

MANAJEMEN DATA DAN ANALISIS SISTEM # 1



Penulis :

**Bambang Suhartono
Hero Wintolo
Imam Riadi
Joko Handoyo**

**Muhammad
Safiq Rosad
Tri Rochmadi**



Manajemen Data Dan Analisis Sistem # 1

Penulis

Bambang Suhartono

Hero Wintolo

Imam Riadi

Joko Handoyo

Muhammad

Safiq Rosad

Tri Rochmadi

PENERBIT:



UU No 28 tahun 2014 tentang Hak Cipta

Pasal 113

- 1) Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
- 2) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).
- 3) Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/ atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
- 4) Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp. 4.000.000.000,00 (empat miliar rupiah).

Manajemen Data Dan Analisis Sistem # 1

Tim Penulis:

Bambang Suhartono
Hero Wintolo
Imam Riadi
Joko Handoyo
Muhammad
Safiq Rosad
Tri Rochmadi

Desain Cover:

Sulaiman

Tata Letak:

Sulaiman

ISBN:

978-634-04-6525-9

Cetakan Pertama:

Desember, 2025

Hak Cipta 2025, Pada Penulis

Hak Cipta Dilindungi Oleh Undang-Undang

Copyright © 2025
by HADLA Media Informasi
All Right Reserved

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari Penerbit

PENERBIT:



Website: www.media.hadlacorp.com

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penyusunan buku *Manajemen Data dan Analisis Sistem* ini. Buku ini hadir sebagai upaya untuk memberikan pemahaman komprehensif mengenai bagaimana data dikelola secara efektif serta bagaimana analisis sistem dilakukan secara sistematis untuk mendukung proses pengambilan keputusan, pengembangan aplikasi, dan peningkatan kinerja organisasi.

Perkembangan teknologi informasi yang begitu pesat telah menempatkan data sebagai aset strategis yang menentukan keberhasilan operasional maupun arah strategi perusahaan. Di sisi lain, kemampuan melakukan analisis sistem yang tepat, terstruktur, dan berbasis kebutuhan menjadi semakin penting untuk menjamin solusi teknologi yang dihasilkan benar-benar relevan dan bermanfaat. Oleh karena itu, materi yang disajikan dalam buku ini disusun secara runtut, mulai dari konsep dasar manajemen data, metode pengolahan dan penyimpanan data, hingga teknik analisis sistem yang mencakup pemodelan proses, perancangan basis data, serta evaluasi sistem.

Buku ini diharapkan dapat menjadi referensi bagi mahasiswa, peneliti, maupun praktisi yang ingin memahami konsep dan praktik manajemen data serta analisis sistem secara lebih mendalam. Selain itu, penyajian contoh kasus dan ilustrasi di dalamnya juga

dimaksudkan agar pembaca dapat mengaitkan teori dengan situasi nyata yang sering ditemui dalam dunia teknologi informasi.

Penulis menyadari bahwa buku ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan karya ini di masa mendatang. Semoga buku ini dapat memberikan kontribusi yang berarti bagi pengembangan ilmu pengetahuan dan praktik manajemen data serta analisis sistem di berbagai bidang.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan motivasi dalam proses penyusunan buku ini. Semoga buku ini bermanfaat bagi para pembaca dan menjadi salah satu referensi yang berguna dalam memahami tantangan dan peluang di era digital saat ini.

Hormat kami,

Tim Penulis

DAFTAR ISI

KATA PENGANTAR.....	v
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB 1 Pengantar Data Dan Informasi	15
A. Pendahuluan.....	15
B. Sifat dan Karakteristik Data.....	1
C. Pengelolaan Data.....	9
D. Teknik mengelola dan analisis data.....	11
E. Konsep Informasi.....	18
F. Studi Kasus Smart Mosque.....	23
G. Konsep dasar data, Informasi.....	27
H. Kesimpulan	31
BAB 2 Pengembangan Basis Data (SR).....	32
A. Pendahuluan.....	32
B. Pengertian dan Tujuan Basis Data	34
C. Siklus Hidup Basis Data (<i>Database Life Cycle</i>).....	39
D. Implementasi Basis Data dalam Internet of Things	48
E. Kesimpulan	64

BAB 3 Penanganan Missing Value pada Basis Data	66
A. Pendahuluan	66
B. Konsep Dasar Missing Value	72
C. Dampak Missing Value pada Analisis Data	77
D. Strategi Penanganan Missing Value	78
E. Praktik Penanganan Missing Value di SQL dan Python	83
F. Alasan Penting Penanganan Missing Value	86
G. Kesimpulan	89
BAB 4 Administrasi Basis Data (Trigger dalam Basis Data)	92
A. Definisi dan Konsep Trigger	92
B. Jenis-Jenis Trigger	94
C. Cara Kerja dan Penempatan Trigger	96
D. Fungsi Trigger dalam Audit dan Logging	99
E. Automasi Audit dan Monitoring Basis Data	103
F. Contoh Implementasi Trigger pada Kesiapan DB Forensik	107
G. Kesimpulan	109
BAB 5 Pengantar Basis Data Terdistribusi	112
A. Pendahuluan	112

B.	Konsep Dasar Basis Data Terdistribusi.....	114
C.	Data dan Informasi Kerawanan Banjir.....	117
D.	Perancangan Basis Data Terdistribusi untuk Sistem Aplikasi Kerawanan Banjir	122
E.	Implementasi dan Pengelolaan Basis Data Terdistribusi	127
F.	Studi Kasus Analisis Spasial Tingkat Kerawanan Banjir di Kecamatan Cepu	128
G.	Tren dan Tantangan Masa Depan	131
H.	Kesimpulan	134
Bab 6	Business Intelligence	136
A.	Pendahuluan	136
B.	Konsep, Komponen, Tren, dan Inovasi BI.....	137
C.	Proses BI	144
D.	Penerapan BI dalam Organisasi	146
E.	Tantangan dan Isu Terkait BI.....	150
F.	Hubungan BI dengan Tata Kelola Data dan Strategi TI	151
G.	Kesimpulan	152
Bab 7	Forensik Basis Data	154
A.	Pendahuluan	154

B. Konsep Dasar Forensik Basis Data	158
C. Metode dalam Forensik Basis Data.....	163
D. Studi Kasus.....	169
E. Kesimpulan	173
BAB 8 Keamanan dan Optimalisasi Basis Data	174
A. Pendahuluan	174
B. Fondasi Basis Data Modern dan Model Data	176
C. Keamanan Basis Data: Autentikasi, Otorisasi, Enkripsi dan Integritas Data	179
D. Partisi Basis Data untuk Skalabilitas dan Kinerja.....	182
E. Optimalisasi Kinerja Query dan Indeks	187
F. Audit, Logging, dan Pemantauan Aktivitas Basis Data 191	
G. Arsitektur Ketersediaan Tinggi (High Availability) dan Pemulihan.....	194
H. Manajemen Kapasitas dan Penyetelan Server.....	197
Daftar Pustaka	200

DAFTAR GAMBAR

Gambar 1. 1. Sifat dan Karakteristik Data	7
Gambar 1. 2. Alur Pemrosesan Data (Smart Mosque).....	11
Gambar 1. 3. Teknik Pengelolaan dan Analisis Data	13
Gambar 1. 4. Tahapan Transformasi Data menjadi Informasi.....	23
Gambar 1. 5. Studi kasus: Smart Mosque.....	25
Gambar 1. 6. Siklus Pemrosesan Data Smart Mosque.....	26
Gambar 2. 1. Database Life Cycle.....	41
Gambar 2. 2. ERD pada basis data IoT: Smart Home berbasis Firebase	56
Gambar 4. 1. Trigger pada Basis Data MySQL	100
Gambar 4. 2. Log Google Drive	103
Gambar 4. 3. Flowchart Trigger untuk Monitoring	107
Gambar 6. 1. Siklus pengelolaan Business Intelligence	145
Gambar 6. 2. Visualiasi hasil data dengan penerapan business intelligence	151
Gambar 7. 1. Struktur Basis Data pada komputer.....	156
Gambar 7. 2. Contoh sertifikat CHFI.....	158
Gambar 7. 3. Perintah untuk merekam ukuran basis data dan tabel pada oracle basis data.....	161
Gambar 7. 4. Teknik Sinkronisasi Dua Database Server MelaluiInternet.....	162
Gambar 7. 5. Metode Forensik Basis Data	164
Gambar 7. 6. Contoh Dokumen Laporan Forensik.....	169
Gambar 8. 1. Gambaran Keamanan Basis Data.....	181
Gambar 8. 2. Kinerja Query dan Indeks	189

DAFTAR TABEL

Tabel 1. 1. Karakteristik Kualitas Data	5
Tabel 1. 2. Struktur Data	8
Tabel 1. 3. Contoh Studi Kasus pada Smart Mosque.....	9
Tabel 1. 4. Tabel Tantangan & Strategi Pengelolaan Data Modern	12
Tabel 1. 5. Data Terstruktur	14
Tabel 1. 6. Data Tidak Terstruktur.....	15
Tabel 1. 7. Integrasi Terstruktur & Tidak Terstruktur	16
Tabel 1. 8. Big Data dan NoSQL untuk Data Tidak Terstruktur ...	17
Tabel 1. 9. Analisis Gabungan untuk Insight Holistik	18
Tabel 1. 10. Tahapan Transformasi Data ke Informasi.....	21
Tabel 2. 1. Tujuan Basis Data (Ekspansi).....	38
Tabel 2. 2. Analisis Kebutuhan basis data IoT: Smart Home berbasis Firebase	52
Tabel 2. 3. Atribut pada basis data IoT: Smart Home berbasis Firebase	55
Tabel 2. 4. Implementasi basis data	58
Tabel 2. 5. Pengujian basis data IoT Smart Home berbasis Firebase	61
Tabel 2. 6. Pemeliharaan basis data IoT berbasis Firebase.....	63
Tabel 4. 1. Log Aktivitas Pengguna pada Moodle.....	102
Tabel 4. 2. Praktek Terbaik Penggunaan Trigger untuk Log	104
Tabel 4. 3. Peran Trigger dalam Automasi	105

Tabel 6. 1. Tools/ aplikasi Business Intelligence beserta feature	140
Tabel 6. 2. Penjelasan tahapan siklus Business Intelligence dalam studi kasus “Menganalisis perilaku belanja pelanggan, memprediksi produk laris, dan merancang promosi yang efektif”	147
Tabel 7. 1. Jenis log pada basis data	166
Tabel 7. 2. Perbedaan redo dan undo log.....	167
Tabel 7. 3. Kasus Serangan Berdasarkan Artikel.....	170
Tabel 8. 1. Perbandingan Antara Beberapa Model Data.....	180
Tabel 8. 2. Perbandingan Antara Berbagai Teknik Partisi Basis Data	185
Tabel 8. 3. Tabel Perbedaan Antara Audit Dan Logging Dalam Pengelolaan Basis Data.....	193

BAB 1

Pengantar Data Dan Informasi

A. Pendahuluan

Data, informasi, dan basis data menjadi komponen fundamental dalam berbagai bidang kehidupan. Data merupakan representasi fakta mentah yang belum diolah, sementara informasi adalah hasil pengolahan data yang memiliki makna dan nilai guna (Elmasri & Navathe, 2016). Data telah menjadi aset yang sangat berharga bagi organisasi, pemerintah, dan bahkan individu. Data bukan hanya sekadar informasi mentah, tetapi juga bahan baku untuk menciptakan wawasan yang dapat mendukung pengambilan keputusan yang lebih baik, lebih cepat, dan lebih efisien. Namun, untuk mengubah data menjadi informasi yang berguna, dibutuhkan pemahaman yang mendalam tentang sifat, karakteristik, serta cara pengelolaan data yang tepat. Buku ini dirancang untuk memberikan wawasan tentang bagaimana data dapat dikumpulkan, diolah, dan digunakan melalui berbagai sistem basis data dan teknologi informasi terkini.

Dengan fokus pada manajemen data, buku ini menyajikan berbagai konsep penting dalam dunia pengelolaan data, mulai dari perkembangan basis data, teknik pengelolaan data terstruktur dan tidak terstruktur, hingga penerapan *Business Intelligence* (BI) dan *Internet of Things* (IoT) dalam analisis data (Fitriani & Suryanegara, 2020). Setiap bab dalam buku ini disusun untuk memperkenalkan pembaca pada teori dasar yang mendasari pengolahan data, serta

memberikan contoh penerapan praktis yang relevan melalui studi kasus, seperti Smart Mosque, yang menggambarkan penerapan teknologi *Cyber-Physical Systems* (CPS) untuk mengelola data di ruang ibadah.

Melalui pendekatan yang menyeluruh, buku ini bertujuan untuk memberikan pemahaman yang mendalam mengenai data, informasi, dan basis data, serta bagaimana teknologi dapat digunakan untuk mengatasi tantangan dalam pengelolaan informasi di berbagai sektor kehidupan. Pembaca diharapkan dapat memahami bagaimana teknologi data dan sistem basis data yang efisien dapat membantu menciptakan keputusan yang berbasis bukti, yang pada akhirnya mendukung keberhasilan operasional dalam organisasi atau institusi.

B. Sifat dan Karakteristik Data

Sifat dan karakteristik data ini mempengaruhi bagaimana data dikumpulkan, disimpan, diolah, dan digunakan untuk menghasilkan informasi yang akurat, relevan, dan berguna bagi pengambilan keputusan. Kualitas data yang baik sangat bergantung pada seberapa baik data tersebut memenuhi beberapa prinsip dasar yang terkait dengan sifat dan karakteristiknya. Dengan memahami keduanya, organisasi dapat merancang sistem pengelolaan data yang efisien dan efektif, sehingga dapat menghasilkan informasi yang benar-benar bernilai.

Data pada dasarnya dapat dikategorikan berdasarkan bentuk dan struktur. Berdasarkan bentuknya, data terbagi menjadi data kuantitatif dan data kualitatif. Data kuantitatif berbentuk angka yang

dapat diukur atau dihitung, seperti data keuangan, statistik populasi, atau hasil pengukuran fisik lainnya. Data ini sering digunakan dalam analisis numerik untuk menggali pola atau tren yang terukur (Stair & Reynolds, 2018). Sebaliknya, data kualitatif berupa deskripsi atau atribut yang tidak bisa diukur dalam angka, seperti warna, jenis kelamin, atau opini. Data kualitatif ini lebih bersifat subjektif dan sering dianalisis secara kualitatif untuk memberikan pemahaman yang lebih mendalam mengenai fenomena yang diamati (Laudon & Laudon, 2020).

Dari sisi struktur, data dapat berupa data terstruktur atau data tidak terstruktur. Data terstruktur disimpan dalam format yang jelas, seperti tabel yang terdiri dari baris dan kolom, memudahkan proses pencarian dan manipulasi data. Data tidak terstruktur, seperti teks, gambar, atau video, memerlukan teknologi yang lebih canggih untuk diolah, seperti pengolahan bahasa alami atau analisis citra. Kedua jenis data ini memiliki peranannya masing-masing dalam pengolahan informasi, dan sering kali keduanya digunakan bersama-sama untuk memberikan gambaran yang lebih holistik (Elmasri & Navathe, 2016).

Berbicara mengenai karakteristik data, ada beberapa faktor yang sangat memengaruhi kualitas dan kegunaan data dalam pengelolaan informasi. Pertama, akurasi data adalah salah satu karakteristik yang paling fundamental. Data yang akurat mencerminkan kondisi yang sebenarnya tanpa adanya kesalahan atau distorsi. Ketepatan data sangat penting untuk menghasilkan informasi yang dapat dipercaya, terutama dalam konteks pengambilan keputusan yang membutuhkan

dasar yang valid. Tanpa akurasi, informasi yang dihasilkan akan dipertanyakan, yang tentunya dapat berdampak negatif pada keputusan yang diambil (Marakas & O'Brien, 2013).

Selain akurasi, kelengkapan data juga menjadi elemen penting. Data harus lengkap, artinya tidak ada bagian yang hilang atau terlewat, terutama data yang diperlukan dalam proses analisis atau pengambilan keputusan. Kelengkapan data akan memastikan bahwa informasi yang dihasilkan tidak hanya valid, tetapi juga menyeluruh, sehingga keputusan yang diambil akan lebih tepat dan berbasis bukti yang komprehensif (Stair & Reynolds, 2018).

Tabel 1. 1. Karakteristik Kualitas Data

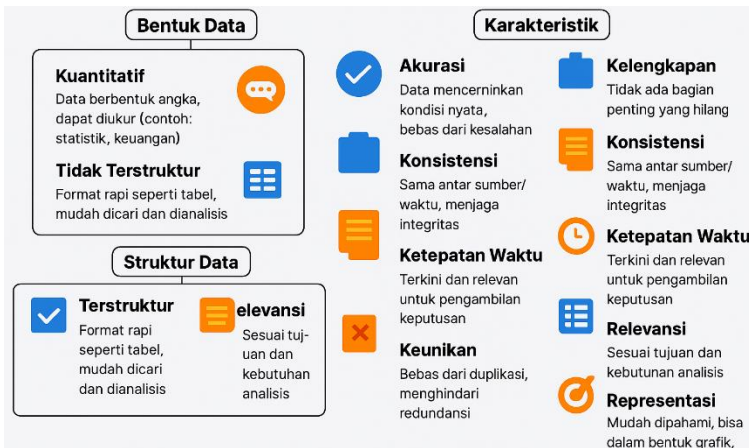
Karakteristik	Penjelasan	Contoh Implikasi Teknis
Akurasi	Data harus mencerminkan kondisi nyata, tanpa error atau bias.	Kalibrasi sensor akustik, inkronisasi clock perangkat DSP
Kelengkapan	Data tidak boleh ada bagian penting yang hilang.	Semua nilai parameter akustik harus direkam, seperti frekuensi respons dan SPL
Konsistensi	Format dan nilai data harus seragam antar waktu atau sumber.	Format konfigurasi data antar node IoT harus serupa agar interoperabilitas terjaga
Ketepatan Waktu	Data harus tersedia dan relevan saat dibutuhkan.	Stream audio real-time untuk kontrol speaker aktif di ruang ibadah
Validitas	Data harus sesuai dengan aturan atau logika sistem.	Format data sesuai protokol MQTT atau REST API dalam sistem CPS
Keunikan	Tidak ada data ganda, setiap entri	Identifikasi sensor dengan ID unik untuk pemetaan lokasi dan log

	harus dibedakan.	bisa data
Relevansi	Hanya data yang sesuai dengan tujuan yang dikumpulkan dan digunakan.	Data preferensi pengguna digunakan saat menyusun pengaturan mode audio khusus
Representasi	Data disajikan secara visual atau tekstual yang mudah dipahami.	Diagram distribusi suara, grafik historis frekuensi, atau dashboard antarmuka pengguna

Konsistensi data menjadi karakteristik yang sangat diperlukan dalam menjaga integritas informasi. Data yang konsisten berarti tidak ada perbedaan yang bertentangan dalam data yang seharusnya sama di berbagai sumber atau waktu. Ketidakkonsistenan dalam data dapat menyebabkan kebingungan dan pengambilan keputusan yang salah, yang mengarah pada hasil yang kurang optimal. Salah satu karakteristik penting lainnya adalah ketepatan waktu atau timeliness. Data yang tidak diperbarui atau sudah ketinggalan zaman tidak akan lagi memberikan nilai informasi yang relevan. Keakuratan waktu dalam penyajian data memungkinkan pengambilan keputusan yang cepat dan sesuai dengan kondisi terkini, yang sangat penting dalam konteks dunia bisnis atau manajemen yang dinamis.

Validitas data menjadi faktor berikutnya yang tidak kalah penting. Validitas mengacu pada apakah data yang terkumpul sesuai dengan aturan atau batasan yang telah ditentukan. Data yang valid menjamin bahwa data tersebut dapat dipertanggungjawabkan dan sesuai dengan standar yang telah ditetapkan, sehingga pengguna dapat menggunakan data tersebut dengan percaya diri dalam analisis

atau pembuatan keputusan. Data juga harus unik, artinya setiap data yang tercatat harus bebas dari duplikasi atau redundansi yang dapat membebani sistem dan menyulitkan proses pengelolaan data. Keunikan data memastikan bahwa informasi yang diproses adalah data yang relevan dan tidak mengandung entri yang tidak perlu. Ini membantu dalam meningkatkan efisiensi pengelolaan data dan mempermudah proses integrasi antar sistem.



Gambar 1. 1. Sifat dan Karakteristik Data

Relevansi data untuk memastikan bahwa data yang dikumpulkan sesuai dengan tujuan pengumpulan dan kebutuhan penggunaannya. Data yang relevan akan menghasilkan informasi yang efektif dan tepat sasaran, sedangkan data yang tidak relevan akan mengarah pada kesalahan dalam analisis dan keputusan yang tidak tepat (Laudon & Laudon, 2020). Terakhir, representasi data memainkan peran penting dalam bagaimana data tersebut disajikan kepada pengguna. Data harus bisa direpresentasikan dalam bentuk

yang mudah dipahami dan dapat dianalisis, apakah itu dalam bentuk grafik, tabel, atau visualisasi lainnya.

BAB 2

Pengembangan Basis Data (SR)

A. Pendahuluan

Sebelum munculnya basis-data, semua data yang dibutuhkan oleh program aplikasi disimpan dalam file (file-based) yang dirancang dan terkait dengan program aplikasi. Pendekatan file-based sebagai penyimpanan data ini menimbulkan banyak permasalahan, terutama yang mencakup redundansi data, inkonsistensi data, kesulitan dalam berbagi data, masalah keamanan, dan keterbatasan dalam pengelolaan data skala besar (Gordon, 2013). Sistem ini cenderung kurang fleksibel dan lebih rentan terhadap kesalahan manusia serta kesulitan dalam integrasi dengan sistem lain, untuk mengatasi masalah ini, basis data dikembangkan. Perbedaan utama file-based dan basis-data adalah bahwa sistem file digunakan untuk menyimpan informasi tentang file dan folder pada harddisk, sementara basis-data digunakan untuk menyimpan informasi yang terkait dengan basis data.

Inti dari sistem informasi modern, dituntut untuk dapat melakukan pengolahan data menjadi informasi yang bermanfaat untuk berbagai keperluan. Basis data (database) menjadi tulang punggung dalam berbagai sistem modern, mulai dari aplikasi bisnis,

e-commerce, hingga sistem informasi manajemen. Oleh karena itu, membangun aplikasi atau sistem informasi yang efektif, handal, scalable dan terstruktur memerlukan proses khusus yang melibatkan:

Penyimpanan Data: Data disimpan di media yang aman dan mudah diakses, seperti basis data relasional (MySQL, PostgreSQL), NoSQL (MongoDB), atau penyimpanan cloud (Firebase, Amazon S3).

Pengelolaan Data: Melibatkan validasi, normalisasi, dan pengaturan relasi antar data. Proses ini sering dijalankan melalui backend aplikasi menggunakan bahasa pemrograman seperti PHP, Python, Node.js, atau Java.

Pengambilan Data (Retrieval): Data yang sudah tersimpan diambil sesuai kebutuhan pengguna atau sistem melalui query dan API. Pengambilan data yang efisien sangat penting untuk performa aplikasi, terutama pada **Transformasi Menjadi Informasi:** Data mentah diolah menjadi informasi yang berguna, misalnya melalui laporan, dashboard, atau visualisasi yang membantu pengambilan keputusan.

Keamanan dan Konsistensi: Sistem harus memastikan integritas data, perlindungan terhadap akses tidak sah, serta audit trail untuk pelacakan.

Pengembangan basis data tidak hanya sebatas membuat tabel dan menyusun relasi antar data. Proses ini mencakup serangkaian tahapan sistematis yang dimulai dari analisis kebutuhan pengguna secara mendalam, perancangan model data yang komprehensif, penerapan normalisasi untuk memastikan integritas data, implementasi teknis dalam lingkungan *Database Management System* (DBMS), hingga tahap optimasi performa dan pengamanan data yang krusial. Setiap fase dalam siklus pengembangan ini

memerlukan tiga elemen fundamental: pemahaman konseptual yang mendalam tentang teori basis data, penguasaan keterampilan teknis yang relevan, serta kemampuan menerapkan pendekatan yang sesuai dengan karakteristik proyek. Keseimbangan ketiga aspek ini, sistem basis data dapat mencapai tingkat optimalisasi yang memadai sekaligus mampu memenuhi berbagai kebutuhan fungsional dan non-fungsional pengguna akhir.

B. Pengertian dan Tujuan Basis Data

Basis data merupakan sistem terstruktur yang menyimpan dan mengelola data secara elektronik, serta memungkinkan akses, manipulasi, dan pengelolaan data secara efisien sebagai kumpulan file sistem operasi, atau disimpan dalam DBMS seperti: MySQL, Oracle, PostgreSQL, atau MongoDB, yang membantu dalam penyimpanan, pengambilan, dan pengamanan data (Ramakrishnan et al., 2000). Basis data biasanya dikelola menggunakan Sistem Manajemen Basis Data (DBMS) seperti MySQL, Oracle, PostgreSQL, atau MongoDB, yang membantu dalam penyimpanan, pengambilan, dan pengamanan data.

Tujuan Basis Data adalah: Efisiensi Penyimpanan Data: Mengurangi redundansi (pengulangan data) dan memastikan konsistensi Data. Kemudahan Akses dan Pengelolaan: Memungkinkan pencarian, penyisipan, pembaruan, dan penghapusan data secara cepat dan terstruktur. Keamanan Data: Melindungi data dari akses tidak sah melalui mekanisme otentikasi dan enkripsi. Integritas Data: Memastikan data akurat, konsisten,

dan memenuhi aturan yang ditetapkan. Dukungan untuk Multi-User: Memungkinkan banyak pengguna mengakses data secara bersamaan tanpa konflik. Pemulihan Data (Recovery): Memiliki mekanisme backup dan restore untuk mengembalikan data jika terjadi kegagalan sistem. Penghematan Biaya dan Waktu: Mengurangi kebutuhan penyimpanan fisik (seperti kertas) dan mempercepat proses pengolahan informasi. Dengan basis data, organisasi atau individu dapat mengelola informasi secara lebih terstruktur, aman, dan efisien.

Data dalam basis data disusun secara sistematis dalam sistem komputer sehingga mudah dikelola, diperbarui, dan dicari sesuai kebutuhan. Memungkinkan pemeriksaan, pengelolaan, dan pengambilan data melalui antarmuka terstruktur (contoh: SQL). Konsep "terstruktur secara elektronik" merujuk pada penerapan (Santayana & Wilkes, 2004): *Metadata* (skema tabel, relasi), *Constraint* (validasi data), dan *ACID Properties* (konsistensi transaksi). Sistem ini dirancang untuk **meminimalkan redundansi** dan **memaksimalkan efisiensi akses** (Coronel et al., 2011). Kumpulan data terintegrasi ini, biasanya sangat besar sehingga harus disimpan pada perangkat penyimpanan sekunder seperti disk atau kaset.

Basis data menggunakan model struktural (relasional, hierarkis, NoSQL) untuk mengoptimalkan kecepatan operasi CRUD (*Create, Read, Update, Delete*). Contoh: Tabel dalam database relasional memanfaatkan *indexing* untuk pencarian milidetik (Elmasri & Navathe, 2007). Perancang dan mengelola sistem data dasar, ada

beberapa tujuan utama yang harus dicapai untuk memastikan kinerja, kerahasiaan data, dan Keamanan informasi yang optimal. Masing-masing tujuan seperti pada Tabel 2.1 memerlukan pendekatan teknis tertentu dan dapat diimplementasikan secara nyata dalam berbagai sistem.

Tabel 2. 1. Tujuan Basis Data (Ekspansi)

Tujuan	Keterangan Teknis	Contoh Implementasi
Akses Cepat	Penggunaan <i>B-tree indexing & hashing</i>	Pencarian data nasabah bank dalam 0.1 detik
Integritas	<i>Foreign key</i> dan <i>transaction logging</i>	Verifikasi otomatis nomor KTP yang unik
Skalabilitas	<i>Sharding</i> pada database terdistribusi	Sistem e-commerce mengelola 1M transaksi/hari
Keamanan	<i>Role-Based Access Control</i> (RBAC)	Dokter hanya bisa melihat rekam medis pasien

Pertama, **akses cepat** menjadi sangat krusial terutama dalam sistem yang membutuhkan waktu respon rendah. Untuk mencapainya, teknik seperti *B-tree indexing* dan *hashing* digunakan. Kedua metode ini memungkinkan sistem menemukan data dengan efisien tanpa harus memindai seluruh tabel. Sebagai contoh, pencarian data nasabah di sistem perbankan bisa dilakukan hanya dalam waktu 0.1 detik, yang sangat penting untuk layanan pelanggan yang cepat dan responsif.

Selanjutnya, **integritas data** dijaga melalui penggunaan *foreign key* dan *transaction logging*. Foreign key memastikan keterkaitan antar data antar tabel tetap konsisten, sedangkan transaction logging mencatat semua perubahan untuk memastikan data tetap utuh bahkan jika terjadi gangguan sistem. Implementasi nyatanya dapat dilihat pada proses verifikasi nomor KTP yang unik secara otomatis, untuk menghindari duplikasi data identitas. Untuk memenuhi tuntutan **skalabilitas**, terutama dalam sistem yang menangani volume data dan transaksi besar, digunakan pendekatan *sharding* pada database terdistribusi. Teknik ini membagi data ke dalam beberapa server atau node, sehingga beban tidak tertumpu pada satu tempat. Contohnya adalah sistem e-commerce besar yang mampu mengelola hingga satu juta transaksi per hari tanpa mengalami penurunan performa.

Kemudian, **keamanan** menjadi aspek yang tidak kalah penting, terutama dalam sistem yang mengelola data sensitif seperti rekam medis atau informasi keuangan. Penggunaan *Role-Based Access Control (RBAC)* memungkinkan sistem memberikan hak akses berdasarkan peran pengguna. Sebagai ilustrasi, dalam sistem informasi rumah sakit, hanya dokter yang memiliki akses untuk melihat rekam medis pasiennya sendiri, sehingga privasi pasien tetap terjaga.

Secara keseluruhan, kombinasi dari pendekatan teknis tersebut memberikan dasar yang kuat dalam membangun sistem basis data yang cepat, andal, skalabel, dan aman. Keberhasilan implementasi prinsip-prinsip ini sangat menentukan efektivitas suatu sistem

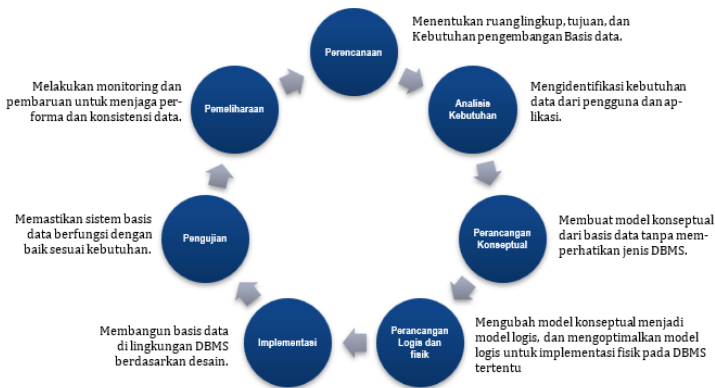
informasi dalam mendukung operasional organisasi secara menyeluruh.

C. Siklus Hidup Basis Data (*Database Life Cycle*)

Proses dalam tahapan pengembangan basis data terdiri dari beberapa fase berikutnya, Tahapan pengembangan basis data (merupakan) proses sistematis yang digunakan untuk merancang, membangun, dan mengelola sistem basis data secara efektif. Masing-masing fase memiliki tujuannya sendiri yang berasal dari penggabungan langkah-langkah yang diperlukan dalam pengembangan basis data, dimulai dari analisis kebutuhan dan diakhiri dengan pemantauan dan modifikasi. Siklus basis data tidak pernah berakhir karena pemantauan, modifikasi, dan pemeliharaan basis data merupakan bagian dari siklus, dan aktivitas ini terus berlanjut selama basis data tersebut hidup dan digunakan. Hal ini dinamakan dengan Database Life Cycle (DBLC) (Teorey et al., 2011).

Secara garis besar DBLC terdiri atas empat fase, yaitu: analisis kebutuhan, desain, implementasi, dan pemeliharaan (Teorey et al., 2011), namun ada yang menyatakan lima fase (Hoffer et al., 2019), dan enam fase (Corone & Morris, 2019). Hal ini menyoroti bagaimana proses atau langkah-langkah pengembangan basis data dapat berbeda istilah dan urutannya berdasarkan referensi yang digunakan, namun pada dasarnya memiliki inti aktivitas yang

serupa. Tujuan utama adalah menunjukkan bahwa, meskipun terdapat variasi dalam penamaan dan pengelompokan tahapan. Perbandingan ini membantu memahami bahwa pengembangan basis data merupakan proses bertahap yang sistematis, dengan setiap tahapan saling berkaitan untuk menghasilkan sistem basis data yang efektif dan efisien. Gambar 2.1 Merupakan DBLC yang diilustrasikan secara grafis, fase-fase pengembangan basis data yang menunjukkan diagram bentuk keluaran setiap langkah, sehingga perkembangan proses desain dari ide hingga implementasi basis data yang sebenarnya dapat terlihat.



Gambar 2. 1. Database Life Cycle

BAB 3

Penanganan Missing Value pada Basis Data

A. Pendahuluan

Manajemen basis data merupakan aspek penting dalam pengelolaan informasi di berbagai bidang, mulai dari bisnis hingga penelitian ilmiah. Salah satu tantangan utama dalam manajemen data adalah keberadaan *missing value* atau nilai yang hilang. *Missing value* dapat muncul karena berbagai alasan, seperti kesalahan pengumpulan data, responden yang tidak mengisi data, kesalahan teknis saat pengambilan data, atau gangguan teknis saat penyimpanan data, kesalahan manusia dalam pencatatan, kerusakan media penyimpanan data, hingga responden yang tidak memberikan jawaban dalam survei. Jika tidak ditangani dengan tepat, *missing value* dapat mengganggu validitas dan reliabilitas analisis data, serta menghasilkan kesimpulan yang bias. Oleh karena itu, pemahaman dan penanganan *missing value* menjadi sangat penting dalam proses manajemen data modern (Kim, 2021). Bab ini akan mengulas secara komprehensif konsep dasar *missing value*, dampaknya, strategi penanganan, serta praktik implementasinya menggunakan SQL dan Python.

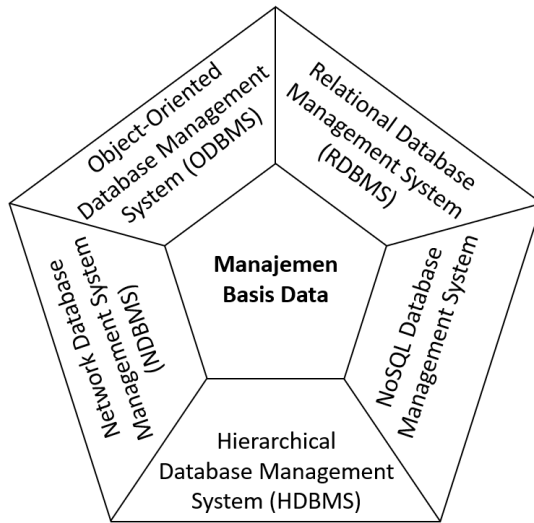
Manajemen akses dalam basis data berfungsi untuk mengatur siapa yang dapat mengakses informasi tertentu, serta apa saja yang boleh dilakukan oleh pengguna tersebut. Hal ini penting untuk menjamin bahwa hanya entitas yang sah yang dapat membaca, menulis, memperbarui, atau menghapus data tertentu. DBMS modern seperti PostgreSQL, Oracle Database, dan MySQL menyediakan skema manajemen hak akses berbasis *role-based access control* (RBAC), di mana hak akses pengguna diberikan berdasarkan peran tertentu dalam sistem. Misalnya, dalam sebuah

sistem pelaporan keuangan internal universitas, peran "admin" dapat memiliki akses penuh untuk semua tabel keuangan dan log aktivitas, sedangkan peran "dosen" hanya memiliki akses baca terhadap laporan pribadi dan tidak bisa mengubah data keuangan lainnya. Selain RBAC, beberapa sistem juga mengimplementasikan *mandatory access control* (MAC) atau *discretionary access control* (DAC) tergantung pada tingkat sensitivitas data.

Seiring meningkatnya kompleksitas dan jumlah data, muncul pula tantangan dalam hal keamanan, privasi, dan audit data. Data yang tidak dikelola dengan baik berisiko terhadap kebocoran, manipulasi, dan penyalahgunaan, yang tidak hanya menimbulkan kerugian finansial tetapi juga mengancam kedaulatan digital suatu negara. Oleh karena itu, kebutuhan akan manajemen basis data yang aman, andal, dan forensik-ready menjadi sangat mendesak.

Dalam konteks digital forensik, kemampuan untuk menelusuri siapa yang mengakses data tertentu menjadi sangat penting. Kasus pelanggaran data sering kali diawali dengan penyalahgunaan hak akses oleh pihak internal (*insider threat*). Oleh karena itu, sistem basis data harus dapat mengidentifikasi pengguna yang melakukan akses ilegal, baik secara sengaja maupun tidak disengaja. Misalnya, dalam investigasi pelanggaran data pasien di sistem rumah sakit digital, penyidik forensik dapat menganalisis histori akses data medis oleh pengguna non-otoritatif menggunakan log otentikasi dan audit.

1. Jenis-jenis Manajemen Basis Data



Gambar 3. 1. Jenis-jenis Manajemen Basis Data

a. Manajemen Basis Data Relasional (RDBMS)

Relational Database Management System (RDBMS) adalah jenis sistem manajemen basis data yang paling banyak digunakan. Sistem ini mengorganisasikan data dalam bentuk tabel yang saling berhubungan melalui kunci primer dan kunci asing. RDBMS menggunakan Structured Query Language (SQL) untuk operasi data. Keunggulan utama RDBMS adalah konsistensi data dan kemampuan untuk menangani hubungan kompleks antar data. Contoh implementasi RDBMS antara lain MySQL, PostgreSQL, Oracle Database, dan Microsoft SQL Server (Elmasri & Navathe, 2016).

b. Manajemen Basis Data NoSQL

NoSQL Database Management System, (NoSQL, Not Only SQL) dikembangkan untuk mengatasi keterbatasan RDBMS dalam menangani data tidak terstruktur dan kebutuhan skalabilitas horizontal. NoSQL menggunakan berbagai model data seperti dokumen (MongoDB), grafik (Neo4j), key-value (Redis), dan kolom lebar (Cassandra). Sistem ini cocok untuk aplikasi big data dan real-time web (Stonebraker & Çetintemel, 2005).

c. Manajemen Basis Data Hierarki (HDBMS)

Hierarchical Database Management System (HDBMS) mengorganisasikan data dalam struktur pohon dengan hubungan parent-child. Setiap child hanya memiliki satu parent, menciptakan struktur yang kaku namun efisien untuk data yang memang bersifat hierarkis. Contoh klasik HDBMS adalah IBM Information Management System (IMS) yang masih digunakan di beberapa sistem warisan (legacy systems).

d. Manajemen Basis Data Jaringan (NDBMS)

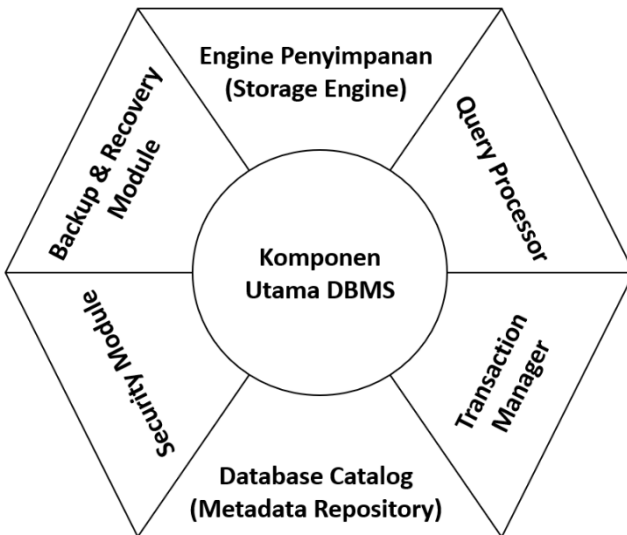
Network Database Management System (NDBMS) merupakan pengembangan dari model hierarki yang memungkinkan child memiliki multiple parents. Model ini menggunakan struktur grafik untuk merepresentasikan hubungan data yang kompleks. Integrated

Data Store (IDS) adalah salah satu implementasi awal dari model jaringan ini.

e. Manajemen Basis Data Objek (ODBMS)

Object-Oriented Database Management System (ODBMS) menyimpan data sebagai objek, mirip dengan konsep dalam pemrograman berorientasi objek. Sistem ini memungkinkan penyimpanan langsung objek kompleks beserta method-nya. ODBMS cocok untuk aplikasi yang membutuhkan manipulasi objek kompleks seperti CAD, sistem geografis, dan aplikasi multimedia.

2. Komponen Utama DBMS



Gambar 3. 2. Komponen utama DBMS

Komponen utama Database Management System adalah:

1. Engine Penyimpanan (Storage Engine)

Menyediakan struktur penyimpanan fisik, seperti file-file di disk atau penyimpanan cloud terenskripsi.

2. Query Processor

Menafsirkan dan menjalankan perintah SQL (Structured Query Language) atau bahasa lainnya.

3. Transaction Manager

Menangani eksekusi transaksi yang konsisten dan *atomic*; penting untuk audit trail dalam forensik digital.

4. Database Catalog (Metadata Repository)

Berisi informasi tentang struktur database (skema, tipe data, indeks, dsb).

5. Security Module

Mengatur hak akses, enkripsi data, autentikasi pengguna, dan logging aktivitas komponen esensial dalam investigasi digital.

6. Backup & Recovery Module

Menyediakan mekanisme pemulihan pasca kegagalan sistem, yang menjadi bukti digital penting dalam insiden forensik.

B. Konsep Dasar Missing Value

Missing value adalah data yang seharusnya ada dalam dataset tetapi tidak tersedia atau tidak tercatat. Dalam konteks database relasional, missing value seringkali direpresentasikan sebagai *NULL*. Dalam dataset statistik atau *machine learning*, missing value

harus diidentifikasi dan diolah agar tidak mengganggu proses analisis.

1. Jenis-jenis Missing Value

Penelitian oleh Little dan Rubin (2020) mengkategorikan missing value menjadi tiga jenis utama berdasarkan pola dan mekanismenya:

1. Missing Completely at Random (MCAR)

Data hilang secara acak tanpa pola atau hubungan dengan data lain. Contohnya adalah hilangkan data umur pada data mahasiswa.

2. Missing at Random (MAR)

Data hilang bergantung pada variabel lain yang tersedia dalam dataset. Misalnya, responden dengan umur tertentu lebih mungkin tidak menjawab pertanyaan tertentu.

3. Missing Not at Random (MNAR)

Data hilang terkait dengan nilai yang hilang itu sendiri. Misalnya, responden yang pendapatnya negatif tidak mengisi kuesioner secara sengaja.



Gambar 3. 3. Jenis-jenis Missing Value

Memahami jenis missing value penting untuk memilih metode penanganan yang tepat agar bias analisis bisa diminimalkan (Schafer & Graham, 2020).

a. Penyebab Utama Missing Value

1. Kesalahan Teknis dan Sistem

Missing value dapat muncul akibat kegagalan teknis seperti error dalam pengumpulan data, kegagalan penyimpanan data, korupsi data, atau masalah integrasi data dari beberapa sumber. Contohnya adalah kehilangan data karena proses transfer data yang tidak sempurna atau malfungsi perangkat lunak. (Little & Rubin, 2019)

2. Kesalahan atau Kelalaian Manusia

Dalam pengisian data manual atau input data, kesalahan manusia seperti lupa mengisi kolom tertentu, kesalahan pencatatan, atau pelaporan yang tidak akurat dapat menyebabkan missing value. Ini sering terjadi pada data survei atau entri data operasional. (Schafer & Graham, 2002)

3. Desain Instrumen Pengumpulan Data yang Kurang Optimal

Instrumen pengumpulan data, seperti kuesioner atau sistem pengisian data otomatis, jika tidak didesain dengan baik, misalnya memiliki terlalu banyak pertanyaan bersarang atau pilihan jawaban yang tidak mencakup semua kemungkinan akan menyebabkan responden melewati pengisian atau menolak menjawab pertanyaan tertentu. (Rubin, 1976)

4. Faktor Responden atau Sumber Data

Ada faktor yang berkaitan dengan responden seperti keengganan menjawab pertanyaan sensitif, ketidaktahuan responden, atau ketidakmampuan menjawab pertanyaan tertentu, yang menyebabkan nilai data kosong. Selain itu, dalam data operasional, kondisi sumber data yang tidak tersedia pada waktu pengambilan data juga bisa menyebabkan missing value. (Van Buuren, 2018)

5. Data Tidak Masuk Secara Acak (Missing Not at Random - MNAR)

Seringkali missing value muncul bukan secara acak tetapi terkait dengan nilai yang hilang itu sendiri. Contohnya, orang dengan pendapatan rendah mungkin tidak bersedia melaporkan pendapatannya, sehingga data pendapatan yang hilang tersebut memiliki pola tersembunyi yang harus diperhatikan secara khusus. (Rubin, 1976; Schafer & Graham, 2002)

b. Peran Manajemen Basis Data dalam Penanganan Missing Value

Missing value atau nilai yang hilang dalam basis data merupakan masalah umum yang dapat mempengaruhi kualitas data dan hasil analisis. Manajemen basis data memainkan peran penting dalam mengidentifikasi, mengelola, dan menangani missing value

untuk memastikan integritas data (Batini et al., 2016). Peran sistem manajemen basis data (DBMS) dalam penanganan missing value dan dampaknya terhadap kualitas data sebagai berikut:

1) Identifikasi Missing Value

DBMS menyediakan mekanisme untuk mendeteksi missing value melalui validasi data saat input (constraints, triggers), pemeriksaan integritas referensial, analisis metadata dan statistik data (García et al., 2020).

2) Pengelolaan Missing Value

DBMS modern menyediakan berbagai pendekatan untuk menangani missing value, diantaranya default value dan automatic imputation, Integrasi dengan algoritma machine learning untuk imputasi (Little & Rubin, 2020), mekanisme transaction logging untuk pelacakan perubahan data.

3) Dampak pada Kualitas Data dan Analisis

Penanganan missing value yang tepat oleh DBMS memberikan manfaat, yaitu meningkatkan akurasi analisis data (Zhang et al., 2020), mengurangi bias dalam pengambilan keputusan, memenuhi standar data governance dan compliance.

C. Dampak Missing Value pada Analisis Data

Penanganan missing value yang kurang tepat dapat menyebabkan berbagai masalah serius dalam analisis data, diantaranya:

1. Bias dalam estimasi parameter jika data hilang tidak secara acak (MNAR). Penghapusan data secara langsung dapat menyebabkan

estimasi parameter menjadi bias karena data yang dihapus memiliki pola tertentu (Jakobsen et al., 2019).

2. Pengurangan ukuran sampel penghapusan data yang memiliki missing value dapat mengurangi jumlah sampel yang tersedia secara signifikan, yang berakibat pada menurunnya kekuatan statistik (statistical power).
3. Kesimpulan yang tidak valid hasil analisis yang didasarkan pada data yang tidak lengkap atau tidak diolah dengan benar bisa menghasilkan kesimpulan yang salah atau menyesatkan.
4. Kesulitan dalam proses prediksi model prediktif seperti regresi atau machine learning memerlukan dataset yang lengkap agar bisa berfungsi dengan baik. Missing value yang tidak diatasi akan mengurangi kinerja model.

D. Strategi Penanganan Missing Value

Penanganan missing value merupakan tahapan penting dalam proses analisis data akademik, karena nilai yang hilang dapat memengaruhi validitas dan reliabilitas hasil penelitian. Missing value terjadi ketika data untuk suatu variabel tidak tercatat, dapat disebabkan oleh kesalahan pengisian, responden yang tidak menjawab, atau masalah teknis lainnya. Oleh karena itu, strategi yang tepat untuk menangani missing value sangat diperlukan agar data tetap representatif dan analisis tetap valid. Berikut adalah strategi penanganan missing value:

1. Penghapusan Data (Listwise atau Case Deletion). Menghapus seluruh pengamatan yang memiliki missing value pada variabel

manapun. Strategi ini mudah diterapkan namun berisiko mengurangi ukuran sampel secara signifikan sehingga menurunkan power statistik (Field, 2018).

2. Penghapusan Variabel (Variable Deletion). Menghapus variabel yang memiliki persentase missing value sangat tinggi. Cocok apabila variabel tersebut tidak krusial untuk analisis.
3. Imputasi Sederhana. Mengganti missing value dengan nilai rata-rata, median, atau modus. Mudah diaplikasikan, namun dapat mengurangi variabilitas data dan menyebabkan bias (Hair et al., 2019).
4. Imputasi Lanjutan (Multiple Imputation). Menggunakan model statistik untuk mengestimasi nilai yang hilang dengan mempertimbangkan hubungan antar variabel. Metode ini lebih akurat dan mempertahankan variabilitas data (Jumintoko & Sutiman, 2022).
5. Model Berbasis Machine Learning. Menggunakan algoritma seperti Decision Tree, Random Forest, atau k-Nearest Neighbors untuk memprediksi nilai yang hilang berdasarkan pola data lain (Jom FTI Budiluhur, 2022). Cocok untuk dataset yang besar dan kompleks.
6. Analisis Sensitivitas. Melakukan analisis tambahan untuk melihat dampak dari berbagai strategi imputasi terhadap hasil penelitian.

1. Contoh Kasus

Contoh Kasus adalah Model Adaptif Penanganan Missing Value pada Data Keuangan di Perguruan Tinggi.

Permasalahan missing value sangat krusial dalam data keuangan perguruan tinggi yang biasanya berisi informasi seperti pembayaran kuliah, dana hibah, biaya operasional, dan lain-lain. Data keuangan yang hilang atau tidak lengkap dapat menyebabkan kesalahan dalam perencanaan anggaran,

BAB 4

Administrasi Basis Data (Trigger dalam Basis Data)

A. Definisi dan Konsep Trigger

1. Pengertian Trigger

Dalam sistem basis data, trigger merupakan obyek basis data yang terdiri atas sekumpulan perintah SQL yang secara otomatis dijalankan ketika terjadi peristiwa (*event*) dari sebuah tabel atau objek yang diinginkan. Trigger ini sudah ada sejak adanya basis data relasional (Ceri et al., 2024), namun penggunaan trigger masih jarang dimaksimalkan oleh beberapa programmer atau *database administrator* (Davidson & Davidson, 2021). Event dari trigger dapat terjadi pada operasi *Insert*, *Update*, atau *Delete* dari suatu data (Shaik & Chemuduru, 2023). Trigger bekerja otomatis dan reaktif dari peristiwa yang terjadi pada perubahan data di basis data (bin Uzayr, 2023), bukan terjadi karena perintah pengguna (*user*).

2. Komponen Dasar Triger

Sebuah trigger umumnya memiliki tiga komponen utama:

- a. Event
Menentukan aksi apa yang memicu trigger, misalnya *After Insert*, *Before Update*, atau *After Delete*.
- b. Timing
 - *Before*: dijalankan sebelum aksi data dilakukan.
 - *After*: dijalankan setelah aksi data selesai dilakukan.
 - *Instead of*: menggantikan operasi yang akan dilakukan (khusus untuk *view* di beberapa *database management systems* (DBMS)).
- c. Action
Kode SQL atau blok perintah yang dijalankan ketika trigger aktif. Biasanya berupa perintah *Insert*, *Update*, *Delete*, atau bahkan pemanggilan prosedur lain.

3. Fungsi dan Tujuan Penggunaan Trigger

Trigger digunakan dalam berbagai skenario penting, antara lain:

- Menjaga konsistensi data, secara otomatis memperbarui atau memvalidasi data antar tabel.
- Audit trail (Li et al., 2024a), melacak semua perubahan yang terjadi dalam tabel (siapa, kapan, dan apa yang diubah).
- Otomatisasi proses, seperti menghitung ulang total, *update* stok barang, atau kirim notifikasi setelah data ditambahkan.
- Penegakan aturan bisnis, memastikan data yang dimasukkan mematuhi logika atau aturan tertentu, di luar batasan (*constraint*) biasa (Davidson & Davidson, 2021).

2. Kapan Trigger Harus Digunakan?

Trigger sangat berguna ketika:

- Aksi yang ingin dilakukan harus konsisten untuk semua perubahan data.
- Dibutuhkan *log* atau catatan otomatis untuk audit (Li et al., 2024a).
- Ingin meminimalisir penulisan ulang logika dalam berbagai aplikasi (Davidson & Davidson, 2021).

Namun, penggunaan trigger harus dilakukan dengan bijak. Trigger yang terlalu banyak atau kompleks dapat memengaruhi performa basis data, menyulitkan *debugging* dan rawan kesalahan dengan berbagai alasan dan kondisi (de Brock, 2023).

B. Jenis-Jenis Trigger

Trigger dalam basis data memiliki beberapa jenis utama yang dibedakan berdasarkan waktu pemicu (*timing*) dan aksi (*event*) yang menyertainya. Memahami perbedaan antara jenis-jenis trigger ini sangat penting untuk merancang sistem basis data yang efisien dan terkontrol. Berikut adalah jenis-jenis trigger dalam basis data:

1. Trigger Before

Trigger Before dijalankan sebelum operasi *Insert*, *Update*, atau *Delete* dilakukan pada tabel.

Fungsi Umum:

- Memvalidasi atau memodifikasi data sebelum disimpan.
- Menolak operasi tertentu berdasarkan kondisi logis.

Contoh:

```
CREATE TRIGGER cek_nilai_before
BEFORE INSERT ON nilai
FOR EACH ROW
BEGIN
    IF NEW.nilai < 0 OR NEW.nilai > 100 THEN
        SIGNAL SQLSTATE '45000' SET
        MESSAGE_TEXT = 'Nilai harus antara 0 - 100';
    END IF;
END;
```

Trigger ini akan menggagalkan penyisipan data jika nilai tidak berada dalam rentang valid.

2. Trigger After

Trigger After dijalankan setelah operasi *Insert*, *Update*, atau *Delete* selesai dilakukan pada tabel.

Fungsi Umum:

- Mencatat *log* aktivitas.
- Memperbarui tabel lain berdasarkan perubahan.
- Menjalankan audit atau pelacakan jejak data.

Contoh:

```
CREATE TRIGGER log_delete_after
AFTER DELETE ON pengguna
FOR EACH ROW
BEGIN
    INSERT INTO log_pengguna (id_pengguna,
waktu_hapus)
VALUES (OLD.id, CURRENT_TIMESTAMP);
END;
```

Trigger ini otomatis mencatat penghapusan data pengguna ke tabel log setelah proses *Delete*.

3. Trigger Instead Of

Trigger *Instead of* digunakan menggantikan operasi *Data Manipulation Language* (DML) yang dimaksud. Trigger ini umum digunakan pada *View*, karena *view* biasanya tidak bisa dimodifikasi secara langsung.

Keterangan:

- Umumnya digunakan di SQL Server dan Oracle.
- Menyediakan fleksibilitas agar *view* bisa dipakai seperti tabel biasa.

Contoh:

Misalnya kita memiliki *view* gabungan dari dua tabel:

```
CREATE VIEW v_data_mahasiswa AS
SELECT s.id, s.nama, k.nama_kelas
```

```
FROM mahasiswa s
JOIN kelas k ON s.id_kelas = k.id;
```

View `v_data_mahasiswa` tidak bisa diubah langsung. Maka kita bisa buat *instead of* trigger:

```
CREATE TRIGGER insert_v_data_mahasiswa
INSTEAD OF INSERT ON v_data_mahasiswa
FOR EACH ROW
BEGIN
    INSERT INTO siswa (id, nama, id_kelas)
    VALUES (NEW.id, NEW.nama,
            (SELECT id FROM kelas WHERE nama_kelas =
NEW.nama_kelas));
END;
```

Trigger ini memungkinkan kita melakukan *Insert* ke *view*, tetapi sebenarnya yang dimasukkan adalah ke tabel mahasiswa.

C. Cara Kerja dan Penempatan Trigger

Untuk memahami efektivitas penggunaan trigger, kita perlu mengetahui bagaimana trigger bekerja secara internal dan bagaimana cara penempatan yang tepat di dalam sistem basis data.

1. Cara Kerja Trigger

- a. Eksekusi Otomatis oleh Sistem
Trigger tidak dipanggil secara eksplisit oleh pengguna atau aplikasi. Ia dipicu secara otomatis saat terjadi aksi *Insert*, *Update*, atau *Delete* pada tabel atau *view* yang sudah dikaitkan dengannya (Petride et al., 2023).
- b. Bagian dari Transaksi

Trigger dijalankan sebagai bagian dari satu transaksi. Jika ada kesalahan dalam pelaksanaan trigger, maka seluruh transaksi bisa dibatalkan (*rollback*) (Wu et al., n.d.).

3. Urutan Eksekusi

Jika terdapat beberapa trigger pada satu tabel dan *event* yang sama, maka sistem basis data biasanya:

- Menjalankan trigger sesuai urutan dibuatnya.
- Memberi opsi untuk mengatur urutan dengan perintah khusus (tergantung DBMS, contoh: *FOLLOWS* atau *PRECEDES* di MySQL 8.0).

2. Penempatan Trigger

a. Trigger pada tabel

Trigger paling umum diletakkan pada tabel, misalnya untuk:

- Validasi nilai sebelum dimasukkan.
- Sinkronisasi data antar tabel.
- *Logging* perubahan data penting.

Contoh:

```
CREATE TRIGGER validasi_email
BEFORE INSERT ON pengguna
FOR EACH ROW
BEGIN
    IF NEW.email NOT LIKE '%@%' THEN
        SIGNAL SQLSTATE '45000' SET
MESSAGE_TEXT = 'Format email tidak valid';
    END IF;
END;
```

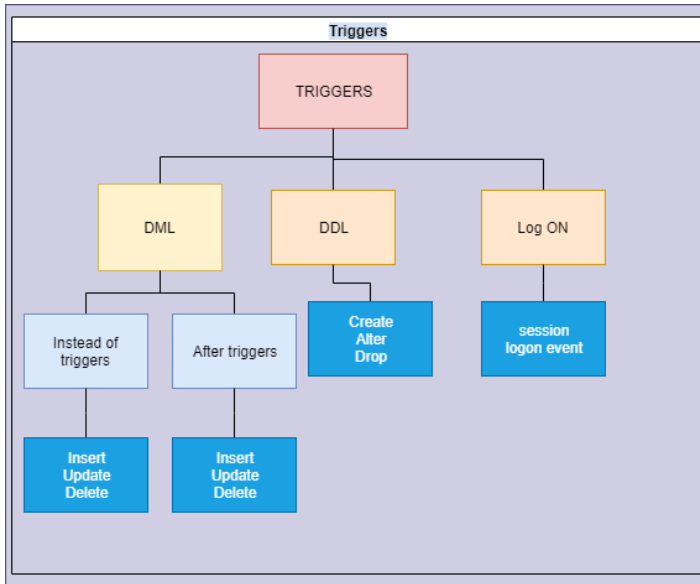
b. Trigger pada view (*instead of*)

Trigger ini digunakan untuk memungkinkan modifikasi view yang kompleks.

Contoh:

```
CREATE TRIGGER tr_update_view_mahasiswa
INSTEAD OF UPDATE ON view_mahasiswa
AS
BEGIN
    UPDATE mahasiswa
    SET nama = inserted.nama
    FROM mahasiswa
    JOIN inserted ON siswa.id = inserted.id;
END;
```

Secara garis besar, trigger dapat dijalankan pada operasi *Data Manipulation Language* (DML), *Data Definition Language* (DDL), maupun operasi *Log On* atau *Session*, seperti pada Gambar 4.1 (Maurice, 2022). Beberapa aplikasi atau sistem informasi yang sudah menerapkan *security by design*, trigger selalu diterapkan dan dijadikan sebuah *log* untuk mencatat semua informasi penting dan yang dirasa krusial jika terjadi perubahan data.



Gambar 4. 1. Trigger pada Basis Data MySQL

D. Fungsi Trigger dalam Audit dan Logging

Salah satu kekuatan utama penggunaan trigger dalam administrasi basis data adalah kemampuannya untuk membantu dalam audit data (Li et al., 2024a) dan *logging* otomatis (Rifqi et al., 2024). Audit merupakan proses penting dalam sistem informasi, terutama yang berkaitan dengan data sensitif, keuangan, akademik, maupun legal. Audit adalah proses pencatatan semua aktivitas atau perubahan yang dilakukan terhadap data di dalam sistem basis data

BAB 5

Pengantar Basis Data Terdistribusi

A. Pendahuluan

Banjir, sebagai salah satu bencana alam yang paling sering melanda berbagai penjuru dunia, termasuk Indonesia, membawa dampak destruktif yang signifikan terhadap tatanan kehidupan. Lebih dari sekadar genangan air, banjir mampu melumpuhkan denyut nadi aktivitas sosial-ekonomi masyarakat, menghancurkan infrastruktur vital seperti jalan, jembatan, dan bangunan, serta yang paling tragis, merenggut nyawa manusia (Smith, 2020). Oleh karena itu, pengelolaan risiko banjir yang efektif menjadi imperatif untuk meminimalisir kerugian dan melindungi masyarakat. Upaya ini memerlukan pemahaman komprehensif mengenai akar permasalahan banjir, termasuk faktor-faktor meteorologis seperti curah hujan ekstrem dan perubahan iklim, serta faktor hidrologis dan antropogenik seperti kondisi topografi, tata guna lahan yang tidak tepat, dan sistem drainase yang buruk (Kundzewicz et al., 2014).

Lebih lanjut, pengelolaan risiko banjir yang optimal tidak dapat dilepaskan dari ketersediaan informasi yang akurat dan tepat waktu. Pemantauan kondisi lingkungan secara real-time, prediksi kejadian banjir dengan tingkat akurasi yang tinggi, serta penyebaran informasi yang cepat dan efektif kepada pihak-pihak terkait, termasuk pemerintah daerah, badan penanggulangan bencana, dan masyarakat umum, merupakan elemen-elemen krusial dalam mengurangi risiko dan dampak banjir (UNISDR, 2015). Dalam konteks ini, sistem informasi memainkan peran sentral dan krusial. Sistem ini berfungsi sebagai tulang punggung dalam mengumpulkan data dari berbagai sumber, menyimpannya secara terstruktur, memproses dan menganalisis data tersebut menjadi informasi yang

bermakna, serta menyajikannya dalam format yang mudah dipahami oleh para pemangku kepentingan terkait kerawanan banjir.

Namun demikian, pengelolaan data banjir menghadirkan tantangan yang tidak sedikit. Karakteristik data banjir yang masif dalam volume, sangat beragam dalam jenisnya (mencakup data spasial seperti peta ketinggian dan citra satelit, data temporal seperti data curah hujan dan tinggi muka air sungai, serta data atribut seperti informasi demografis dan infrastruktur) (Handoyo et al., 2025), dan tersebar di berbagai lokasi geografis, menjadi kendala signifikan apabila dikelola menggunakan sistem basis data terpusat tradisional (Elmasri & Navathe, 2016). Keterbatasan dalam hal skalabilitas, ketersediaan, dan kinerja seringkali muncul ketika berhadapan dengan volume dan kompleksitas data banjir yang terus meningkat.

Menyadari keterbatasan sistem basis data terpusat, konsep dan teknologi basis data terdistribusi menawarkan solusi yang menjanjikan untuk mengatasi tantangan pengelolaan data banjir. Dengan mendistribusikan penyimpanan data dan beban pemrosesan ke beberapa *node* atau *server* yang saling terhubung melalui jaringan, sistem basis data terdistribusi mampu meningkatkan secara signifikan skalabilitas sistem untuk menangani volume data yang besar, meningkatkan ketersediaan data dan layanan meskipun terjadi kegagalan pada salah satu *node*, serta meningkatkan kinerja aplikasi kerawanan banjir melalui pemrosesan data yang paralel dan lebih dekat dengan sumber data (Özsu & Valduriez, 2020). Artikel ini bertujuan untuk mengeksplorasi secara mendalam bagaimana konsep dan teknologi basis data terdistribusi dapat

diimplementasikan secara efektif untuk membangun sistem aplikasi kerawanan banjir yang handal, responsif, dan mampu mendukung pengambilan keputusan yang tepat dalam upaya mitigasi dan penanggulangan bencana banjir di Indonesia.

B. Konsep Dasar Basis Data Terdistribusi

Basis data terdistribusi merupakan paradigma inovatif dalam pengelolaan data modern, yang secara fundamental berbeda dari model basis data terpusat tradisional. Inti dari konsep ini adalah penyimpanan data yang tidak lagi terpusat pada satu server tunggal, melainkan tersebar dan direplikasi di berbagai komputer independen, yang dikenal sebagai *node*, dan saling terhubung melalui infrastruktur jaringan komunikasi yang andal (Yuliansyah & Hermawan, 2014). Meskipun secara fisik data terfragmentasi dan tersebar, sistem basis data terdistribusi dirancang sedemikian rupa agar pengguna memiliki ilusi transparansi lokasi. Artinya, pengguna dapat mengakses, memanipulasi, dan berinteraksi dengan data seolah-olah seluruhnya tersimpan dalam satu entitas basis data logis yang terpadu (Tanenbaum & Van Steen, 2007).

Arsitektur basis data terdistribusi menunjukkan variasi yang signifikan, mencerminkan beragam kebutuhan aplikasi dan karakteristik data yang berbeda-beda. Beberapa arsitektur yang umum dijumpai meliputi model klien-server, di mana satu atau lebih server bertindak sebagai penyedia data dan layanan kepada banyak klien; arsitektur *peer-to-peer*, di mana setiap *node* memiliki kemampuan yang setara dalam menyediakan dan mengonsumsi

data dan layanan; serta arsitektur multi-database, yang mengintegrasikan beberapa sistem basis data otonom yang berbeda menjadi satu sistem logis (Date, 2000). Pemilihan arsitektur yang tepat menjadi krusial dan sangat bergantung pada persyaratan spesifik aplikasi, termasuk volume data yang diharapkan, tingkat konkurensi pengguna, kebutuhan akan ketersediaan dan toleransi kegagalan, serta pertimbangan geografis dari distribusi data.

Implementasi basis data terdistribusi menawarkan serangkaian keuntungan substansial dibandingkan dengan sistem terpusat (Hermawan, 2008). Peningkatan skalabilitas menjadi salah satu keunggulan utama, di mana sistem dapat dengan mudah mengakomodasi pertumbuhan *volume* data dan peningkatan jumlah pengguna dengan menambahkan lebih banyak *node* ke dalam jaringan tanpa memerlukan *downtime* yang signifikan (Stonebraker & Hellerstein, 2005). Ketersediaan data yang lebih tinggi juga terjamin, karena kegagalan pada satu atau beberapa node tidak akan menyebabkan seluruh sistem menjadi tidak berfungsi, mengingat data seringkali direplikasi di beberapa lokasi. Selain itu, kinerja aplikasi dapat ditingkatkan secara signifikan melalui distribusi beban pemrosesan data ke berbagai node, memungkinkan eksekusi kueri secara paralel dan mengurangi latensi waktu respons, terutama untuk aplikasi dengan data yang tersebar secara geografis. Terakhir, setiap node dalam sistem terdistribusi dapat memiliki tingkat otonomi lokal dalam mengelola sebagian data yang menjadi tanggung jawabnya, memberikan fleksibilitas dan kontrol yang lebih besar kepada pemilik data di lokasi yang berbeda.

Meskipun menawarkan banyak keunggulan, implementasi dan pengelolaan basis data terdistribusi juga menghadirkan tantangan yang perlu diatasi dengan cermat. Kompleksitas dalam manajemen transaksi yang melibatkan banyak node, memastikan konsistensi data di seluruh sistem (kontrol konkurensi), dan mengimplementasikan mekanisme keamanan data yang efektif di lingkungan yang terdistribusi memerlukan solusi yang canggih dan strategi yang matang (Bernstein et al., 1981). Oleh karena itu, pemahaman mendalam tentang prinsip-prinsip desain basis data terdistribusi, pemilihan arsitektur yang tepat, serta implementasi mekanisme kontrol dan keamanan yang kuat menjadi kunci keberhasilan dalam membangun sistem basis data terdistribusi yang handal dan efisien.

C. Data dan Informasi Kerawanan Banjir

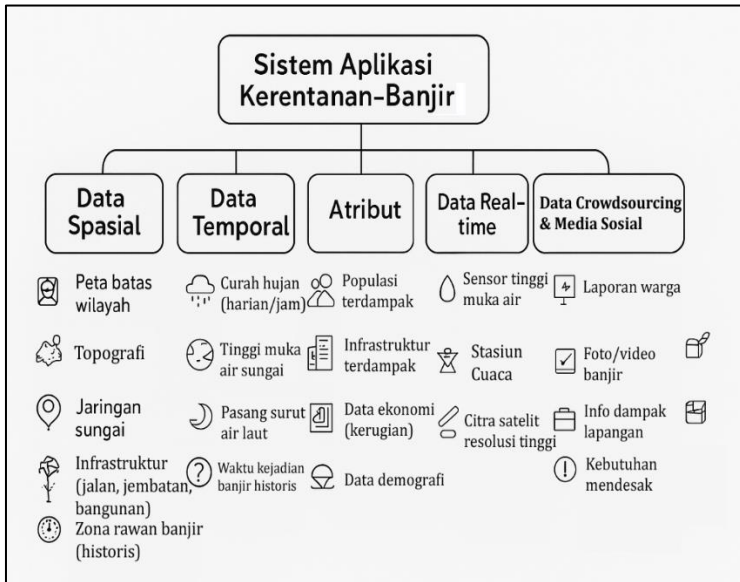
Sistem aplikasi kerawanan banjir, sebagai sebuah *platform* krusial dalam upaya mitigasi dan penanggulangan bencana hidrologi ini, memerlukan fondasi data yang komprehensif dan beragam untuk dapat berfungsi secara efektif. Keakuratan prediksi, efisiensi respons, dan efektivitas perencanaan jangka panjang sangat bergantung pada ketersediaan dan kualitas berbagai jenis data yang diintegrasikan ke dalam sistem (Goodchild, 2007). Secara garis besar, kebutuhan data untuk sistem aplikasi kerawanan banjir dapat dikategorikan menjadi beberapa jenis utama:

- **Data Spasial:** Informasi geografis memegang peranan fundamental dalam pemahaman konteks kerawanan banjir. Data spasial mencakup representasi visual dan atribut lokasi geografis, seperti peta batas wilayah administrasi (provinsi, kabupaten/kota, kecamatan, desa) yang menjadi unit analisis dalam perencanaan dan respons bencana (Longley et al., 2015). Batas Daerah Aliran Sungai (DAS) juga krusial untuk memahami unit hidrologis dan interkoneksi antar wilayah dalam konteks aliran air. Data topografi, yang menyajikan informasi ketinggian permukaan bumi, sangat penting untuk memodelkan arah aliran air dan potensi genangan. Jaringan sungai, termasuk sungai utama dan anak sungai, memberikan gambaran jalur air dan potensi luapan. Informasi infrastruktur, seperti jaringan jalan, jembatan, bangunan permukiman, dan fasilitas umum lainnya, diperlukan untuk analisis dampak dan perencanaan evakuasi. Terakhir, zona rawan banjir yang telah dipetakan berdasarkan data historis dan pemodelan menjadi informasi inti untuk identifikasi wilayah berisiko. Data spasial ini umumnya disimpan, diolah, dan divisualisasikan menggunakan teknologi Sistem Informasi Geografis (SIG) (Burrough & McDonnell, 1998).
- **Data Temporal:** Aspek waktu sangat kritical dalam dinamika banjir. Data temporal mencakup informasi yang berubah seiring berjalannya waktu, memungkinkan pemantauan kondisi dan prediksi kejadian. Data curah hujan, baik harian maupun per jam, menjadi indikator utama potensi banjir akibat curah hujan ekstrem (Huffman et al., 2019). Data ketinggian air sungai dan waduk yang diukur secara real-time memberikan gambaran langsung kondisi hidrologi dan potensi luapan. Data pasang surut air laut juga relevan, terutama untuk wilayah pesisir yang rentan terhadap banjir rob. Catatan waktu kejadian banjir historis menjadi data penting untuk analisis frekuensi, durasi, dan intensitas kejadian di masa lalu, yang dapat digunakan untuk validasi model dan perencanaan mitigasi.

- **Data Atribut:** Data atribut melengkapi informasi spasial dan temporal dengan detail deskriptif. Informasi ini mencakup karakteristik wilayah yang terdampak banjir, seperti data populasi (jumlah penduduk, kepadatan), informasi tentang infrastruktur yang mungkin terpengaruh (jenis, kondisi), data demografi (usia, kelompok rentan), data kerugian ekonomi akibat banjir (kerusakan properti, sektor usaha), dan informasi mengenai kapasitas sumber daya penanggulangan bencana (jumlah personel, peralatan, lokasi posko). Data atribut ini penting untuk analisis dampak sosial-ekonomi, perencanaan bantuan, dan alokasi sumber daya.
- **Data Real-time:** Untuk respons yang cepat dan akurat, sistem aplikasi kerawanan banjir sangat membutuhkan data yang dikumpulkan secara langsung dari sensor-sensor. Sensor ketinggian air yang ditempatkan di sungai dan waduk memberikan pembacaan kondisi terkini. Stasiun cuaca otomatis menyediakan data meteorologi real-time seperti curah hujan, kecepatan angin, dan kelembaban. Citra satelit dengan resolusi temporal yang tinggi dapat memberikan gambaran visual kondisi permukaan bumi, termasuk Extent of genangan air secara cepat setelah kejadian banjir (Schumann et al., 2016). Integrasi data real-time memungkinkan pemantauan situasi terkini dan peringatan dini yang efektif.
- **Data Crowdsourcing dan Media Sosial:** Sumber informasi yang semakin penting adalah data yang dikumpulkan dari masyarakat secara langsung melalui platform online dan media sosial. Laporan ketinggian air yang dibagikan oleh warga, foto dan video kondisi banjir di lapangan, serta informasi tentang dampak dan kebutuhan mendesak dapat menjadi sumber informasi yang berharga, terutama untuk area yang tidak terjangkau oleh sensor formal (Goodchild & Glennon, 2010). Meskipun perlu diverifikasi, data crowdsourcing dapat memberikan gambaran situasi lapangan yang lebih komprehensif dan real-time.

Integrasi dan analisis berbagai jenis data ini secara efektif dalam sistem aplikasi kerawanan banjir memungkinkan pemahaman yang

lebih mendalam tentang risiko, prediksi yang lebih akurat, respons yang lebih terkoordinasi, dan perencanaan mitigasi yang lebih efektif untuk mengurangi dampak bencana banjir. Data-data ini bersumber dari berbagai organisasi dan sistem yang berbeda, seperti BMKG, BNPB/BPBD, pemerintah daerah, jaringan sensor, dan partisipasi masyarakat. Integrasi dan standardisasi data dari berbagai sumber dengan format dan struktur yang berbeda menjadi tantangan krusial dalam membangun sistem aplikasi kerawanan banjir yang efektif dan dapat di lihat pada gambar 5.1.



Gambar 5. 1. Taxonomi Data dan Informasi Kerawanan Banjir

Gambar 5.1 menyajikan sebuah diagram hirarkis yang menggambarkan struktur data yang dibutuhkan oleh "Sistem Aplikasi Kerentanan Banjir". Diagram ini mengelompokkan kebutuhan data ke dalam lima kategori utama, masing-masing

dengan rincian elemen data spesifik dan ikon visual yang relevan. Secara keseluruhan, diagram ini secara efektif memvisualisasikan arsitektur kebutuhan data untuk sistem aplikasi kerentanan banjir, menunjukkan bahwa sistem tersebut memerlukan integrasi berbagai jenis informasi dari data geografis statis hingga laporan dinamis dari masyarakat untuk dapat berfungsi secara komprehensif dalam penilaian, pemantauan, dan respons terhadap bencana banjir.

D. Perancangan Basis Data Terdistribusi untuk Sistem Aplikasi Kerawanan Banjir

Perancangan basis data terdistribusi untuk aplikasi kerawanan banjir Perancangan basis data terdistribusi untuk sistem aplikasi kerawanan banjir merupakan tahapan krusial yang menentukan efisiensi, keandalan, dan skalabilitas sistem secara keseluruhan. Proses ini memerlukan analisis mendalam dan pertimbangan yang cermat terhadap berbagai aspek, termasuk jenis dan volume data yang akan dikelola, pola akses data oleh berbagai pengguna dan komponen sistem, serta karakteristik geografis wilayah yang menjadi fokus pemantauan (Özsu & Valduriez, 2020). Implementasi basis data terdistribusi yang efektif harus didasarkan pada pemahaman yang kuat tentang prinsip-prinsip desain yang relevan. Beberapa prinsip penting yang perlu dipertimbangkan dalam perancangan basis data terdistribusi untuk aplikasi kerawanan banjir meliputi:

- **Identifikasi Kebutuhan Data:** Langkah awal yang esensial adalah mengidentifikasi secara komprehensif jenis dan format

data yang dibutuhkan oleh setiap komponen fungsional dalam sistem aplikasi kerawanan banjir (Connolly & Begg, 2015). Misalnya, modul pemantauan memerlukan akses real-time ke data sensor ketinggian air dan curah hujan. Modul prediksi membutuhkan data historis curah hujan, topografi, dan tata guna lahan untuk menjalankan model hidrologi.

Bab 6

Business Intelligence

A. Pendahuluan

1. Definisi Business Intelligence (BI)

Business Intelligence (BI) adalah kombinasi teknologi, aplikasi, dan praktik bisnis yang digunakan untuk mengumpulkan, mengintegrasikan, menganalisis, dan mempresentasikan informasi bisnis (Rouse, M.,2022). BI dapat membantu organisasi dalam (Kimball, R., & Ross, M.,2013):

- a. Mengubah data mentah menjadi informasi yang dapat ditindaklanjuti (actionable insights).
- b. Menyediakan wawasan bisnis (business insights) melalui visualisasi, laporan, dan analisis prediktif.
- c. Memberikan keunggulan kompetitif dengan mendukung pengambilan keputusan berbasis data.

BI bukan hanya alat teknologi, tetapi mencakup proses bisnis, kebijakan manajemen data, dan kompetensi SDM dalam memanfaatkan informasi.

2. Peran BI dalam Manajemen Data Lanjutan

Manajemen data lanjutan (Advanced Data Management) adalah pendekatan modern yang mencakup pengelolaan, analisis, dan pemanfaatan data dalam skala besar dan kompleks untuk mendukung keputusan bisnis yang lebih cerdas (Negash, S.,2004). Business Intelligence (BI) memainkan peran sentral dalam proses ini dengan mengubah data mentah menjadi wawasan yang dapat ditindaklanjuti.

Manajemen data lanjutan menekankan pada (Ponniah, P.,2010.):

- a. Proses integrasi data dari berbagai sistem (heterogen dan besar).
- b. Proses analisis mendalam (misalnya prediksi tren, identifikasi pola tersembunyi).
- c. Proses pengambilan keputusan strategis berbasis data.

BI menjadi jembatan penting dalam siklus kegiatan bisnis (Turban, E., Sharda, R., & Delen, D., 2020):

- a. Menghubungkan data mentah dengan kebutuhan bisnis.
- b. Memastikan data yang dikumpulkan terolah dan disajikan dalam bentuk yang bermakna.
- c. Mengoptimalkan operasional dan mendukung transformasi digital.
- d. Contoh: Dalam industri retail, BI memungkinkan perusahaan memantau penjualan harian, memprediksi tren musiman, dan mengoptimalkan stok.

B. Konsep, Komponen, Tren, dan Inovasi BI

Dalam era digital yang didorong oleh data, Business Intelligence (BI) telah berkembang menjadi fondasi kritis bagi organisasi untuk mengubah informasi mentah menjadi wawasan strategis (Sharda, R., Delen, D., & Turban, E., 2021). Sub-bab ini akan menguraikan konsep inti, komponen utama, tren terkini, dan inovasi dalam BI, dengan fokus pada arsitektur yang mendukung untuk proses analisis data.

1. Arsitektur BI

Arsitektur Business Intelligence (BI) merupakan kerangka terstruktur yang memungkinkan organisasi mengumpulkan, mengolah, dan menganalisis data dari berbagai sumber untuk menghasilkan insights yang bernilai. Komponen-komponen utamanya mencakup integrasi data mentah, transformasi, penyimpanan terpusat, hingga visualisasi, yang bersama-sama membentuk alur kerja BI yang efektif dan skalabel (Kimball & Ross, 2013).

BI memiliki arsitektur yang mencakup:

- a. Data Sources (Internal & Eksternal): ERP, CRM, media sosial, data pasar, IoT.
- b. ETL (Extract, Transform, Load): Proses memindahkan data dari berbagai sumber ke penyimpanan terpusat.
- c. Data Warehouse: Penyimpanan data historis yang terintegrasi untuk analisis mendalam.
- d. OLAP (Online Analytical Processing): Teknologi untuk eksplorasi data multidimensi dan analisis cepat.

Bab 7

Forensik Basis Data

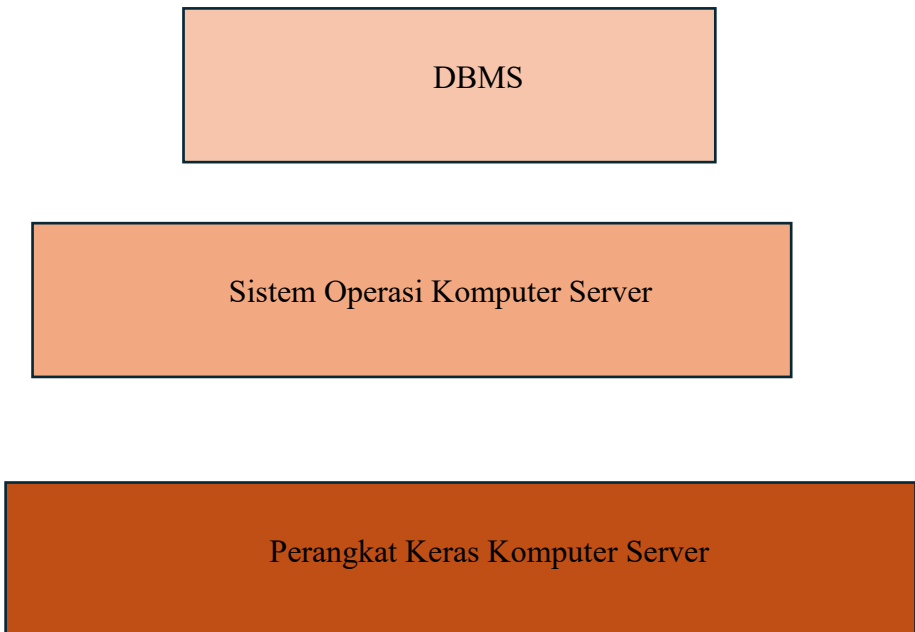
A. Pendahuluan

Forensik basis data merupakan cabang dari ilmu forensik digital yang dapat digunakan untuk mencari nilai dalam basis data sebagai bagian integral dari tata laksana hukum pidana yang ada di Indonesia. Sebagai bagian dari forensik digital, forensik basis data berfokus pada penyelidikan, analisis, dan pelacakan aktivitas mencurigakan dalam sistem manajemen basis data (DBMS). Komponen yang digunakan untuk membangun DBMS antara lain :

1. Perangkat keras komputer berupa komputer server yang memiliki spesifikasi terbaik untuk memberikan pelayanan 24 jam selama 7 hari berturut-turut tanpa dimatikan.
2. Perangkat lunak komputer yang dipasang pada komputer server berupa sistem operasi, under linux atau microsoft, serta perangkat lunak untuk pengelolaan basis data sebagai contoh mySql atau Oracle Database.

Dalam konteks hukum dan keamanan siber, forensik basis data memainkan peran penting dalam penyelidikan kriminal digital, baik itu oleh penegak hukum, auditor keamanan, maupun tim keamanan internal perusahaan.

Sebagai bagian dari forensik digital, forensik basis data tidak dapat berdiri sendiri saat digunakan untuk mencari bukti tindakan kejahatan siber yang akan digunakan di muka hukum. Hal ini terkait dengan validasi data yang diambil baik pada basis datanya serta pada sistem operasinya yang berada di bawah basis data. Sudah menjadi pengetahuan publik bahwa perangkat lunak DBMS tidak dapat bekerja jika langsung ditempatkan pada perangkat keras komputer (Kim, 2014; Kumar Gupta, 2013), DBMS membutuhkan sistem operasi server yang memiliki fasilitas file log yang perlu diaktifkan saat proses instalasi dan konfigurasi seperti yang dapat dilihat pada Gambar 7.1.



Gambar 7. 1. Struktur Basis Data pada komputer

Seperti yang terlihat pada Gambar 7.1., file log menjadi bagian dari sistem operasi server berada di atas perangkat keras dan di bawah DBMS. File log pada komputer server adalah file yang mencatat semua aktivitas atau kejadian penting yang terjadi di dalam sistem server, baik dari sisi perangkat keras, perangkat lunak, jaringan, hingga interaksi pengguna. File ini sangat penting untuk pemantauan, pemeliharaan, keamanan, dan troubleshooting (pemecahan masalah) server. File log ini sangat penting sekali dalam proses validasi data aktivitas pengguna komputer server, karena pada DBMS tidak tersedia sehingga pola validasi akan mencari perubahan data pada DBMS berdasarkan laporan kejadian serangan.

Serangan yang dilakukan oleh para pelaku tindakan kejahatan ini secara teknis hampir sama dengan serangan pada komputer baik personal atau server. Dampak serangan yang terjadi pada basis data meliputi perubahan data pada tabel, penghapusan data pada tabel, injeksi data pada tabel hingga penyalinan data ilegal pada tabel yang semuanya ada pada DBMS.

Sebagai bagian dari forensik digital, forensik basis data juga mengikuti aturan yang berlaku pada forensik digital. Hal-hal yang perlu dilakukan dalam melaksanakan tata laksana forensik basis data harus sesuai dengan standar dan aturan yang berlaku secara

internasional. Sehingga para auditor pelaksana dari forensik basis data juga merupakan auditor forensik digital yang wajib memiliki sertifikat keahlian dalam bidang forensik digital sebagai contoh dapat dilihat pada Gambar 7.2



Gambar 7. 2. Contoh sertifikat CHFI

Terlihat pada Gambar 7.2 contoh sebuah sertifikat *Computer Hacking Forensic Investigator* (CHFI)(Porostimur, 2021) yang wajib dimiliki oleh seseorang yang akan terlibat sebagai saksi ahli dalam persidangan kasus forensik digital. Sertifikat dari EC-Council merupakan bukti kompetensi dalam bidang forensik komputer dan penyelidikan peretasan. Sertifikat ini sangat dihargai di industri keamanan siber. Selain CHFI tentu masih ada sertifikasi lainnya yang dapat memberikan pengakuan kemampuan seseorang dalam bidang forensik basis data yang dapat digunakan untuk

memberikan keterangan saksi ahli dalam persidangan. Kejahatan siber yang terkait dengan basis data pada tahun 2024 belum berhasil dibawa ke pengadilan. Bjorka, seorang hacker, diduga bertanggung jawab atas kebocoran data 6 juta data NPWP termasuk data dari mantan presiden Republik Indonesia yang ke 7(Tempo, 2024) belum juga berhasil dibawa ke pengadilan meski proses penyelidikan teru dilakukan oleh aparat yang memiliki keahlian dalam bidang forensik basis data.

B. Konsep Dasar Forensik Basis Data

Sebelum dan sesudah terjadinya serangan pada objek basis data menjadi sangat penting sekali untuk dipotret keadaannya sebagai langkah dalam validasi data. Dengan demikian tentunya forensik basis data memiliki tujuan(Morrison, 2022) dalam pelaksanaannya antara lain :

1. Mendeteksi adanya aktivitas mencurigakan atau pelanggaran kebijakan yang terjadi di dalam sistem basis data.
2. Menyimpan dan memverifikasi bukti secara sah agar dapat digunakan dalam proses persidangan.
3. Menelusuri tindakan pengguna atau proses dalam sistem basis data, termasuk siapa yang mengakses apa, kapan, dan bagaimana.
4. Menggunakan hasil forensik untuk mengembalikan dan memperbaiki kerusakan setelah terjadi insiden.
5. Memberikan rekomendasi peningkatan keamanan berdasarkan temuan forensik.

Beberapa pembuktian tindakan kejahatan siber memang berhenti pada ditemukannya bukti forensik, tetapi pada kasus basis data tentu tidak berhenti begitu saja pada penemuan bukti forensik mengingat pentingnya data yang tersimpan pada basis data untuk segera dipulihkan dalam proses mitigasinya. Proses mitigasi yang menjadi jaminan kembalinya data seperti sebelum terjadinya serangan juga bukan perkara yang mudah karena struktur file pada basis data tidak sama dengan file dan folder pada umumnya. Jika file dan folder pada sebuah server dapat dimonitor dengan menggunakan tools tripwire(Wintolo, 2025), maka di basis data tidak bisa demikian. Sebuah basis data di dalamnya pasti berisi tabel dan setiap tabel selain berelasi dengan tabel lainnya juga memiliki record data di dalamnya.

Ukuran basis data dan tabel menjadi faktor penting dalam penanganan serangan pada sebuah database server. Ukuran basis data dan tabel yang di luar nalar kenaikannya perlu menjadi perhatian para investigator dalam melakukan forensik basis data(Ge & Budowle, 2021). Secara umum hal yang perlu dilakukan dalam rangka memonitor ukuran basis data yaitu :

1. Ukur kondisi terkini bisa menggunakan durasi waktu yang kontinu dan disimpan
2. Bandingkan ukuran terbaru dengan ukuran basis data yang terakhir disimpan

Langkah-langkah di atas tentu tidak secara mutlak dapat diambil kesimpulan telah terjadi serangan jika ditemukan perbedaan yang di luar nalar. Sehingga para investigator perlu meneruskan pada langkah selanjutnya yaitu membandingkan ukuran tabel yang ada pada basis data dengan langkah yang hampir sama. Cara melakukan pengukuran basis data dan tabel dapat dilihat pada Gambar 7.3 dengan menggunakan basis data Oracle.

```
SELECT
SUM(bytes) /1024/10
24 AS ukuran_mb
FROM
dba_data_files;
```

```
SELECT
segment_name AS "Tabel",
ROUND (SUM(bytes) /1024/10
24, 2) AS "Ukuran_MB"
FROM
dba_segments
WHERE
```

Gambar 7. 3. Perintah untuk merekam ukuran basis data dan tabel pada oracle basis data

Proses penyimpanan data, baik dalam bentuk basis data maupun tabel dalam basis data, yang dilakukan secara kontinu dapat menimbulkan berbagai permasalahan baru (Varga, 2009). Salah satu

BAB 8

Keamanan dan Optimalisasi Basis Data

A. Pendahuluan

Dalam era digital yang semakin maju, data telah menjadi aset yang sangat penting bagi organisasi di seluruh dunia. Basis data modern menyediakan fondasi yang diperlukan untuk menyimpan, mengelola, dan mengakses data dengan cepat dan efisien. Basis data adalah kumpulan data yang disusun dengan cara yang memungkinkan pengelolaan yang efisien dan akses yang cepat. Sistem manajemen basis data (DBMS) bertanggung jawab untuk menyimpan, mengelola, dan menyediakan akses data kepada pengguna atau aplikasi secara terstruktur. Basis data memainkan peran penting dalam pengelolaan informasi di berbagai sektor, mulai dari bisnis, pemerintahan, hingga teknologi informasi (Ramakrishnan & Gehrke, 2003).

Namun, dengan meningkatnya volume data dan kompleksitas aplikasi, kebutuhan untuk mengamankan dan mengoptimalkan basis data menjadi semakin kritis. Keamanan data tidak hanya tentang melindungi informasi dari ancaman eksternal, tetapi juga menjaga integritas dan kerahasiaan data yang dapat mempengaruhi keputusan bisnis dan kepercayaan pelanggan. Di sisi lain, optimisasi basis data memastikan bahwa sistem dapat menangani volume data yang besar

dan permintaan query yang kompleks tanpa mengorbankan kinerja (Bertino et al., 2005; Gulutzan & Pelzer, 2006).

Keamanan dan optimalisasi basis data menjadi semakin penting di era digital saat ini, karena data telah menjadi salah satu aset terpenting bagi organisasi. Organisasi bergantung pada data untuk pengambilan keputusan yang berbasis pada analisis dan operasional bisnis sehari-hari. Namun, semakin besar volume data yang dikelola, semakin besar pula potensi ancaman terhadap data tersebut, seperti serangan peretas, kebocoran informasi, dan serangan DDoS terhadap server basis data. (IBM, 2023).

Bab ini bertujuan untuk memberikan pemahaman mendalam tentang aspek keamanan dan optimalisasi dalam pengelolaan basis data modern. Pembahasan akan mencakup berbagai teknik yang digunakan untuk mengamankan data, seperti autentikasi, otorisasi, enkripsi, serta praktik terbaik dalam mengoptimalkan kinerja basis data, termasuk penggunaan indeks, partisi data, dan query optimization. Semua ini bertujuan untuk memastikan integritas, ketersediaan, dan performa data yang optimal (Ramakrishnan & Gehrke, 2003; Gulutzan & Pelzer, 2006).

Seiring dengan upaya untuk meningkatkan kinerja basis data, aspek keamanan dan optimalisasi menjadi dua faktor yang saling mendukung. Keamanan berfokus pada perlindungan data dari ancaman yang dapat merusak integritas dan kerahasiaannya, dengan autentikasi, otorisasi, enkripsi, dan pemantauan aktivitas sebagai komponen utama. Ketidakamanan dapat mengakibatkan eksposur atau manipulasi data yang merugikan organisasi secara finansial dan

reputasional (Bertino et al., 2005). Sementara itu, optimalisasi berperan dalam meningkatkan efisiensi dan kecepatan sistem, terutama dengan volume data yang besar, melalui teknik seperti partisi, penggunaan indeks, dan optimasi query untuk memperbaiki kinerja sistem (Gulutzan & Pelzer, 2006).

Meskipun keamanan dan optimisasi sangat penting, banyak tantangan yang harus dihadapi dalam pengelolaannya. Dalam aspek keamanan, ancaman dapat datang dari berbagai sumber, termasuk peretas, kesalahan manusia, atau serangan DDoS, yang dapat mengekspos atau merusak data yang sensitif. Selain itu, dalam hal optimisasi, tantangan utama termasuk masalah skalabilitas, keterbatasan hardware, dan teknik pengoptimalan yang kadang rumit, yang memerlukan pemahaman mendalam dan perawatan rutin agar dapat tetap efektif (IBM, 2023; Google Cloud, 2023).

Dengan pemahaman dasar mengenai pentingnya keamanan dan optimisasi dalam pengelolaan basis data, bab ini akan membahas lebih dalam mengenai topik-topik kunci yang mendasari kedua aspek tersebut.

B. Fondasi Basis Data Modern dan Model Data

Kebutuhan untuk mengelola volume data besar dan bervariasi telah mengubah desain sistem basis data. Basis data modern dirancang untuk menangani data besar, beragam jenis data, dan skala yang lebih besar, baik dalam transaksi maupun pengguna, dibandingkan dengan sistem relasional tradisional. Sistem ini harus

mampu mengelola data terstruktur dan tidak terstruktur, serta tersebar di berbagai lokasi fisik, didorong oleh revolusi teknologi seperti big data, cloud computing, dan pemrosesan real-time (Ramakrishnan & Gehrke, 2003).

Basis data modern penting di dunia digital karena memungkinkan akses data cepat, efisien, dan andal. Teknologi seperti cloud computing mendukung penyimpanan dan akses data terdistribusi, meningkatkan skalabilitas dan ketersediaan. Selain itu, basis data modern vital dalam big data analytics untuk memproses dan menganalisis data real-time. Aplikasi seperti e-commerce, analisis media sosial, dan pengelolaan data kesehatan mengandalkan sistem basis data ini (Google Cloud Database Performance, 2023).

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak untuk mengelola, mengorganisir, dan mengakses data. DBMS modern memiliki fitur untuk mendukung manajemen transaksi, keamanan data, pencarian efisien, serta mekanisme redundansi untuk menjaga keamanan data saat kegagalan sistem. DBMS juga mampu mengelola berbagai jenis data, termasuk terstruktur, semi-terstruktur, dan tidak terstruktur (Ramakrishnan & Gehrke, 2003).

Teknologi seperti cloud computing dan distributed databases mengubah cara basis data dikelola, memungkinkan penyimpanan data di server terdistribusi secara geografis untuk meningkatkan skalabilitas dan toleransi kesalahan (*fault tolerance*). NoSQL databases, seperti MongoDB dan Cassandra, muncul untuk memenuhi kebutuhan aplikasi yang tidak dapat ditangani dengan sistem basis data relasional tradisional (Gulutzan & Pelzer, 2006).

Model relasional (*Relational Model*), yang mengorganisir data dalam tabel, baris, dan kolom, masih banyak digunakan di aplikasi modern. Data dalam model ini biasanya terstruktur dengan jelas, dengan normalisasi untuk mengurangi redundansi. Meskipun ada kemajuan dengan model lain, model relasional tetap menjadi pilihan utama untuk aplikasi yang memerlukan integritas dan kemudahan akses data (Ramakrishnan & Gehrke, 2003).

Model NoSQL digunakan untuk aplikasi yang memerlukan scalability tinggi, serta fleksibilitas dalam mengelola data yang tidak terstruktur atau semi-terstruktur. Beberapa jenis model NoSQL antara lain:

- Key-Value Stores: Menyimpan data dalam pasangan key-value (misalnya, Redis).
- Document Stores: Menyimpan data dalam format dokumen (misalnya, MongoDB).
- Column-Family Stores: Menyimpan data dalam kolom yang dapat dipartisi (misalnya, Cassandra).
- Graph Databases: Menyimpan data dalam bentuk graf untuk menangani hubungan antar data (misalnya, Neo4j).

Model Berbasis Objek (*Object-Oriented Data Model*) mengatasi keterbatasan model relasional dengan menyimpan data sebagai objek, di mana data dan operasi terkait disimpan dalam satu unit (Garcia-Molina et al, 2009).

Model Data Grafik (*Graph Model*), model graf digunakan untuk data yang sangat terhubung, seperti analisis jaringan sosial, di mana entitas diwakili sebagai simpul (node) dan hubungan antar entitas sebagai sisi (edges) (Bertino et al., 2005).

Tabel berikut ini menyajikan perbandingan antara beberapa model data yang umum digunakan dalam sistem basis data.

Tabel 8. 1. Perbandingan Antara Beberapa Model Data

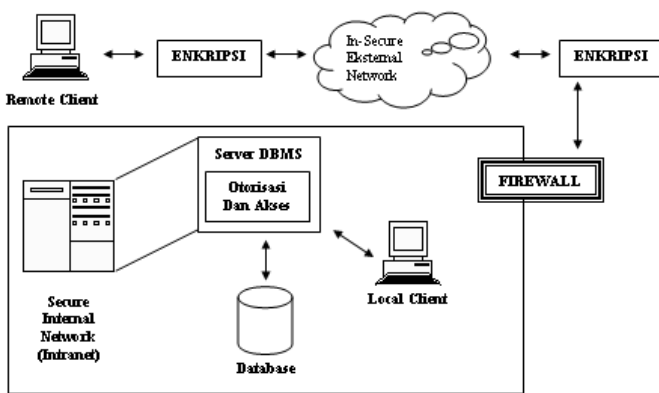
Model Data	Kelebihan	Kekurangan
Relasional	Kuat dalam transaksi dan integritas data	Skalabilitas terbatas untuk data besar
NoSQL	Scalability tinggi, fleksibel, cocok untuk data tidak terstruktur	Kurang mendukung konsistensi data yang ketat
Berbasis Objek	Penyimpanan data yang lebih natural untuk objek kompleks	Tidak cocok untuk aplikasi dengan query yang rumit
Grafik	Ideal untuk data yang sangat terhubung	Tidak efisien untuk data yang lebih terstruktur

Pemilihan model data bergantung pada kebutuhan aplikasi dan karakteristik data. Model relasional cocok untuk aplikasi yang memerlukan integritas data, sementara NoSQL menawarkan fleksibilitas dan skalabilitas untuk data besar dan tidak terstruktur. Model berbasis objek dan graf lebih tepat untuk aplikasi dengan data kompleks dan hubungan yang saling terhubung. Pemahaman mendalam tentang kelebihan dan kekurangan masing-masing model sangat penting untuk merancang sistem basis data yang optimal.

C. Keamanan Basis Data: Autentikasi, Otorisasi, Enkripsi dan Integritas Data

Pentingnya Keamanan dalam Basis Data karena data yang disimpan dalam basis data sering kali bersifat sensitif dan sangat

berharga. Data yang tidak terlindungi dapat menyebabkan kerugian besar baik secara finansial maupun reputasi bagi organisasi. Keamanan data bertujuan untuk menjaga tiga aspek utama: kerahasiaan, integritas, dan ketersediaan data. Keamanan ini tidak hanya melindungi data dari ancaman eksternal (seperti peretas), tetapi juga mencegah akses atau manipulasi yang tidak sah oleh pihak internal (Bertino et al., 2005).



Gambar 8. 1. Gambaran Keamanan Basis Data

Keamanan basis data terdiri dari empat komponen utama yang bekerja bersama untuk melindungi data dari ancaman: Autentikasi, Otorisasi, Enkripsi, Integritas Data.

Autentikasi adalah langkah pertama dalam mengamankan akses ke basis data, memastikan hanya pengguna sah yang dapat mengakses data. Metode autentikasi seperti password, biometrik, dan multifactor authentication (MFA) digunakan untuk meningkatkan keamanan. Setelah autentikasi, otorisasi mengatur hak akses pengguna. Model otorisasi yang umum adalah Role-Based Access Control (RBAC), yang mengatur akses berdasarkan peran,

dan Attribute-Based Access Control (ABAC), yang mengatur akses berdasarkan atribut pengguna atau data, memastikan pengguna hanya mengakses data sesuai izin mereka. (Ramakrishnan & Gehrke, 2003).

Enkripsi data dalam transit dan at-rest melindungi data dari akses yang tidak sah. Enkripsi in-transit menggunakan protokol seperti TLS/SSL untuk melindungi data yang berpindah melalui jaringan, sementara enkripsi at-rest mengamankan data yang disimpan menggunakan algoritma seperti AES. Kedua metode ini memastikan data tidak dapat dibaca oleh pihak yang tidak berwenang meskipun berhasil diakses (IBM, 2023).

Integritas data memastikan bahwa data dalam basis data tetap konsisten, akurat, dan tidak dimodifikasi tanpa izin. Beberapa metode digunakan untuk menjaga integritas data, seperti checksum, yang menghitung nilai hash dari data untuk memastikan data tidak berubah selama transmisi atau penyimpanan. Hashing juga digunakan untuk membandingkan integritas data, memastikan bahwa data tidak telah dimanipulasi. Selain itu, tanda tangan digital memberikan jaminan bahwa data berasal dari sumber yang sah dan tidak mengalami perubahan.

Audit trail dan logging memainkan peran penting dalam mendeteksi perubahan data yang tidak sah. Dengan memantau siapa yang mengakses data dan perubahan apa yang dilakukan, organisasi dapat mendeteksi potensi pelanggaran keamanan dan memastikan integritas data tetap terjaga (ISO/IEC 27001, 2022).

Praktik terbaik untuk keamanan basis data melibatkan beberapa langkah penting. Keamanan jaringan dan firewall memainkan peran utama dalam membatasi akses tidak sah ke basis data. Dengan menggunakan firewall yang tepat, baik perangkat keras maupun perangkat lunak, akses dari luar organisasi dapat dicegah.

Selain itu, pengelolaan hak akses yang ketat memastikan bahwa hanya pengguna dengan izin yang tepat yang dapat mengakses atau memodifikasi data. Prinsip least privilege, yang memberikan akses minimal yang diperlukan, sangat penting untuk mengelola hak akses dengan hati-hati.

Pemantauan basis data secara terus-menerus juga sangat penting untuk mendeteksi ancaman dan aktivitas mencurigakan. Sistem seperti Intrusion Detection System (IDS) atau Intrusion Prevention System (IPS) dapat memperkuat pertahanan dan merespons potensi serangan dengan cepat (Microsoft, 2023).

Keamanan basis data adalah upaya berkelanjutan untuk melindungi data dari ancaman dengan teknik seperti autentikasi, otorisasi, enkripsi, dan integritas data. Pengelolaan hak akses, pemantauan terus-menerus, serta sistem deteksi intrusi juga penting untuk menjaga keamanan. Organisasi harus terus memperbarui kebijakan dan praktik keamanannya untuk menghadapi ancaman yang berkembang.

D. Partisi Basis Data untuk Skalabilitas dan Kinerja

Partisi basis data adalah proses membagi data dalam tabel besar menjadi bagian-bagian yang lebih kecil atau "partisi" untuk

mempermudah pengelolaan. Setiap partisi dapat disimpan di server atau node berbeda, meningkatkan efisiensi pengelolaan data. Misalnya, data dalam tabel transaksi dapat dipartisi berdasarkan rentang tanggal atau ID pelanggan (Ramakrishnan & Gehrke, 2003).

Tujuan utama partisi adalah meningkatkan skalabilitas dan kinerja, dengan memungkinkan sistem mengelola volume data besar secara efisien. Dalam sistem basis data modern, terutama yang menangani big data dan aplikasi real-time, partisi penting untuk mendistribusikan data dan beban kerja secara merata, memungkinkan sistem menangani lebih banyak transaksi dengan cepat (Gulutzan & Pelzer, 2006).

Jenis-jenis partisi basis data dapat dibedakan berdasarkan cara data dibagi dan disebar. Partisi horizontal, atau sharding, membagi data berdasarkan rentang nilai atau hashing kolom tertentu, seperti ID pelanggan atau rentang tanggal, yang kemudian disebar ke beberapa node atau server. Sharding ini sangat berguna dalam sistem yang memerlukan distribusi data secara merata, mengurangi beban pada server tunggal dan memungkinkan penanganan volume data besar (MongoDB Documentation, 2023).

Sementara itu, partisi vertikal membagi data berdasarkan kolom dalam tabel, memungkinkan pemisahan data yang sering diakses dari yang jarang diakses. Sebagai contoh, dalam sistem transaksi, data profil pelanggan yang sering diakses dapat dipartisi terpisah dari data transaksi yang lebih jarang diakses. Selain itu, ada juga partisi berdasarkan aplikasi atau fungsi tertentu, di mana data dibagi sesuai kebutuhan aplikasi, seperti memisahkan data transaksi dari

data analitik atau data historis. Pendekatan ini membantu meningkatkan kinerja dan mempermudah pengelolaan data (Ramakrishnan & Gehrke, 2003).

Keuntungan partisi basis data terlihat dalam peningkatan skalabilitas, efisiensi, dan performa. Dengan membagi data ke dalam partisi, sistem dapat menangani lebih banyak data dan permintaan secara paralel. Partisi horizontal (sharding) mendistribusikan data secara merata, mengurangi beban pada server tunggal, dan meningkatkan efisiensi. Selain itu, partisi juga mempercepat query karena hanya data yang relevan yang perlu diproses, seperti pada query yang mencari data berdasarkan tanggal (Gulutzan & Pelzer, 2006).

Tabel berikut ini memberikan perbandingan antara berbagai teknik partisi basis data, menyoroti keuntungan, tantangan, dan situasi yang tepat untuk menerapkan masing-masing teknik tersebut.

Tabel 8. 2. Perbandingan Antara Berbagai Teknik Partisi Basis Data

Jenis Partisi	Deskripsi	Keuntungan	Tantangan	Kapan Digunakan
Partisi Horizontal	Data dibagi berdasarkan rentang nilai atau hash dan disebar	Meningkatkan skalabilitas horizontal dan kinerja query tertentu	Pengelolaan partisi yang rumit dan potensi data skew	Sistem dengan volume data besar dan distribusi yang

	an ke banyak server			merata
Partisi Vertikal	Data dibagi berdasarkan kolom, seperti memisahkan data yang sering diakses dari data yang jarang diakses	Meningkatkan kinerja dengan mempercepat akses data yang sering dipakai	Pengelolaan yang lebih kompleks dan mungkin tidak cocok untuk semua aplikasi	Data yang memiliki kolom dengan frekuensi akses berbeda
Composite Partitoning	Kombinasi partisi horizontal dan vertikal	Memaksimalkan kinerja dan distribusi data	Memerlukan konfigurasi dan pemeliharaan yang lebih rumit	Ketika kombinasi partisi dibutuhkan untuk beban kerja kompleks

Partisi mempermudah pemeliharaan basis data, karena pengarsipan atau penghapusan data lama dapat dilakukan pada

Daftar Pustaka

- Ade Kurniawan Rusdy. (2015). Laporan Dokumen Investigasi Digital Forensics Chain of Custody.
- Al-Dhaqm, A., Abd Razak, S., Othman, S. H., Ali, A., Ghaleb, F. A., Rosman, A. S., & Marni, N. (2020). Database forensic investigation process models: A review. *IEEE Access*, 8, 48477–48490. <https://doi.org/10.1109/ACCESS.2020.2976885>
- Almeida, F., & Calistru, C. (2023). The main challenges of business intelligence implementation. *International Journal of Business Intelligence Research*, 14(1), 1–15. <https://doi.org/10.4018/IJBIR.20230101>
- Alraddadi, A. S. (2024). Reconstruction Investigation Model for Database Management Systems. *Journal of Computer Science*, 20(1), 33–43. <https://doi.org/10.3844/jcssp.2024.33.43>
- Ayeni, P. O., & Adesoba, O. C. (2025). IoT-based home control system using NodeMCU and Firebase. *Journal of Edge Computing*, 4(1), 17–34. <https://doi.org/10.55056/jec.814>
- Baars, H., & Kemper, H.-G. (2008). "Management Support with Structured and Unstructured Data—An Integrated Business Intelligence Framework." *Information Systems Management*, 25(2), 132–148.
- Batini, C., Scannapieco, M., & Jarke, M. (2016). Data Quality: Concepts, Methodologies and Techniques. *Data & Knowledge Engineering*, 5(1), 101-130. <https://doi.org/10.1016/j.datak.2016.02.002>
- Bernstein, P. A., Goodman, N., & Hadzilacos, V. (1981). Concurrency Control in Distributed Database Systems. *Proceedings of the Seventh International Conference on Very Large Data Bases*, 13(2), 548–559. <https://doi.org/10.1145/356842.356846>
- bin Uzayr, S. (2023). *Mastering SQL: A Beginner's Guide*. CRC Press.

- Burrough, P. A., & McDonnell, R. A. (1998). *Principles of Geographical Information Systems*. Oxford University Press.
- Caturjayanti, V. D. (2020). *Konsep Privacy By Design Sebagai Perlindungan Data The Concept Of Privacy By Design As The Protection Of Personal Data Of Users Of The Application "Peduli Lindungi"*. <https://covid19.go.id/>,
- Cavoukian, A. (2011). *Privacy by design: The 7 foundational principles*. Information and Privacy Commissioner of Ontario.
- Ceri, S., Bernasconi, A., Gagliardi, A., Martinenghi, D., Bellomarini, L., & Magnanimiti, D. (2024). PG-Triggers: Triggers for Property Graphs. *Companion of the 2024 International Conference on Management of Data*, 373–385. <https://doi.org/10.1145/3626246.3653386>
- Chen, H., Chiang, R. H., & Storey, V. C. (2012). Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, 36(4), 1165-1188. <https://doi.org/10.2307/41703503>
- Chodorow, K. (2013). *MongoDB: The Definitive Guide (2nd ed.)*. O'Reilly Media.
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6), 377-387.
- Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.)*. Pearson Education.
- Corone, C., & Morris, S. (2019). *Database Systems: Design, Implementation, and Management*. Cengage Learning, Inc.
- Coronel, Carlos., Morris, Steven., & Rob, Peter. (2011). *Database systems: Design, Implementation, and Management*. Course Technology Cengage Learning.
- Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2011). *Distributed Systems: Concepts and Design (5th ed.)*. Addison-Wesley.

- Date, C. J. (2000). *An Introduction to Database Systems* (8th ed.). Addison-Wesley.
- Davenport, T. H. (2014). *Big Data at Work: Dispelling the Myths, Uncovering the Opportunities*. Harvard Business Review Press.
- Davidson, L., & Davidson, L. (2021). Data Protection Patterns with Check Constraints and Triggers. *Pro SQL Server Relational Database Design and Implementation: Best Practices for Scalability and Performance*, 427–505.
- de Brock, B. (2023). Managing Database Trigger Design. In B. Shishkov (Ed.), *Business Modeling and Software Design* (pp. 352–364). Springer Nature Switzerland.
- Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson Education.
- Enders, C. K. (2022). *Applied missing data analysis* (2nd ed.). Guilford Press.
- Falah, A. S., SE, M. M., Setiana, A. R., & SE, M. M. (2024). *Sistem Informasi Manajemen Keuangan: Implementasi dan Pengelolaan*. Takaza Innovatix Labs.
- Field, A. (2018). *Discovering statistics using IBM SPSS statistics* (5th ed.). Sage.
- Fitriani, W. R., & Suryanegara, M. (2020). IoT-Based Smart Mosque: A Systematic Literature Review. *International Journal of Advanced Computer Science and Applications*, 11(8), 287–294. <https://doi.org/10.14569/IJACSA.2020.0110837>
- Fitriani, W. R., & Suryanegara, M. (2020). IoT-Based Smart Mosque: A Systematic Literature Review. *International Journal of Advanced Computer Science and Applications*, 11(8), 287–294. <https://doi.org/10.14569/IJACSA.2020.0110837>

- Gandomi, A., & Haider, M. (2015). Beyond the Hype: Big Data Concepts, Methods, and Analytics. *International Journal of Information Management*, 35(2), 137–144. <https://doi.org/10.1016/j.ijinfomgt.2014.10.007>
- García, S., Luengo, J., & Herrera, F. (2020). *Data Preprocessing in Data Mining*. Springer. <https://doi.org/10.1007/978-3-030-32392-0>
- García-Laencina, P. J., Sancho-Gómez, J. L., & Figueiras-Vidal, A. R. (2020). Pattern classification with missing data: A review. *Neural Computing and Applications*, 32(8), 2701-2718.
- Gartner (2023). *Business Intelligence Trends*. Retrieved from [gartner.com](https://www.gartner.com)
- Gartner. (2023). *Market guide for data and analytics governance platforms (ID G00741131)*. <https://www.gartner.com/document/4567891>
- Ge, J., & Budowle, B. (2021). Forensic investigation approaches of searching relatives in DNA databases. *Journal of Forensic Sciences*, 66(2), 430–443. <https://doi.org/10.1111/1556-4029.14615>
- Ghorbani, A., Modarres, R., & Khosravi, A. (2021). Data Quality and Missing Data Challenges in Big Data Analytics. *Journal of Data and Information Quality*, 13(1), Article 2. <https://doi.org/10.1145/3456467>
- Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4), 211–221.
- Goodchild, M. F., & Glennon, J. A. (2010). *Crowdsourcing geographic information for disaster response and preparedness: A research agenda*. National Center for Geographic Information and Analysis, University of California, Santa Barbara.
- Gordon, K. (2013). *Principles Of Data Management: Facilitating Information Sharing (Second edition)*. BCS, The Chartered Institute for IT,.

- Graham, J. W. (2021). *Missing data: Analysis and design*. Springer.
- Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2019). *Multivariate data analysis* (8th ed.). Cengage Learning.
- Hammerly, A., & Gragnarella, S. (2025, May 20). *Firestore Realtime Database*. <https://firebase.google.com/docs/database>.
- Handoyo, J., Yudhana, A., & Sunardi. (2025). Analisis Spasial Tingkat Kerawanan Banjir di Kecamatan Cepu Kabupaten Blora dengan Pendekatan Metode Skoring Berbasis Sistem Informasi Geografis. *Jurnal Sainteks*, 22(1). <https://doi.org/10.30595/sainteks.v22i1.25180>
- Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Khan, S. U. (2015). The rise of “big data” on cloud: Review and open research issues. *Information Systems*, 47, 98–115.
- Hermawan, A. F. K. F. F. (2008). PENGEMBANGAN SISTEM BASIS DATA PRESENSI PERKULIAHAN DENGAN KARTU MAHASISWA BER-BARCODE. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, Vol 6, No 1: April 2008, 65–72. <http://journal.uad.ac.id/index.php/TELKOMNIKA/article/view/552/361>
- Hidayat, A. W. (2020). *Perlindungan Data Pribadi Dalam Penggunaan Aplikasi Pedulilindungi Selama Pandemi Covid-19*. <https://covid19.go.id/>,
- Hoffer, J. A. ., Ramesh, V. ., & Topi, Heikki. (2019). *Modern database management*. Pearson Education.
- Hoffer, J. A. ., Ramesh, V. ., & Topi, Heikki. (2019). *Modern database management*. Pearson Education.
- Howson, C., Sallam, R. L., & Richardson, J. (2022). *Magic quadrant for analytics and business intelligence platforms*. Gartner Research. <https://www.gartner.com/document/1234567>

- Huffman, G. J., Bolvin, D. T., Nelkin, E. J., Wolff, D. B., Adler, R. F., Gu, G., & Stocker, E. F. (2019). The TRMM Multisatellite Precipitation Analysis (TMPA), Version 3G68. NASA Goddard Space Flight Center, Greenbelt, MD, USA.
- Ibrahim, S. S. M., Faisol, A., & Promaswara, R. (2023). Rancang Bangun Aplikasi Controlling Dan Monitoring Ruangan Jemuran Pintar Menggunakan Firebase Pada Arduino Dan Android. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1).
- Ikhsan, K., & Setyawan, B. (2024). Rancang Bangun Smart Home Berbasis Internet Of Things (IOT) Menggunakan Arduino Mega 2560. *Jurnal Ilmiah Scroll: Jendela Teknologi Informasi*, 12(1). <https://univ45sby.ac.id/ejournal/index.php/informatika>
- Inmon, W. H. (2005). *Building the Data Warehouse* (4th ed.). Wiley.
- Inmon, W. H. (2016). *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*. Technics Publications.
- Inmon, W. H. (2016). *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*. Technics Publications.
- Jakobsen, J. C., Gluud, C., Wetterslev, J., & Winkel, P. (2019). When and how should multiple imputation be used for handling missing data in randomised clinical trials - a practical guide with flowcharts. *BMC Medical Research Methodology*, 17(1), 162.
- Jom FTI Budiluhur. (2022). Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Klasifikasi. *Jom FTI Budiluhur*, 10(1), 1-12.
- Jumintoko, A. W., & Sutiman, S. (2022). Strategi penanganan missing data dalam penelitian akademik. *Jurnal Penelitian Akademik*, 10(2), 123-135.
- Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed.). Pearson.

- Khan, S., & Vorley, T. (2023). Big data governance and organizational change. *Journal of Business Research*, 158, 113672. <https://doi.org/10.1016/j.jbusres.2023.113672>
- Khatri, V., & Brown, C. V. (2023). Designing data governance. *Communications of the ACM*, 66(1), 72-81. <https://doi.org/10.1145/3571723>
- Kim, J. K. (2021). Statistical assumptions of missing data and modern missing data handling techniques. *Communications of the Korean Statistical Society*, 28(4), 797–815.
- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley.
- Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley.
- Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3rd ed.). Wiley.
- Kumar Gupta, M., Verma, V., & Singh Verma, M. (2013). In-Memory Database Systems-A Paradigm Shift. *International Journal of Engineering Trends and Technology (IJETT)*, 6(6). <http://www.ijettjournal.orgwww.vedantatree.com>
- Kumrashan Indranil Iyer. (2021). Adaptive honeypots: Dynamic deception tactics in modern cyber defense. *International Journal of Science and Research Archive*, 4(1), 340–351. <https://doi.org/10.30574/ijrsra.2021.4.1.0210>
- Kundzewicz, Z. W., Kanae, S., Seneviratne, S. I., Handmer, J., Nicholls, N., Peduzzi, P., Dell’acqua, M., Kabat, P., Jiménez, B., Kreibich, H., Llasat, M. C., Mach, K., Matsuura, K., Nordbeck, R., & Zhou, D. (2014). Flood risk and climate change: global and regional perspectives. *Hydrological Sciences Journal*, 59(1), 1–28.
- Laudon, K. C. ., & Laudon, J. P. . (2020). *Management information systems : managing the digital firm*. Pearson.

- Laudon, K. C. ., & Laudon, J. P. . (2020). Management information systems : managing the digital firm. Pearson.
- Lee, E. A. (2008). Cyber Physical Systems: Design Challenges. UCB/EECS-2008-8.
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-8.html>
- Li, Y., Li, C., & Cui, Z. (2024a). Intelligent Database Triggers Enable Advanced Analysis of Data Recorded in Audit Logs. The International Conference on Cyber Security Intelligence and Analytics, 184–193.
- Little, R. J. A., & Rubin, D. B. (2019). Statistical analysis with missing data (3rd ed.). Wiley.
- Liu, X., Zhou, Q., Arulraj, J., & Orso, A. (2022). Automatic detection of performance bugs in database systems using equivalent queries. Proceedings of the 44th International Conference on Software Engineering, 225–236.
- Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). Geographic Information Systems and Science. John Wiley & Sons.
- Mahendra, D. A., & Winardi, S. (2023). Perancangan Realtime Database Firebase untuk IoT dan Unity Menggunakan Metode SDLC. Jurnal Ilmu Komputer Dan Bisnis, 14(2a), 72–82.
<https://doi.org/10.47927/jikb.v14i2a.525>
- Marakas, G. M. ., & O'Brien, J. A. . (2013). Introduction to information systems. McGraw-Hill/Irwin.
- Marakas, G. M. ., & O'Brien, J. A. . (2013). Introduction to information systems. McGraw-Hill/Irwin.
- Maurice, M. (2022, December 5). Introduction to Database (part1). Medium.
<https://medium.com/@michaelmaurice410/introduction-to-database-part1-db3ef6eb4c12>

- Microsoft (2023). Power BI Documentation. Retrieved from docs.microsoft.com
- Moroney, L. (2017). The Definitive Guide to Firebase. In The Definitive Guide to Firebase. Apress. <https://doi.org/10.1007/978-1-4842-2943-9>
- Morrison, G. S. (2022). Advancing a paradigm shift in evaluation of forensic evidence: The rise of forensic data science. *Forensic Science International: Synergy*, 5, 100270. <https://doi.org/https://doi.org/10.1016/j.fsisyn.2022.100270>
- Muhammad, Tole S, Imam R. (2024). K-means clustering as an imputation strategy for missing values in scholarship candidate data. *Jurnal Mantik*, Vol. 8, No. 4, 1656-1665.
- Negash, S. (2004). "Business Intelligence." *Communications of the Association for Information Systems*, 13(1), 177–195.
- Olabanji, S. O. (2023). Advancing cloud technology security: Leveraging high-level coding languages like Python and SQL for strengthening security systems and automating top control processes. *Journal of Scientific Research and Reports*, 29(9), 42–54.
- Özsu, M. T., & Valduriez, P. (2020). *Principles of Distributed Database Systems* (4th ed.). Springer.
- Peng, Z.-R. (2014). *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks*. John Wiley & Sons.
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454. <https://doi.org/10.1109/SURV.2013.042313.00197>
- Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414–454. <https://doi.org/10.1109/SURV.2013.042313.00197>

- Petride, S., Sotolongo, D., Michels, J., Witkowski, A., Haas, C., & Hughes, J. (2023). Lessons Learned from Efforts to Standardize Streaming In SQL. ArXiv Preprint ArXiv:2311.03476.
- Ponniah, P. (2010). *Data Warehousing Fundamentals for IT Professionals* (2nd ed.). Wiley.
- Porostimur. (2021). *Asah Skill Kejahatan Siber, Kasi Intel Kejari Halsel Ambil Sertifikasi CHFI*.
- Rahman, A., Marfai, M. A., & Wijaya, A. (2019). Spatial modeling of flood vulnerability using geographic information system (GIS) and analytical hierarchy process (AHP) in Semarang City, Indonesia. *IOP Conference Series: Earth and Environmental Science*, 284(1), 12033.
- Ramakrishnan, R., & Gehrke, J. (2003). *Database Management Systems* (3rd ed.). McGraw-Hill.
- Ramakrishnan, R., Gehrke, J., Derstadt, J., Selikoff, S., & Zhu, L. (2000). *DataBase Management Systems Solutions Manual*.
- Ramakrishnan, R., Gehrke, J., Derstadt, J., Selikoff, S., & Zhu, L. (2000). *DataBase Management Systems Solutions Manual*.
- Riadi, I., Umar, R., & Bernadisman, D. (2019). Analisis Forensik Database Menggunakan Metode Forensik Statis. *JURNAL SISTEM INFORMASI BISNIS*, 9(1), 9. <https://doi.org/10.21456/vol9iss1pp9-17>
- Rifqi, M. A., Husodo, A. Y., & Wijayanto, H. (2024). Development of SQL and NoSQL Database Integration Using Query Direct Access and Change Data Capture Approaches in an Android Application. *2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 671–679.
- Rochmadi, T., Fadlil, A., & Riadi, I. (2024). Tinjauan Pustaka Sistematis: Tantangan Dan Faktor-Faktor Pengembangan Kesiapan Forensik Digital. *CyberSecurity Dan Forensik Digital*, 7(2), 81–89. <https://doi.org/10.14421/csecurity.2024.7.2.4861>

- Ross, J. W., Beath, C. M., & Mocker, M. (2022). *Designed for digital: How to architect your business for sustained success*. MIT Press.
- Rouse, M. (2022). *Business Intelligence (BI)*. TechTarget. Retrieved from [techtarget.com](https://www.techtarget.com)
- Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3), 581-592.
- Rubin, D. B. (2020). *Multiple imputation for nonresponse in surveys*. John Wiley & Sons.
- Santayana, G., & Wilkes, M. V. (2004). *An Introduction to Database Systems*. Pearson Education, Inc.
- Sargiotis, D. (2024). Data security and privacy: Protecting sensitive information. In *Data governance: a guide* (pp. 217–245). Springer.
- Sarker, S., Chatterjee, S., & Xiao, X. (2023). How business intelligence creates business value. *Information Systems Journal*, 33(2), 434-461. <https://doi.org/10.1111/isj.12389>
- Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, 7(2), 147-177.
- Schumann, G. J. P., Bates, P. D., Neal, J. C., Andreadis, K. M., & بسيط, م. ع. (2016). Global flood hazard mapping. *Wiley Interdisciplinary Reviews: Water*, 3(4), 529–546.
- Shaik, B., & Chemuduru, D. K. (2023). Triggers. In B. Shaik & D. K. Chemuduru (Eds.), *Procedural Programming with PostgreSQL PL/pgSQL: Design Complex Database-Centric Applications with PL/pgSQL* (pp. 241–262). Apress. https://doi.org/10.1007/978-1-4842-9840-4_15
- Sharda, R., Delen, D., & Turban, E. (2021). *Business Intelligence, Analytics, and Data Science: A Managerial Perspective* (5th ed.). Pearson.
- Sharda, R., Delen, D., & Turban, E. (2021). *Business intelligence, analytics, and data science: A managerial perspective* (5th ed.). Pearson.

- Shearer, C. (2000). The CRISP-DM Model: The New Blueprint for Data Mining. *Journal of Data Warehousing*, 5(4), 13–22.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). *Database System Concepts* (6th ed.). McGraw-Hill.
- Smith, K. (2020). *Environmental Hazards: Assessing Risk and Reducing Disaster* (9th ed.). Routledge.
- Stair, R. M. ., & Reynolds, G. W. . (2018). *Principles of information systems : a managerial approach*. Cengage learning.
- Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson Education.
- Stonebraker, M., & Çetintemel, U. (2005). "One size fits all": An idea whose time has come and gone. *Proceedings of the 21st International Conference on Data Engineering*, 2-11.
- Szeliski, R. (2022). *Computer Vision: Algorithms and Applications* (2nd ed.). Springer. <https://doi.org/10.1007/978-3-030-34372-9>
- Tableau Software (2023). *Tableau Platform Overview*. Retrieved from tableau.com
- Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms* (2nd ed.). Prentice Hall.
- Templ, M., Alfons, A., & Filzmoser, P. (2016). Exploring incomplete data using visualization techniques. *Advances in Data Analysis and Classification*, 10(3), 385–405. <https://doi.org/10.1007/s11634-016-0250-9>
- Tempo. (2024). *Sederet Kasus Kebocoran Data Terbaru 6 Juta Data NPWP Diredas Bjorka, Siapa Tanggung Jawab?*
- Teorey, T., Lightstone, S., Nadeau, T., & Jagadish, V. ,H. (2011). *Database Modeling and Design*. Elsevier Inc.
- Turban, E., Sharda, R., & Delen, D. (2020). *Decision Support and Business Intelligence Systems* (11th ed.). Pearson.

- UNISDR. (2015). Sendai Framework for Disaster Risk Reduction 2015-2030. United Nations Office for Disaster Risk Reduction.
- Van Buuren, S. (2018). Flexible imputation of missing data (2nd ed.). CRC Press.
- Van Buuren, S. (2021). Flexible imputation of missing data. CRC press.
- Varga, M. (2009). Challenges of data management in always-on enterprise information systems. In *Always-On Enterprise Information Systems for Business Continuance: Technologies for Reliable and Scalable Operations* (pp. 109–128). IGI Global. <https://doi.org/10.4018/978-1-60566-723-2.ch007>
- Wang, R. Y., & Strong, D. M. (2022). Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 39(4), 1264-1290. <https://doi.org/10.1080/07421222.2022.2089156>
- Wardhani, W. (2018). Dampak perubahan tata ruang terhadap peningkatan risiko banjir di kawasan perkotaan (Studi Kasus: Kota Semarang). *Jurnal Pembangunan Wilayah & Kota*, 14(1), 1–12.
- White, T. (2015). *Hadoop: The Definitive Guide* (4th ed.). O'Reilly Media.
- Wintolo, H. (2010). Sinkronisasi Data pada Tabel yang Tersimpan di Dua Database Server yang Berbeda. *Angkasa Jurnal Ilmiah Bidang Teknologi*, 2(1).
- Wintolo, H., Riadi, I., & Yudhana, A. (2025). Analisis Deteksi Penyusup pada Layanan Open Journal System Menggunakan Metode Network Forensic Development Life Cycle. *SKANIKA: Sistem Komputer Dan Teknik Informatika*, 8(1), 133–144. <https://doi.org/10.36080/skanika.v8i1.3284>
- Wixom, B. H., & Watson, H. J. (2022). The BI-based organization. *International Journal of Business Intelligence Research*, 13(1), 1-16. <https://doi.org/10.4018/IJBIR.2022010101>

- Wu, Z., Liang, J., Fu, J., Deng, W., & Jiang, Y. (n.d.). DDLUMOS: Understanding and Detecting Atomic DDL Bugs in DBMSs.
- Xu, J., Liu, L., Zhang, X., & Liu, Z. (2022). An overview of missing data imputation methods for wireless sensor networks. *Sensors*, 22(3), 826
- Yuliansyah, H., & Hermawan, A. F. K. F. F. (2014). PERANCANGAN REPLIKASI BASIS DATA MYSQL DENGAN MEKANISME PENGAMANAN MENGGUNAKAN SSL ENCRYPTION. *Jurnal Informatika*, Vol 8, No 1 (2014): Januari, 65–72. <http://journal.uad.ac.id/index.php/JIFO/article/view/2081/1331>
- Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., & others. (2016). Apache Spark: A Unified Engine for Big Data Processing. *Communications of the ACM*, 59(11), 56–65. <https://doi.org/10.1145/2934664>
- Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., & others. (2016). Apache Spark: A Unified Engine for Big Data Processing. *Communications of the ACM*, 59(11), 56–65. <https://doi.org/10.1145/2934664>
- Zakarnah, S. K. A. (2023). Forensic Investigation in SQL Server Database Using Temporal Tables & Extended Events Artifacts. *International Journal of Applied Sciences and Smart Technologies*, 5(1), 1–16.
- Zhang, S., Chen, T., & Yu, P. S. (2020). Big Data and Missing Value Imputation: Methods and Challenges. *IEEE Transactions on Big Data*, 6(1), 34-45. <https://doi.org/10.1109/TBDDATA.2019.2919930>
- Zhao, Y., Long, Q., & Li, Y. (2022). Machine learning approaches for missing data imputation. *Journal of Data Science*, 20(1), 45-68.