

Solusi Database di Era Digital



Big Data *dan* NoSQL

Solusi Database di Era Digital

Dr. Ir. Mohammad Givi Efgivia, M.Kom.

BIG DATA DAN NoSQL: SOLUSI DATABASE DI ERA DIGITAL

Penulis:

Dr. Ir. Mohammad Givi Efgivia, M.Kom

Desain Cover:

Andi Juliandi

Sumber Ilustrasi:

www.freepik.com

Tata Letak:

Handarini Rohana

Editor:

Neneng Sri Wahyuni

ISBN:

978-634-246-529-5

Cetakan Pertama:

Desember, 2025

Hak Cipta Dilindungi Oleh Undang-Undang

by Penerbit Widina Media Utama

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

PENERBIT:

WIDINA MEDIA UTAMA

Komplek Puri Melia Asri Blok C3 No. 17 Desa Bojong Emas
Kec. Solokan Jeruk Kabupaten Bandung, Provinsi Jawa Barat

Anggota IKAPI No. 519/JBA/2025

Website: www.penerbitwidina.com

Instagram: [@penerbitwidina](https://www.instagram.com/penerbitwidina)

Telepon (022) 87355370

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan buku ini yang berjudul “Dasar-Dasar Database dan Penerapan SQL–NoSQL dalam Era Modern”. Buku ini disusun dengan tujuan memberikan pemahaman komprehensif kepada mahasiswa, dosen, praktisi, maupun pembaca umum mengenai konsep dasar database, mulai dari teori hingga praktik, dengan pendekatan yang sistematis dan mudah dipahami.

Dalam era transformasi digital saat ini, data telah menjadi aset yang sangat berharga. Peran database sebagai media penyimpanan, pengelolaan, dan pengolahan data semakin penting, baik di bidang akademik, bisnis, maupun industri. Oleh karena itu, pemahaman mengenai database, termasuk SQL (Structured Query Language) maupun NoSQL (Not Only SQL), menjadi kompetensi mendasar yang wajib dimiliki oleh generasi digital.

Buku ini terbagi menjadi beberapa bagian utama. Bagian awal membahas konsep dasar database, terminologi penting, hingga relasi dan keamanan data. Bagian selanjutnya mengupas perintah dasar SQL serta contoh implementasi query dalam berbagai skenario. Selain itu, buku ini juga memperkenalkan konsep NoSQL, arsitektur polyglot persistence, serta penerapannya dalam sistem modern seperti e-commerce. Untuk menunjang pembelajaran, disertakan pula lampiran berupa glossary istilah, referensi cepat sintaks SQL, panduan instalasi DBMS (MySQL dan PostgreSQL), serta jawaban latihan pilihan.

Ucapan terima kasih penulis sampaikan kepada semua pihak yang telah memberikan dukungan, baik berupa dorongan moral maupun bantuan teknis, dalam penyusunan buku ini. Semoga buku ini dapat menjadi referensi yang bermanfaat bagi pembaca dalam memahami, mempelajari, dan menerapkan konsep-konsep database secara lebih baik.

Akhir kata, penulis menyadari bahwa buku ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang konstruktif sangat penulis harapkan demi perbaikan dan penyempurnaan di masa mendatang.

Selamat membaca, semoga bermanfaat.

DAFTAR ISI

KATA PENGANTAR	iii
DAFTAR ISI	v

BAGIAN I: FONDASI KONSEPTUAL DATABASE

BAB 1: PENGENALAN DUNIA DATABASE	2
A. Apa Itu Data, Informasi, dan Database?	5
B. Masalah Pada Sistem Berbasis File Tradisional	11
C. Pengenalan DBMS: Peran dan Manfaatnya	17
D. Komponen-Komponen Utama Dalam Sebuah DBMS.....	23
E. Pengenalan Singkat Jenis-Jenis Database.....	27
F. Peran dan Tanggung Jawab Profesi di Dunia Database	36
BAB 2: ARSITEKTUR DAN MODEL DATA.....	50
A. Arsitektur Tiga-Schema Dbms	53
B. Model Data	67
C. Keunggulan Model Relasional	88
D. Konsep Independensi Data.....	94
E. Bahasa Database: DDL VS DML	101
BAB 3: MODEL DATA RELASIONAL	111
A. Konsep Dasar	114
B. Karakteristik Relation yang Baik	117
C. Konsep Kunci dalam Database	124
D. Integritas Data	136
E. Operasi Dasar Relasional	148

BAGIAN II:

DESAIN DATABASE DENGAN PEMODELAN KONSEPTUAL & LOGIS

BAB 4: PEMODELAN KONSEPTUAL	
DENGAN ENTITY-RELATIONSHIP (ER)	159
A. Komponen Diagram ER: Entitas, Atribut, Tipe Entitas	166

B. Jenis-Jenis Relationship	178
C. Atribut pada Relationship	191
D. Teknik Pembuatan Diagram ER yang Efektif	194
E. Studi Kasus Komprehensif	203
BAB 5: NORMALISASI DATABASE.....	219
A. Tujuan Normalisasi	222
B. Konsep Functional Dependency	230
C. Bentuk Normal Pertama (1NF)	237
D. Bentuk Normal Kedua (2NF)	243
E. Bentuk Normal Ketiga (3NF)	250
F. Pengenalan Boyce-Codd Normal Form (BCNF)	257
G. Pro dan Kontra Normalisasi	264
 BAGIAN III: BAHASA DAN IMPLEMENTASI (SQL)	
BAB 6: PENGENALAN SQL DAN DDL	286
A. Standar SQL dan Implementasi	292
B. Membuat Tabel (Create Table)	298
C. Tipe Data SQL	302
D. Constraint	309
E. Modifikasi Tabel (Alter Table)	319
F. Menghapus Tabel dan Database	325
BAB 7: MEMANIPULASI DATA DENGAN DML	335
A. Perintah `Insert` Untuk Menambah Data	337
B. Perintah Select Yang Komprehensif	348
C. Join: Menggabungkan Tabel	368
D. Subquery (Nested Query)	384
E. Update: Mengubah Data Dengan Subquery dan Nested Select.....	387
F. Delete: Menghapus Data Dengan Kondisi.....	395
BAB 8: PENGEMBANGAN LANJUTAN – VIEW, PROCEDURE, FUNCTION, TRIGGER, DAN TRANSAKSI	422
A. Membuat Dan Menggunakan View	424

B. Stored Procedures: Konsep dan Keuntungan.....	427
C. Functions Dalam SQL.....	429
D. Triggers.....	432
E. Pengenalan Transaksi, Begin Transaction, Commit, Rollback.....	435
BAB 9: PENGATURAN AKSES DAN KEAMANAN (DCL)	446
A. Membuat User (`Create User`).....	451
B. Memberikan Izin Dengan Grant	457
C. Mencabut Izin Dengan Revoke.....	461
D. Peran (Roles) Dalam Database	467
 BAGIAN IV: ADMINISTRASI DAN OPTIMASI DATABASE	
BAB 10: PENGANTAR ADMINISTRASI DATABASE (DBA)	478
A. Tugas-Tugas Seorang Database Administrator	478
B. Perencanaan Kapasitas dan Instalasi DBMS (Overview)	487
C. Backup dan Recovery: Strategi dan Teknik	497
D. Melakukan Backup dan Restore (Praktik Sederhana)	510
E. Monitoring Kesehatan dan Kinerja Database.....	516
BAB 11: OPTIMASI KINERJA DATABASE).....	527
A. Faktor Yang Mempengaruhi Kinerja Database.....	527
B. Pentingnya Indexing.....	532
C. Keuntungan Penggunaan Index.....	541
D. Kerugian Penggunaan Index.....	544
E. Membuat Index.....	547
F. Menggunakan Explain	550
G. Tips Menulis Query Optimal.....	561
 BAGIAN V: MELAMPAUI DATABASE RELASIONAL	
BAB 12: PENGENALAN DATABASE NOSQL	576
A. Keterbatasan Database Relasional dan Munculnya NoSQL	576
B. Karakteristik Database NoSQL (BASE vs ACID)	588
C. Jenis-Jenis Database NoSQL:	597

D. Pola Pikir NoSQL: Denormalization dan Duplication	623
E. Kapan Memilih SQL vs. NoSQL?	633
LAMPIRAN.....	639
Lampiran A: Glossary Istilah-Istilah Database	639
Lampiran B: Referensi Cepat Sintaks SQL	641
Lampiran C: Setup dan Instalasi DBMS (MySQL/PostgreSQL)	645
Lampiran D: Jawaban Latihan Pilihan	648
DAFTAR PUSTAKA.....	650
INDEKS	651

BAGIAN I:

FONDASI KONSEPTUAL DATABASE

BAB 1

PENGENALAN DUNIA DATABASE

Daftar Isi Bab 1

1. **Pendahuluan: Mengapa Data Itu Penting?**

Menjelaskan posisi data dalam kehidupan sehari-hari serta peranannya dalam dunia bisnis, teknologi, dan akademik.

2. **Definisi Database**

Pemahaman dasar mengenai apa itu database, perbedaannya dengan sekadar kumpulan data biasa, serta konsep fundamental yang mendasarinya.

3. **Sejarah dan Perkembangan Database**

Menelusuri perjalanan evolusi database dari sistem manual hingga sistem modern berbasis cloud dan big data.

4. **Peran Database dalam Kehidupan Modern**

Bagaimana database menjadi fondasi berbagai layanan digital, mulai dari aplikasi e-commerce, perbankan, hingga media sosial.

5. **Jenis-Jenis Database**

Pengenalan pada beberapa jenis database populer, seperti relational database, NoSQL, graph database, dan time-series database.

6. **Manfaat Database**

Menjelaskan keuntungan menggunakan database, antara lain efisiensi, kecepatan, keamanan, serta kemudahan dalam analisis data.

7. **Tantangan di Dunia Database**

Membahas isu-isu yang muncul, seperti keamanan data, skala besar, dan manajemen kompleksitas sistem.

8. **Penutup Bab: Refleksi Awal**

Menyimpulkan pentingnya memahami konsep database sebelum melangkah pada pembahasan teknis lebih lanjut.

BAB 2

ARSITEKTUR DAN MODEL DATA

Pengantar

Dalam dunia sistem informasi modern, database tidak lagi sekadar dipandang sebagai **tempat menyimpan data**. Fungsi database telah berevolusi menjadi **sistem kompleks** yang harus mampu mengelola berbagai kebutuhan: efisiensi pengolahan, keamanan akses, integritas data, serta kemudahan dalam penggunaannya. Bayangkan sebuah perusahaan e-commerce dengan jutaan transaksi harian; tanpa sistem database yang terorganisir dan terkelola dengan baik, data hanya akan menjadi sekumpulan catatan tak bernilai. Karena itu, bab ini akan membahas kerangka konseptual yang menjadi dasar bagaimana database dirancang dan digunakan.

Tujuan utama dari Bab 2 adalah memperkenalkan **arsitektur tiga-schema DBMS**, berbagai **model data** yang pernah berkembang, serta prinsip fundamental seperti **independensi data** dan **bahasa database**. Pemahaman mengenai hal-hal ini sangat penting sebelum pembaca masuk ke ranah teknis berupa bahasa query, terutama SQL, yang akan dibahas di bab-bab selanjutnya. Dengan kata lain, bab ini menjadi **jembatan teoritis** yang menuntun mahasiswa dan praktisi agar memahami “mengapa” dan “bagaimana” suatu database dibangun, bukan hanya “cara memakainya”.

Tujuan

Ada lima tujuan pembelajaran yang diharapkan tercapai setelah membaca bab ini:

1. Memahami Arsitektur Tiga-Schema DBMS

Model tiga-schema (internal, konseptual, eksternal) merupakan konsep dasar yang menjelaskan bagaimana data dilihat, dikelola, dan disajikan pada pengguna. Dengan memahami arsitektur ini, pembaca dapat menyadari bahwa antara penyimpanan fisik data

BAB 3

MODEL DATA RELASIONAL

Pengantar Model Data Relasional

Model data relasional merupakan salah satu pencapaian paling signifikan dalam sejarah sistem basis data modern. Model ini pertama kali diperkenalkan oleh Edgar F. Codd pada tahun 1970 melalui makalah berjudul *“A Relational Model of Data for Large Shared Data Banks”*. Dalam makalah tersebut, Codd mengusulkan sebuah pendekatan baru untuk mengorganisasi data yang didasarkan pada teori himpunan dan logika predikat. Pada masa itu, sistem basis data yang umum digunakan adalah model hirarki dan model jaringan yang memang berfungsi, tetapi memiliki keterbatasan besar dalam hal fleksibilitas dan pemeliharaan.

Latar Belakang Munculnya Model Relasional

Sebelum tahun 1970, perusahaan dan organisasi besar mengandalkan database hirarki (seperti IMS dari IBM) atau database jaringan (seperti CODASYL). Keduanya menyimpan data dalam bentuk pointer dan hubungan langsung antar-record. Hal ini membuat data sulit untuk dimodifikasi, dan setiap perubahan struktur sering memerlukan modifikasi besar pada aplikasi. Dengan kata lain, data tidak independen dari program yang mengaksesnya.

Codd kemudian mengusulkan model relasional untuk mengatasi masalah ini. Ia memperkenalkan gagasan bahwa data sebaiknya direpresentasikan dalam bentuk **relasi**, yang secara sederhana dapat dipandang sebagai sebuah tabel dua dimensi. Setiap tabel terdiri dari baris (tuple) dan kolom (atribut), di mana setiap kolom memiliki domain nilai tertentu. Dengan menggunakan pendekatan ini, manipulasi data dapat dilakukan menggunakan operasi matematis standar seperti seleksi, proyeksi, dan gabungan (*join*).

BAB 4

PEMODELAN KONSEPTUAL DENGAN ENTITY-RELATIONSHIP

Pendahuluan

Dalam dunia sistem basis data, salah satu tahap yang sering kali menentukan keberhasilan atau kegagalan implementasi adalah tahap pemodelan data. Sebuah sistem database bukan sekadar kumpulan tabel atau sekumpulan query yang bisa langsung digunakan. Ia adalah representasi dari realitas bisnis, organisasi, maupun proses operasional yang kompleks. Jika pemodelan dilakukan dengan sembarangan, maka implementasi database akan menghasilkan struktur yang tumpang tindih, sulit dipelihara, dan rawan inkonsistensi data. Oleh karena itu, sebelum terjun ke tahap implementasi teknis—misalnya mendefinisikan tabel dalam SQL atau menyusun query—pemodelan konseptual harus dilakukan dengan cermat.

Salah satu pendekatan yang paling populer dan banyak digunakan dalam pemodelan konseptual adalah Entity-Relationship (ER) Model. Model ini diperkenalkan oleh Peter Chen pada tahun 1976, dan hingga kini tetap menjadi fondasi penting dalam desain database modern. Diagram ER memungkinkan pengembang maupun analis sistem untuk menyusun representasi visual dari struktur data, hubungan antar entitas, serta atribut-atribut yang melekat pada entitas tersebut. Dengan kata lain, ER diagram adalah jembatan komunikasi antara kebutuhan bisnis dengan solusi teknis dalam bentuk database.

Pentingnya Diagram ER dalam Sistem Basis Data

Diagram ER memiliki peran krusial karena ia menawarkan representasi **abstraksi tingkat tinggi** dari data. Abstraksi ini membuat diagram ER mudah dipahami oleh berbagai pihak—baik itu analis sistem, developer, maupun pemangku kepentingan non-teknis seperti manajer atau pengguna akhir. Sebagai contoh, dalam perancangan sistem perpustakaan, diagram ER dapat menggambarkan dengan jelas

BAB 5

NORMALISASI DATABASE

Pendahuluan

Dalam merancang sebuah sistem informasi, kualitas desain basis data (database) menjadi faktor yang sangat menentukan keberhasilan implementasi sistem tersebut. Sebuah database yang terstruktur dengan baik bukan hanya memudahkan penyimpanan data, tetapi juga menjamin konsistensi, integritas, serta efisiensi dalam pengelolaan data jangka panjang. Salah satu pendekatan konseptual yang digunakan untuk mencapai desain database yang optimal adalah **normalisasi**.

Normalisasi merupakan proses sistematis untuk mengorganisasi tabel dan relasi dalam database agar terhindar dari masalah redundansi dan anomali data. Dengan kata lain, normalisasi berfungsi sebagai “alat bantu” untuk memastikan bahwa struktur tabel yang kita bangun mampu merepresentasikan kebutuhan data secara logis, tanpa menimbulkan potensi masalah di kemudian hari. Proses ini dilakukan melalui serangkaian tahapan yang dikenal dengan bentuk normal (normal forms), mulai dari bentuk normal pertama (1NF) hingga bentuk yang lebih tinggi seperti Boyce-Codd Normal Form (BCNF).

Masalah Umum dalam Desain Database

Sebelum memahami lebih jauh mengenai normalisasi, penting untuk menyoroti permasalahan yang sering timbul apabila database tidak dirancang dengan baik:

1. Redundansi Data (Data Redundancy)

Redundansi terjadi ketika informasi yang sama tersimpan berulang kali dalam tabel atau di beberapa tabel. Misalnya, dalam tabel pesanan, jika data pelanggan (nama, alamat, nomor telepon) dicatat berulang di setiap transaksi, maka semakin banyak transaksi, semakin banyak pula data pelanggan yang tersimpan berkali-kali. Redundansi ini menyebabkan pemborosan ruang penyimpanan sekaligus meningkatkan risiko inkonsistensi.

BAGIAN III: BAHASA DAN IMPLEMENTASI (SQL)

BAB 6

PENGENALAN SQL DAN DDL

Pendahuluan Bab 6: Pengenalan SQL dan DDL

Dalam perjalanan mempelajari sistem basis data relasional, pemahaman mengenai bahasa yang digunakan untuk berkomunikasi dengan database menjadi aspek yang tidak dapat diabaikan. Bab ini bertujuan untuk memperkenalkan Structured Query Language (SQL), sebuah bahasa standar yang digunakan hampir di semua sistem manajemen basis data relasional (Relational Database Management System/RDBMS). Fokus utama pembahasan berada pada Data Definition Language (DDL), yaitu bagian dari SQL yang berfungsi untuk mendefinisikan struktur database, mulai dari pembuatan database, pembuatan tabel, pengaturan tipe data, hingga pemberian constraint yang mengatur validitas data.

Tujuan dari bab ini adalah agar pembaca memahami bagaimana cara mendesain dan membangun sebuah database dari awal menggunakan perintah SQL. Dengan mempelajari DDL, seorang desainer atau administrator database dapat memastikan bahwa struktur data yang disimpan sudah sesuai dengan kebutuhan organisasi, memiliki aturan integritas yang jelas, serta dapat dimodifikasi sesuai perkembangan. Pemahaman ini menjadi pondasi penting sebelum melangkah ke tahap pengolahan data yang lebih dinamis menggunakan Data Manipulation Language (DML).

SQL dalam Konteks Database Relasional

SQL lahir dari konsep relational algebra dan relational calculus yang diperkenalkan oleh Edgar F. Codd pada tahun 1970. Bahasa ini kemudian berkembang menjadi standar internasional yang digunakan oleh berbagai vendor, seperti MySQL, PostgreSQL, Oracle, Microsoft SQL Server, dan lain-lain. SQL dirancang agar pengguna dapat berinteraksi dengan data dalam bentuk tabel (relasi) dengan cara yang sederhana namun sangat kuat.

BAB 7

MEMANIPULASI DATA DENGAN DML

Pendahuluan DML

Dalam sistem manajemen basis data relasional (Relational Database Management System/RDBMS), SQL (Structured Query Language) menjadi bahasa utama yang digunakan untuk mengelola data. Secara garis besar, SQL dibagi menjadi dua kelompok perintah besar, yaitu Data Definition Language (DDL) dan Data Manipulation Language (DML). DDL berfokus pada aspek struktur, seperti membuat tabel, mengubah skema, atau menghapus objek. Sebaliknya, DML berperan langsung dalam memanipulasi isi tabel yang berupa baris-baris data. Dengan kata lain, DML adalah “jantung” dari interaksi sehari-hari dengan sebuah database.

Perbedaan utama antara DDL dan DML dapat diibaratkan seperti membangun rumah versus mengatur furnitur di dalamnya. DDL menyediakan dinding, lantai, dan ruangan—yaitu struktur tempat data disimpan. DML, di sisi lain, memungkinkan pengguna untuk menambahkan isi ke dalam rumah tersebut (INSERT), mengambil atau melihat isi (SELECT), mengubah posisi atau kondisi furnitur (UPDATE), serta menghapus barang yang tidak lagi diperlukan (DELETE). Dari perumpamaan ini dapat dipahami bahwa tanpa DML, sebuah tabel hanya akan menjadi wadah kosong tanpa informasi yang bermanfaat.

Pentingnya DML dalam pekerjaan sehari-hari tidak dapat dilebih-lebihkan. Hampir semua operasi bisnis, akademik, maupun riset, mengandalkan kemampuan untuk mengambil dan memodifikasi data. Misalnya, dalam aplikasi e-commerce, setiap kali pelanggan menambahkan produk ke keranjang, maka perintah INSERT digunakan untuk mencatat transaksi baru. Ketika pengguna ingin melihat daftar produk yang tersedia, sistem menggunakan perintah SELECT. Jika pelanggan mengubah alamat pengiriman, maka sistem akan mengeksekusi perintah UPDATE. Akhirnya, jika pelanggan membatalkan pesanan, data transaksi bisa dihapus melalui perintah DELETE. Siklus

BAB 8

PENGEMBANGAN LANJUTAN – VIEW, PROCEDURE, FUNCTION, TRIGGER, DAN TRANSAKSI

Pendahuluan

Bab 8 menekankan pengembangan lanjutan SQL, yang melampaui kemampuan dasar DDL dan DML. Setelah peserta memahami bagaimana membuat, memodifikasi, dan menampilkan data menggunakan perintah dasar SQL, bab ini memperkenalkan konsep-konsep yang memungkinkan pengelolaan database menjadi lebih fleksibel, aman, dan efisien. Fokus utamanya meliputi view, stored procedure, functions, triggers, dan transaksi—fitur-fitur yang mendukung otomatisasi, abstraksi, dan integritas data dalam sistem basis data relasional.

Salah satu fitur penting adalah view, yaitu tabel virtual yang dibuat dari satu atau beberapa tabel dasar menggunakan query SELECT. View berfungsi sebagai lapisan abstraksi: pengguna atau aplikasi dapat menampilkan data yang relevan tanpa mengakses tabel asli secara langsung. Keuntungan utama view adalah memudahkan pengguna untuk melihat data yang telah difilter atau digabung, meningkatkan keamanan karena hanya kolom tertentu yang ditampilkan, dan mengurangi duplikasi query yang kompleks. Misalnya, dalam sistem penjualan, kita dapat membuat view LaporanPenjualan yang menampilkan nama pelanggan, produk, jumlah, dan total belanja, tanpa mengubah tabel asli Pesanan, DetailPesanan, dan Pelanggan. Dengan view, pengguna cukup melakukan `SELECT * FROM LaporanPenjualan` untuk mendapatkan informasi yang dibutuhkan, sehingga kode lebih ringkas dan mudah dipelihara.

Selanjutnya, stored procedure dan functions merupakan mekanisme untuk menyimpan logika SQL yang sering digunakan. Stored procedure adalah sekumpulan perintah SQL yang dapat dijalankan

BAB 9

PENGATURAN AKSES DAN KEAMANAN (DCL)

Pendahuluan Data Control Language (DCL)

Dalam dunia sistem basis data modern, pengaturan akses dan keamanan menjadi salah satu aspek paling penting yang tidak bisa diabaikan. Seiring dengan meningkatnya jumlah pengguna yang berinteraksi dengan database, baik dalam skala perusahaan kecil maupun organisasi besar, kebutuhan untuk menjaga integritas, kerahasiaan, dan ketersediaan data semakin mendesak. Di sinilah Data Control Language (DCL) dalam SQL memainkan peran kunci. DCL merupakan salah satu sub-bahasa SQL yang secara khusus digunakan untuk mengatur hak akses pengguna terhadap objek-objek dalam database, seperti tabel, view, prosedur, maupun skema.

Berbeda dengan Data Definition Language (DDL) yang berfungsi untuk mendefinisikan struktur basis data (misalnya CREATE, ALTER, DROP), dan Data Manipulation Language (DML) yang dipakai untuk mengelola data (misalnya SELECT, INSERT, UPDATE, DELETE), DCL tidak berhubungan langsung dengan isi atau struktur data. DCL lebih berfokus pada pengendalian siapa yang boleh melakukan apa di dalam sistem. Dengan kata lain, DCL adalah lapisan kontrol yang memastikan bahwa hanya pengguna yang berhak yang bisa mengakses atau memodifikasi data tertentu.

Pentingnya keamanan dalam sistem multi-user tidak bisa dipandang remeh. Dalam sebuah organisasi, sering kali ada banyak peran berbeda: administrator, developer, analis data, staf operasional, hingga pengguna akhir yang hanya memerlukan akses baca. Jika seluruh pengguna diberikan akses penuh tanpa pengendalian, maka potensi terjadinya penyalahgunaan data akan sangat besar. Bayangkan jika seorang staf entry data dapat menghapus tabel utama hanya karena tidak adanya pembatasan hak akses. Oleh karena itu, DCL hadir untuk memastikan adanya prinsip least privilege, yaitu setiap pengguna

BAGIAN IV:
ADMINISTRASI DAN
OPTIMASI DATABASE

BAB 10

PENGANTAR ADMINISTRASI DATABASE (DBA)

A. TUGAS-TUGAS SEORANG DATABASE ADMINISTRATOR

Definisi Database Administrator (DBA) dan Peran Strategisnya

Database merupakan salah satu aset terpenting dalam sebuah organisasi modern. Hampir seluruh aktivitas bisnis, mulai dari transaksi keuangan, pengelolaan data pelanggan, operasional produksi, hingga analisis bisnis berbasis data besar (big data analytics), sangat bergantung pada ketersediaan dan keakuratan informasi yang tersimpan dalam sistem database. Karena itu, keberadaan seorang Database Administrator (DBA) menjadi krusial. DBA bukan sekadar pengelola teknis, tetapi juga berperan strategis sebagai penjaga, pengatur, sekaligus pengembang infrastruktur data dalam organisasi.

Definisi Database Administrator (DBA)

Secara umum, Database Administrator (DBA) adalah individu atau tim yang bertanggung jawab atas instalasi, konfigurasi, pengelolaan, pemeliharaan, keamanan, dan optimalisasi sistem database. DBA memastikan bahwa database selalu tersedia, handal, aman, serta mampu memberikan performa optimal dalam mendukung kebutuhan bisnis maupun operasional.

Definisi ini mencakup dua aspek utama:

1. Aspek teknis, yaitu penguasaan sistem manajemen basis data (DBMS) seperti MySQL, PostgreSQL, Oracle Database, atau SQL Server, serta pemahaman infrastruktur server dan jaringan.
2. Aspek strategis, yaitu peran DBA dalam mendukung pencapaian tujuan organisasi dengan menjaga integritas, keamanan, dan ketersediaan data.

BAB 11

OPTIMASI KINERJA DATABASE

A. FAKTOR YANG MEMPENGARUHI KINERJA DATABASE (Bagian 1)

Definisi Kinerja Database dan Pentingnya

Kinerja database (database performance) merujuk pada kemampuan suatu sistem manajemen basis data (DBMS) dalam merespons permintaan pengguna dengan cepat, efisien, dan konsisten. Kinerja ini mencakup berbagai aspek, mulai dari waktu eksekusi query, kecepatan proses transaksi, hingga stabilitas sistem ketika menghadapi beban yang tinggi. Dalam lingkungan bisnis modern, database merupakan tulang punggung operasional: transaksi keuangan, sistem e-commerce, aplikasi mobile, maupun analitik data semuanya bergantung pada akses data yang cepat.

Jika kinerja database menurun, dampaknya bisa sangat signifikan. Misalnya, keterlambatan satu detik dalam transaksi e-commerce dapat menurunkan tingkat konversi penjualan. Di sisi lain, dalam aplikasi kritis seperti sistem perbankan atau rumah sakit, performa yang buruk dapat berimplikasi pada kegagalan layanan, bahkan berisiko terhadap keamanan data. Oleh karena itu, menjaga kinerja database adalah faktor krusial untuk keberlangsungan layanan digital.

Faktor Utama: Hardware (CPU, RAM, Storage, Jaringan)

Kinerja database tidak hanya ditentukan oleh optimasi query atau desain skema, tetapi juga oleh sumber daya perangkat keras (hardware) yang mendasarinya. Empat komponen utama yang paling berpengaruh adalah CPU, RAM, storage, dan jaringan.

1. CPU (Central Processing Unit)

CPU bertugas memproses instruksi dan perhitungan dalam eksekusi query. Query yang kompleks dengan banyak join, agregasi, atau subquery akan menuntut daya komputasi yang besar. Jika CPU

**BAGIAN V:
MELAMPAUI
DATABASE RELASIONAL**

BAB 12

PENGENALAN DATABASE NOSQL

A. KETERBATASAN DATABASE RELASIONAL DAN MUNCULNYA NOSQL

Pendahuluan – Sejarah Perkembangan Database Relasional (SQL), Kekuatan RDBMS, dan Tantangan Skala Besar

Perkembangan teknologi database telah menjadi salah satu tonggak utama dalam dunia komputasi modern. Sejak era 1970-an, ketika Edgar F. Codd memperkenalkan model relasional dalam publikasi legendarisnya “*A Relational Model of Data for Large Shared Data Banks*”, paradigma penyimpanan data mulai bergeser dari sistem hirarki dan jaringan (network database) menuju model relasional yang lebih sederhana, fleksibel, dan sistematis. Model relasional ini menggunakan Structured Query Language (SQL) sebagai bahasa standar untuk mendefinisikan, memanipulasi, dan mengakses data. SQL kemudian menjadi fondasi bagi hampir semua Relational Database Management System (RDBMS) yang mendominasi industri teknologi selama beberapa dekade.

Dalam RDBMS, data disimpan dalam bentuk tabel yang terdiri dari baris (row/tuple) dan kolom (field/attribute). Struktur ini memungkinkan representasi data yang konsisten, mudah dipahami, serta mendukung operasi matematis berbasis *relational algebra*. Keunggulan utama dari RDBMS adalah kemampuannya menjaga integritas data melalui aturan constraints (primary key, foreign key, unique, check) serta dukungan penuh terhadap prinsip ACID (Atomicity, Consistency, Isolation, Durability). Dengan ACID, transaksi dalam database dipastikan berjalan aman: jika salah satu operasi gagal, maka seluruh transaksi dibatalkan (rollback), sehingga data tetap konsisten. Hal ini menjadikan RDBMS pilihan utama bagi aplikasi kritis seperti perbankan, sistem keuangan, kesehatan, dan aplikasi pemerintahan.

LAMPIRAN

Lampiran A: Glossary Istilah-Istilah Database

Definisi Istilah Dasar

- Database: Kumpulan data yang terorganisir dan tersimpan secara sistematis agar mudah diakses, dikelola, dan diperbarui.
- DBMS (Database Management System): Perangkat lunak yang digunakan untuk membuat, mengelola, dan memanipulasi database.
- Schema: Struktur logis database yang mendefinisikan bagaimana data diorganisasikan (tabel, relasi, constraint).
- Table: Struktur dalam database yang menyimpan data dalam bentuk baris dan kolom.
- Row (Record/Tuple): Satu entri data dalam tabel.
- Column (Field/Attribute): Atribut atau kategori data dalam tabel.

Konsep Relasi

- Primary Key: Kolom unik yang digunakan untuk mengidentifikasi setiap baris pada tabel.
- Foreign Key: Kolom yang menghubungkan satu tabel dengan tabel lain (relasi).
- Join: Operasi untuk menggabungkan data dari dua atau lebih tabel berdasarkan kolom terkait.

Istilah NoSQL

- Document: Format penyimpanan data dalam bentuk dokumen (biasanya JSON/BSON).
- Collection: Kumpulan dokumen dalam database NoSQL, mirip tabel dalam RDBMS.
- Node: Unit dasar penyimpanan atau pemrosesan dalam database terdistribusi.
- Edge: Hubungan antar node dalam database graf.

DAFTAR PUSTAKA

- Connolly, T., & Begg, C. (2015). *Database systems: A practical approach to design, implementation, and management* (6th ed.). Pearson Education.
- Date, C. J. (2004). *An introduction to database systems* (8th ed.). Addison-Wesley.
- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson Education.
- Hellerstein, J. M., Stonebraker, M., & Hamilton, J. (2007). Architecture of a database system. *Foundations and Trends in Databases*, 1(2), 141–259. <https://doi.org/10.1561/19000000002>
- Kroenke, D. M., & Auer, D. J. (2014). *Database concepts* (7th ed.). Pearson Higher Ed.
- Pokorný, J. (2015). NoSQL databases: A step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1), 69–82. <https://doi.org/10.1108/IJWIS-01-2013-0004>
- Pratt, P. J., & Last, M. B. (2015). *Concepts of database management* (8th ed.). Cengage Learning.
- 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. IEEE. <https://doi.org/10.1109/ICDE.2005.1>
- Sadalage, P. J., & Fowler, M. (2012). *NoSQL distilled: A brief guide to the emerging world of polyglot persistence*. Addison-Wesley.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database system concepts* (7th ed.). McGraw-Hill.
- Widenius, M., Axmark, D., & Arno, K. (2002). *MySQL reference manual: Documentation from the source*. O'Reilly Media.

INDEKS

A

- ACID (Atomicity, Consistency, Isolation, Durability), 22
- Agregasi, fungsi, 25
- ALTER TABLE, 25

B

- BASE (Basically Available, Soft State, Eventual Consistency), 22
- Backup database, 28
- Basis Data (Database), 21
- BIGINT, tipe data, 11

C

- Caching, 20
- CHAR, tipe data, 11
- Column (Kolom), 21
- Collection (NoSQL), 22
- CREATE TABLE, 24
- CRUD (Create, Read, Update, Delete), 24

D

- Database, 21
- DBMS (Database Management System), 21
- DELETE, 24
- Denormalisasi, 22
- Document (NoSQL), 22
- DROP TABLE, 25

E

- Edge (NoSQL Graph), 22
- E-commerce (arsitektur SQL + NoSQL), 20

F

- Foreign Key, 21
- Fungsi agregasi, 25

G

- GROUP BY, 25

I

- Index (indeks database), 22
- INNER JOIN, 25
- INSERT, 24
- Instalasi DBMS, 27–29

J

- Join, konsep relasi, 21
- JSON, 18

K

- Key-Value (NoSQL), 22
- Kolom (Column), 21

L

- LEFT JOIN, 25

M

- MAX(), 25
- MIN(), 25
- MySQL, instalasi, 27–28

N

- Node (NoSQL Graph), 22
- NoSQL, 22

O

- OLAP, 16
- OLTP, 16
- ORDER BY, 25

P

- PostgreSQL, instalasi, 28–29
- Primary Key, 21

Q

- Query, 22
- Query Optimization, 22

R

- Relasi, 21
- Replication, 22
- RIGHT JOIN, 25
- Row (Baris), 21

S

- Schema, 21
- Security database, 22
- SELECT, 24
- Sharding, 22
- SUM(), 25

T

- Table (Tabel), 21
- Transaksi database, 22
- Trigger, 14

U

- UPDATE, 24
- User privileges, 29

V

- VARCHAR, tipe data, 11
- View, 14

W

- WHERE, 25

Buku Big Data dan NoSQL: Solusi Database di Era Digital menghadirkan pemahaman menyeluruh tentang bagaimana data dikelola, diproses, dan dioptimalkan dalam dunia digital yang semakin kompleks. Pembahasan dimulai dari konsep paling dasar seperti definisi data, peran DBMS, arsitektur tiga schema, hingga model data relasional yang menjadi fondasi utama dalam pengelolaan database modern. Di bagian awal, pembaca dipandu memahami bagaimana model relasional bekerja melalui konsep tabel, kunci, integritas data, dan operasi dasar, disertai contoh kasus sederhana yang dekat dengan kehidupan sehari-hari.

Pada bagian selanjutnya, buku ini memperdalam aspek desain database melalui pemodelan konseptual dan logis menggunakan Entity-Relationship Diagram, dilanjutkan dengan penerapan normalisasi untuk memastikan struktur data efisien dan bebas anomali. Pembaca juga diajak mengenal SQL secara komprehensif, mulai dari pembuatan tabel, manipulasi data, penggabungan tabel, hingga pembuatan view, procedure, function, dan trigger. Materi administratif seperti keamanan, manajemen akses, backup, recovery, dan optimasi kinerja database turut dibahas sebagai bekal pembaca memahami tugas seorang administrator database secara praktis.

Buku ini mengantarkan pembaca pada dunia NoSQL dan tantangan pengelolaan Big Data. Di dalamnya dibahas alasan munculnya NoSQL, karakteristik sistem BASE, serta ragam pendekatan seperti Document Store, Key-Value Store, Column-Family Store, dan Graph Database. Penjelasan mengenai kapan memilih SQL atau NoSQL memberikan perspektif penting tentang praktik polyglot persistence dalam pengembangan sistem modern. Melalui pembahasan teoritis yang kuat dan studi kasus aplikatif, buku ini menjadi rujukan lengkap bagi mahasiswa, praktisi IT, serta siapa pun yang ingin memahami solusi database yang relevan di era digital.

 www.penerbitwidina.com
 [@penerbitwidina](https://www.instagram.com/penerbitwidina)
 [penerbit widina](https://www.facebook.com/penerbitwidina)
 penerbitwidina@gmail.com
 widina store
 widina bookstore

Layanan Pembaca & Penerbitan Buku
 0815-7000-699



SCANME

Teknologi

ISBN 978-634-246-529-5



9

786342

465295