatter plot* dan *bubble chart*.

8.3.3 Visualisasi Komposisi

Composition (komposisi) menggambarkan proporsi komponen tertentu terhadap total keseluruhan. Diagram komposisi dapat bersifat statis, yang menunjukkan komposisi nilai total saat ini, atau dapat juga menunjukkan perubahan dalam komposisi nilai total seiring berjalannya waktu. Contoh pertanyaan yang dapat dijawab melalui diagram komposisi adalah berapa persentase total penjualan berdasarkan region waktu tertentu?

Contoh pilihan jenis *composition chart* antara lain *pie chart* dan *waterfall chart*.

8.3.4 Visualisasi Distribusi

Distribution (distribusi) menggambarkan atau mengeksplorasi bagaimana nilai dalam suatu data ditempatkan dalam kelompok-kelompok. Diagram distribusi dapat memberikan gambaran tentang bentuk data, rentang nilai, dan kemungkinan adanya data yang berada di luar kebiasaan (*outlier*). Beberapa pertanyaan yang dapat dijawab oleh diagram distribusi adalah:

- a. Berapa jumlah pelanggan per kelompok umur?
- b. Kota mana yang paling banyak menggunakan layanan A?

Contoh pilihan jenis *distribution chart* antara lain histogram, *scatter plot*, *box plot*.

8.4 Alat dan Perangkat Lunak untuk Visualisasi Data

Visualisasi data memainkan peran penting dalam analisis data dan pengambilan keputusan. Berikut adalah beberapa alat populer yang digunakan untuk visualisasi data, beserta kelebihan dan kekurangannya.

8.4.1 Microsoft Excel

Microsoft Excel adalah alat spreadsheet yang banyak digunakan untuk analisis data dan visualisasi. Meskipun terutama dikenal untuk fungsi spreadsheet-nya, Excel juga memiliki berbagai alat visualisasi data seperti grafik batang, grafik garis, dan *pie chart*.

Kelebihan:

- a. Mudah digunakan: Antarmuka pengguna yang intuitif membuatnya mudah diakses oleh pengguna dari berbagai latar belakang.
- b. Ketersediaan luas: Excel adalah bagian dari paket Microsoft Office yang banyak digunakan di dunia bisnis dan pendidikan.
- c. Integrasi: Mudah diintegrasikan dengan produk Microsoft lainnya, seperti Word dan PowerPoint.

Kekurangan:

- a. Keterbatasan visualisasi: Alat visualisasi Excel mungkin terbatas untuk analisis data yang lebih kompleks.
- b. Kinerja: Bisa menjadi lambat saat menangani dataset yang sangat besar.

8.4.2 Tableau

Tableau adalah alat visualisasi data yang kuat yang digunakan untuk membuat visualisasi data interaktif dan dasbor. Alat ini populer di kalangan analis data untuk kemampuan visualisasinya yang ekstensif dan interaktif.

Kelebihan:

- a. Kemampuan interaktif: Pengguna dapat membuat dasbor interaktif yang memungkinkan eksplorasi data secara mendalam.
- b. Visualisasi yang kuat: Menyediakan berbagai jenis grafik dan visualisasi.
- c. Integrasi data: Mudah menghubungkan dan mengintegrasikan data dari berbagai sumber, termasuk file, basis data, dan layanan cloud.
- d. Program Akademik: Ada tawaran lisensi tidak berbayar untuk pelajar dan institusi akademik.

Kekurangan:

- Biaya: Jika tidak menggunakan versi *public*, ada biaya tahunan untuk masing-masing jenis lisensi Tableau yang harus dikeluarkan.
- Pembelajaran: Meskipun antarmuka pengguna ramah, fitur lanjutan mungkin memerlukan waktu untuk dipelajari.

8.4.3 Power BI

Power BI adalah alat visualisasi data dan *business intelligence* dari Microsoft. Ini memungkinkan pengguna untuk membuat laporan interaktif dan dasbor yang dapat dibagikan di seluruh organisasi.

Kelebihan:

- a. Integrasi dengan ekosistem Microsoft: Mudah diintegrasikan dengan alat Microsoft lainnya, seperti Excel dan Azure.
- b. Interaktivitas: Memungkinkan pembuatan visualisasi dan dasbor interaktif.
- c. Ketersediaan: Tersedia dalam berbagai paket, termasuk versi gratis dengan fitur dasar.

Kekurangan:

- a. Pembelajaran: Fitur lanjutan mungkin memerlukan waktu untuk dipelajari, terutama bagi pengguna baru.
- b. Keterbatasan versi gratis: Versi gratis memiliki keterbatasan dalam hal fitur dan kapasitas penyimpanan data.

8.4.4 Python (Matplotlib, Seaborn)

Python adalah bahasa pemrograman serbaguna yang digunakan secara luas dalam data science. Matplotlib dan Seaborn adalah dua pustaka utama di Python untuk visualisasi data.

Kelebihan:

- a. Fleksibilitas: Kemampuan tinggi untuk visualisasi yang kustom.
- b. Integrasi: Mudah diintegrasikan dengan ekosistem Python lainnya untuk data analysis (misalnya, Pandas, NumPy).
- c. Komunitas besar: Banyak sumber daya dan dukungan komunitas.

Kekurangan:

- a. Pembelajaran: Membutuhkan pengetahuan pemrograman, yang mungkin memerlukan waktu untuk dipelajari.
- b. Kinerja: Visualisasi interaktif tidak sekuat alat seperti Tableau.
- c. Kustomisasi: Memerlukan lebih banyak kode untuk visualisasi yang sangat kustom dibandingkan dengan alat yang lebih intuitif.

DAFTAR PUSTAKA

- Han, J., & Kamber, M. 2006. Data Mining: Concepts and Techniques. Cambridge: Morgan Kaufmann.
- Microsoft. (n.d.). Microsoft Excel features. Microsoft. <https://www.microsoft.com/en-us/microsoft-365/excel>.
- Waskom, M. (2021). Seaborn: Statistical Data Visualization. Seaborn Development Team. <https://seaborn.pydata.org/>
- Tableau Software. (n.d.). Why Tableau. Tableau Software. <https://www.tableau.com/why-tableau>
- Tufte, E. R. 2001. The Visual Display of Quantitative Information. Connecticut: Graphics Press.

BIODATA PENULIS



Roro Inda Melani, M.T., M.Sc.

Dosen Program Studi Teknik Informatika
Fakultas Sains dan Teknologi

Universitas Islam Maulana Malik Ibrahim Malang

Penulis lahir di Jakarta pada tanggal 25 September 1978. Penulis adalah dosen tetap pada Program Studi Studi Teknik Informatika Fakultas Sains dan Teknologi, Universitas Islam Maulana Malik Ibrahim Malang. Menyelesaikan pendidikan S1 pada Sistem Informasi di STIKI Malang dan melanjutkan studi S2 pada Jurusan Teknik Elektro, Jaringan Cerdas Multimedia di ITS Surabaya. Penulis menekuni bidang Pemrograman dan Interaksi Manusia dengan Komputer.

Pembaca yang ingin berkontak langsung dengan penulis bisa dilakukan melalui email roromelani.study@gmail.com

BIODATA PENULIS



Rizki Dewantara, S.Kom., M.Kom. , CCNA, MTCNA, MTCRE.

Dosen Program Studi Sains Data

Fakultas Sains Teknologi dan Kesehatan

Institut Teknologi Bisnis Dan Kesehatan Bhakti Putra Bangsa
Indoesia

Penulis lahir di Kelurahan Bukit Tunggal Kecamatan Jekan Raya Kota Palangka Raya Kalimantan Tengah, 28 Desember 1995, anak dari Iskandarsyah, SE., SH., MM. dan Wiwik Rukmiyati, SE. Saat ini bekerja sebagai Dosen tetap Sains Data di perguruan tinggi Institut Teknologi Bisnis dan Kesehatan Bhakti Putra Bangsa Indonesia di Kabupaten Purworejo, Jawa Tengah, selain sebagai dosen, juga aktif di dunia pergerakan dan organisasi. Dalam dunia pergerakan, penulis terlibat secara aktif dan memiliki pengalaman di Pergerakan Mahasiswa Islam Indonesia (PMII), Senat Mahasiswa UIN SUKA Yogyakarta periode 2015 – 2017, dan saat ini disamping dosen, bekerja sebagai instructor/trainer di bidang IT Network security management dengan spesialis di bidang : *Computer Network & Security, Big Data Security, Information Security Management, Cloud Computing and Virtualization*. Adapun riwayat pendidikan, Rizki menempuh kuliah S1 Teknik Informatika di Universitas

Islam Negeri Sunan Kalijaga Yogyakarta dan S2 Pascasarjana Magister Informatika Universitas Islam Negeri Sunan Kalijaga Yogyakarta. Memiliki individu yang ramah dan bersahabat dengan kemampuan membangun hubungan jangka panjang yang solid dengan orang lain. Kecerdasan sosial dan interpersonal yang sebanding dengan keterampilan menangani perangkat komputer baik software maupun hardware. Semangat, keteguhan dan kemampuan terbukti dengan baik melalui beberapa pencapaian, dengan memiliki sertifikasi di Bidang Cisco Certification Network Associate (CCNA), MikroTik Certified Network Associate (MTCNA), MikroTik Certified Routing Engineer (MTCRE), Certified Leadership Succession Planner (CLSP), Certified Performance Management Profesional (CPMP), Certified Profesional Risk Management (CPRM), Certified Profesional Leadership Management (CPLM), Certified Information Technology Governance Professional (CITGP), Certified Professional Digital Marketing (CPDM) dan Certified IT Services Management Leadership (CITSML). Semenjak 2022 mulai karirnya sebagai penulis karena memiliki hobi dengan menulis dan sekarang di sibukkan dengan mengajar, penelitian dan pengabdian, serta meluangkan dengan membaca dan menulis jurnal dengan menjadi *reviewer* dan *editor* di beberapa jurnal maupun menulis buku yang dapat diakses di https://linktr.ee/rizki_28 yang diharapkan dapat bermanfaat bagi pribadi dan orang banyak. Semoga penulis dapat melahirkan beberapa karya - karya dibidang IT lainnya. Terima kasih ^_^

BIODATA PENULIS



Reska Mayefis, S.Pd., M.Pd. T

Dosen Program Studi D3 Tata Boga

Fakultas Pariwisata dan Perhotelan Universitas Negeri Padang

Penulis lahir di Padang tanggal 07 Maret 1994. Penulis adalah dosen tetap pada Program Studi D3 Tata Boga Fakultas Pariwisata dan Perhotelan, Universitas Negeri Padang. Menyelesaikan pendidikan S1 pada Jurusan Pendidikan Teknik Informatika dan Komputer di Universitas Bung Hatta dan melanjutkan S2 pada Jurusan Pendidikan Teknologi dan Kejuruan di Universitas Negeri Padang.

BIODATA PENULIS



Sri Rezeki Candra Nursari

Staf Dosen Program Studi Teknik Informatika Fakultas Teknik
Universitas Pancasila

Penulis lahir pada tanggal 22 Juli 1966 di Surabaya, Jawa Timur. Penulis adalah dosen tetap pada Program Studi Teknik Informatika Fakultas Teknik Universitas Pancasila. Menyelesaikan pendidikan Srata 1 pada di Teknik Elektro IKIP PGRI Madiun, Jawa Timur. Gelar magister di Teknik Informatika STTI Benarif Jakarta. Penulis mengkhususkan diri dalam bidang Data Science, Artificial Intelligence dan Software Engineering.

BIODATA PENULIS



Salaki Reynaldo Joshua

Dosen Program Studi Teknik Informatika
Fakultas Teknik Universitas Sam Ratulangi

Penulis lahir di Manado tanggal 02 Oktober 1994. Penulis adalah dosen tetap pada Program Studi Teknik Informatika Fakultas Teknik, Universitas Sam Ratulangi. Menyelesaikan pendidikan S1 pada Program Studi Pendidikan Teknologi Informasi dan Komunikasi (S.Pd), Universitas Negeri Manado, S2 pada program studi Information Technology Management, Staffordshire University (M.Sc), dan melanjutkan S3 pada program studi Electronics, Information and Communication Engineering, Kangwon National University (Ph.D). Penulis menekuni bidang Informatika khususnya bidang Intelligent System, Digital Health, Data Analytics, Software Development dan AIoT.

BIODATA PENULIS

Dahlan Susilo, S.Kom., M.Kom.

Dosen Program Studi Informatika
Fakultas Sains, Teknologi dan Kesehatan
Universitas Sahid Surakarta

Penulis lahir di Sukoharjo tanggal 14 Januari 1967. Penulis merupakan dosen tetap pada Program Studi Informatika Fakultas Sains, Teknologi dan Kesehatan Universitas Sahid Surakarta. Menyelesaikan pendidikan S1 pada Jurusan Teknologi Industri Pertanian IPB lulus tahun 1990 dan melanjutkan S2 pada Jurusan Ilmu Komputer UGM lulus tahun 2002. Penulis mengajar sejak tahun 1993 hingga sekarang.

Penulis mengajar mata kuliah bidang komputer, meliputi pengantar teknologi informasi dan komunikasi, algoritma dan pemrograman, serta struktur data.

BIODATA PENULIS



Dr. Mutia Delina

Dosen Program Studi Fisika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Jakarta

MUTIA DELINA lahir di Jakarta pada bulan November 1980. Penulis mulai menempuh pendidikan di jurusan fisika Universitas Negeri Jakarta (UNJ) pada tahun 1999. Penulis melanjutkan pendidikan ke program Magister Ilmu Material di Universitas Indonesia pada tahun 2005. Setelah lulus program Magister, penulis diterima sebagai Dosen Fisika di Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) UNJ. Penulis mendapat beasiswa untuk melanjutkan studi ke program Doktor Computational Physics-University of Groningen (RuG) Belanda di bawah bimbingan Prof. Hans de Raedt pada tahun 2008. Setelah lulus program doktor penulis kembali aktif mengajar di Program Studi Fisika FMIPA UNJ. Mulai tahun 2021, selain mengajar penulis juga menjadi Tim Pengembang bidang Kerjasama dan Hubungan International di Kantor Wakil Rektor bidang Perencanaan dan Kerjasama, Universitas Negeri Jakarta.

BIODATA PENULIS**Elvi Rahmi, S.T., M.Kom.**

Dosen Jurusan Teknik Informatika
Politeknik Negeri Bengkalis

Penulis lahir di Pekanbaru tanggal 09 Mei 1987. Menyelesaikan pendidikan S1 pada Fakultas Informatika Universitas Telkom Bandung dan melanjutkan S2 pada Jurusan yang sama di UPI YPTK Padang. Penulis merupakan dosen tetap Politeknik Negeri Bengkalis. Mengampu mata kuliah Data Mining, Kecerdasan Buatan, Statistika dan Probabilitas, Etika Profesi, dan Bahasa Indonesia.

Sebelum bergabung menjadi dosen di Politeknik Negeri Bengkalis, penulis bekerja sebagai seorang News Anchor di TVRI Riau selama 10 tahun. Pengalamannya ini menjadikannya kaya akan beragam sudut pandang untuk pengolahan data dalam mencari wawasan baru.

Penulis memiliki Sertifikasi Professional di bidang kecerdasan buatan, yang dikeluarkan pada tahun 2022 oleh Duco Digital Training, Inggris serta Sertifikasi Professional di bidang Data Analytics, yang dikeluarkan tahun 2019 oleh Kementerian Komunikasi dan Informatika Republik Indonesia.