



# Bab 1: Pendahuluan

# Konsep-konsep Sistem Basis Data, Edisi 7.

©Silberschatz, Korth dan Sudarshan

Lihat [www.db-book.com](http://www.db-book.com) untuk mengetahui ketentuan tentang penggunaan ulang



# Contoh Aplikasi Basis Data

- Informasi Perusahaan
  - Penjualan: pelanggan, produk, pembelian
  - Akuntansi: pembayaran, penerimaan, aset
  - Sumber Daya Manusia: Informasi tentang karyawan, gaji, pajak penggajian.
- Manufaktur: manajemen produksi, inventaris, pesanan, rantai pasokan.
- Perbankan dan keuangan
  - informasi nasabah, rekening, pinjaman, dan transaksi perbankan.
  - Transaksi kartu kredit
  - Keuangan: penjualan dan pembelian instrumen keuangan (misalnya saham dan obligasi; menyimpan data pasar secara real-time
- Universitas: pendaftaran, nilai



# Contoh-contoh Aplikasi Basis Data (Cont.)

## (1 menit)

- Maskapai penerbangan: reservasi, jadwal
- Telekomunikasi: catatan panggilan, teks, dan penggunaan data, membuat tagihan bulanan, menjaga saldo pada kartu panggil prabayar
- Layanan berbasis web
  - Pengecer online: pelacakan pesanan, rekomendasi yang disesuaikan
  - Iklan online
- Basis data dokumen
- Sistem navigasi: Untuk mempertahankan lokasi berbagai tempat menarik beserta rute jalan, sistem kereta api, bus, dll.



# Tujuan Sistem Basis Data

Pada masa-masa awal, aplikasi basis data dibangun langsung di atas sistem file, yang mengarah ke:

- Redundansi dan ketidakkonsistenan data: data disimpan dalam berbagai format file yang mengakibatkan induplikasi informasi dalam file yang berbeda
- Kesulitan dalam mengakses data
  - Perlu menulis program baru untuk melaksanakan setiap tugas baru
- Isolasi data
  - Berbagai file dan format
- Masalah integritas
  - Batasan integritas (misalnya, saldo akun  $> 0$ ) menjadi "terkubur" dalam kode program daripada dinyatakan secara eksplisit
  - Sulit untuk menambahkan batasan baru atau mengubah batasan yang sudah ada



# Tujuan Sistem Basis Data (Lanjutan)

- Atomisitas pembaruan
  - Kegagalan dapat meninggalkan database dalam keadaan tidak konsisten dengan pembaruan parsial yang dilakukan
  - Contoh: Transfer dana dari satu rekening ke rekening lain harus dilakukan secara lengkap atau tidak sama sekali
- Akses bersamaan oleh beberapa pengguna
  - Akses bersamaan diperlukan untuk kinerja
  - Akses bersamaan yang tidak terkendali dapat menyebabkan ketidakkonsistenan
    - Contoh: Dua orang membaca saldo (katakanlah 100) dan memperbaruinya dengan menarik uang (masing-masing 50) pada waktu yang sama
- Masalah keamanan
  - Sulit untuk memberikan akses pengguna ke beberapa, tetapi tidak semua, data

**Sistem basis data menawarkan solusi untuk semua masalah di atas**



# Model Data

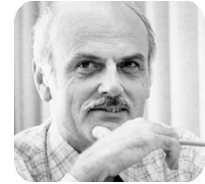
- Kumpulan alat bantu untuk menggambarkan
  - Data
  - Hubungan data
  - Semantik data
  - Batasan data
- Model relasional
- Model data Entity-Relationship (terutama untuk desain database)
- Model data berbasis objek (Berorientasi objek dan Relasi objek)
- Model data semi-terstruktur (XML)
- Model lama lainnya:
  - Model jaringan
  - Model hirarkis





# Model Relasional

- Semua data disimpan dalam berbagai tabel.
- Contoh data tabular dalam model relasional



**Ted Codd**  
Penghargaan  
Turing 1981

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Kolom

Baris

(a) The *instructor* table



# Contoh Basis Data Relasional

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

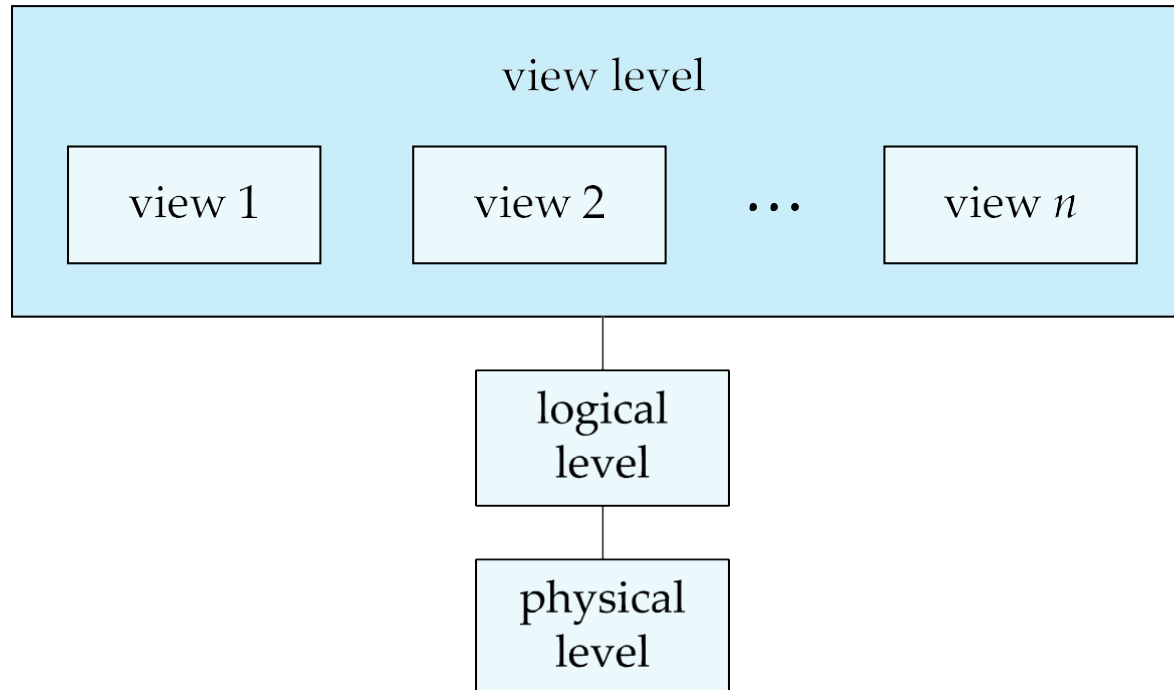
<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table



# Tampilan Data

Arsitektur untuk sistem basis data





# Contoh dan Skema

- Mirip dengan tipe dan variabel dalam bahasa pemrograman
- **Skema Logis** - struktur logis keseluruhan database
  - Contoh: Basis data terdiri dari informasi tentang sekumpulan pelanggan dan rekening di bank dan hubungan di antara mereka
    - Analog dengan informasi jenis variabel dalam sebuah program
- Skema **fisik**- keseluruhan struktur fisik database
- **Instance** - konten aktual dari database pada titik waktu tertentu
  - Analog dengan nilai sebuah variabel



# Independensi Data Fisik

- **Independensi Data Fisik** - kemampuan untuk memodifikasi skema fisik tanpa mengubah skema logis
  - Aplikasi bergantung pada skema logis
  - Secara umum, antarmuka antara berbagai level dan komponen harus didefinisikan dengan baik sehingga perubahan di beberapa bagian tidak terlalu mempengaruhi bagian lainnya.



# Bahasa Definisi Data (DDL)

- Notasi spesifikasi untuk mendefinisikan skema basis

data Contoh: **buat tabel** *instruktur* (  
                                  *ID*                  **char**(5),  
                                  *nama*             **varchar**(20),  
                                  *nama\_departemen*  
                                  **varchar**(20), *gaji*  
   **numeric**(8,2))

- Kompiler DDL menghasilkan satu set templat tabel yang disimpan dalam ***kamus data***
- Kamus data berisi metadata (yaitu data tentang data)
  - Skema basis data
  - Batasan integritas
    - Kunci utama (ID yang secara unik mengidentifikasi instruktur)
  - Otorisasi
    - Siapa yang dapat mengakses apa



# Bahasa Manipulasi Data (DML)

- Bahasa untuk mengakses dan memperbarui data yang diatur oleh model data yang sesuai
  - DML juga dikenal sebagai bahasa kueri
- Dua kelas bahasa
  - **Murni** - digunakan untuk membuktikan properti tentang daya komputasi dan untuk pengoptimalan
    - Aljabar Relasional
    - Kalkulus relasional tuple
    - Kalkulus relasional domain
  - **Komersial** - digunakan dalam sistem komersial
    - SQL adalah bahasa komersial yang paling banyak digunakan



# Bahasa Manipulasi Data (Lanjutan)

- Pada dasarnya ada dua jenis bahasa manipulasi data
  - **DML** prosedural - mengharuskan pengguna untuk menentukan data apa yang dibutuhkan dan bagaimana cara mendapatkan data tersebut.
  - **DML deklaratif** - mengharuskan pengguna untuk menentukan data apa yang dibutuhkan tanpa menentukan bagaimana mendapatkan data tersebut.
- DML deklaratif biasanya lebih mudah dipelajari dan digunakan daripada DML prosedural.
- DML deklaratif juga disebut sebagai DML non-prosedural
- Bagian dari DML yang melibatkan pengambilan informasi disebut bahasa **permintaan**.





# Bahasa Kueri SQL

- Bahasa kueri SQL adalah bahasa yang tidak prosedural. Sebuah kueri mengambil input beberapa tabel (mungkin hanya satu) dan selalu mengembalikan satu tabel.
- Contoh untuk menemukan semua instruktur di Comp. Sci. dept.  
**pilih** *nama*  
**dari** *instruktur*  
**where** *dept\_name* = 'Comp. Sci.'
- SQL **BUKAN** bahasa yang setara dengan mesin Turing
- Untuk dapat menghitung fungsi-fungsi kompleks, SQL biasanya disematkan dalam beberapa bahasa tingkat tinggi
- Program aplikasi umumnya mengakses database melalui salah satu dari
  - Ekstensi bahasa untuk memungkinkan SQL tertanam
  - Antarmuka program aplikasi (misalnya, ODBC/JDBC) yang memungkinkan kueri SQL dikirim ke database



# Akses Basis Data dari Program Aplikasi

- Bahasa kueri non-prosedural seperti SQL tidak sekuat mesin Turing universal.
- SQL tidak mendukung tindakan seperti input dari pengguna, output ke tampilan, atau komunikasi melalui jaringan.
- Komputasi dan tindakan tersebut harus ditulis dalam **bahasa host**, seperti C/C++, Java atau Python, dengan kueri SQL tertanam yang mengakses data dalam database.
- **Program aplikasi** - adalah program yang digunakan untuk berinteraksi dengan database dengan cara ini.



# Desain Basis Data

Proses merancang struktur umum basis data:

- Desain Logis - Memutuskan skema basis data. Desain basis data mengharuskan kita untuk menemukan kumpulan skema relasi yang "baik".
  - Keputusan bisnis - Atribut apa yang harus kita catat dalam database?
  - Keputusan Ilmu Komputer - Skema relasi apa yang harus kita miliki dan bagaimana atribut harus didistribusikan di antara berbagai skema relasi?
- Desain Fisik - Memutuskan tata letak fisik database



# Mesin Basis Data

- Sistem database dipartisi ke dalam modul-modul yang menangani masing-masing tanggung jawab dari keseluruhan sistem.
- Komponen fungsional dari sistem basis data dapat dibagi menjadi
  - Manajer penyimpanan,
  - Komponen pemroses kueri,
  - Komponen manajemen transaksi.



# Manajer Penyimpanan

- Modul program yang menyediakan antarmuka antara data tingkat rendah yang disimpan dalam database dan program aplikasi serta kueri yang dikirimkan ke sistem.
- Manajer penyimpanan bertanggung jawab atas tugas-tugas berikut ini:
  - Interaksi dengan manajer file OS
  - Menyimpan, mengambil, dan memperbarui data secara efisien\
- Komponen manajer penyimpanan meliputi:
  - Manajer otorisasi dan integritas
  - Manajer transaksi
  - Manajer file
  - Manajer penyangga



# Manajer Penyimpanan (1 menit)

- Manajer penyimpanan mengimplementasikan beberapa struktur data sebagai bagian dari implementasi sistem fisik:
  - File data - menyimpan basis data itu sendiri
  - Kamus data - menyimpan metadata tentang struktur basis data, khususnya skema basis data.
  - Indeks - dapat memberikan akses cepat ke item data. Indeks basis data menyediakan petunjuk ke item data yang memiliki nilai tertentu.



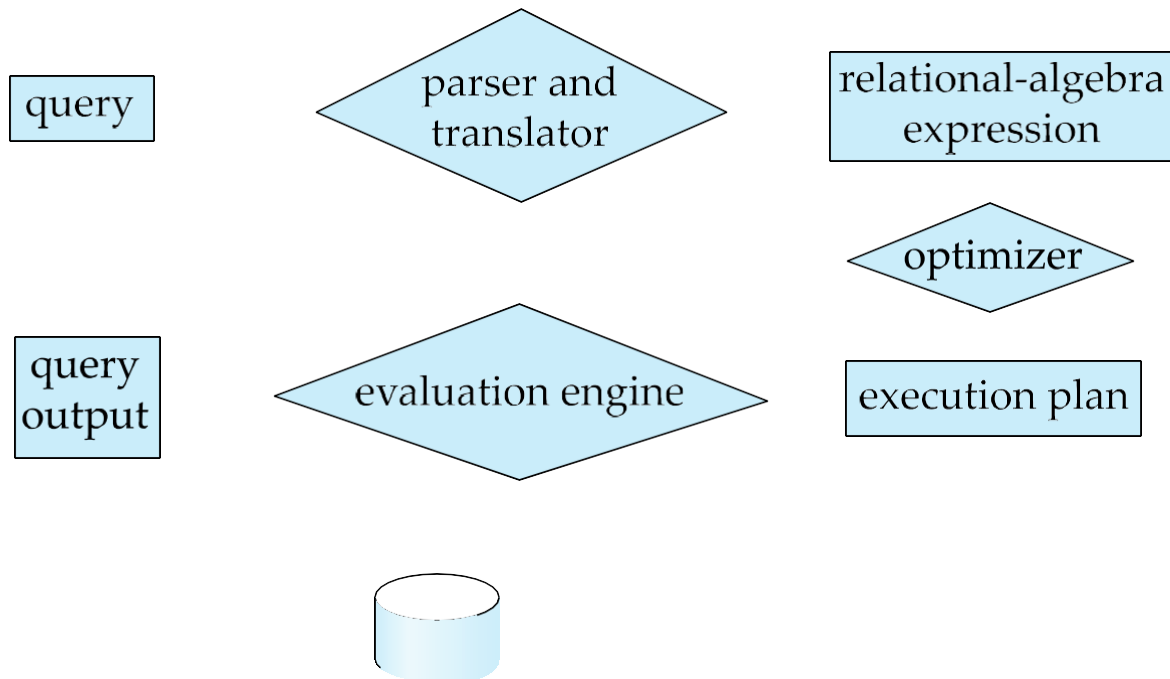
# Pemroses Kueri

- Komponen prosesor kueri meliputi:
  - Penerjemah DDL - menafsirkan pernyataan DDL dan mencatat definisi dalam kamus data.
  - Kompiler DML - menerjemahkan pernyataan DML dalam bahasa kueri ke dalam rencana evaluasi yang terdiri dari instruksi tingkat rendah yang dimengerti oleh mesin evaluasi kueri.
    - Kompiler DML melakukan optimasi kueri; yaitu, memilih rencana evaluasi dengan biaya terendah di antara berbagai alternatif.
  - Mesin evaluasi kueri - mengeksekusi instruksi tingkat rendah yang dihasilkan oleh kompiler DML.



# Pemrosesan Kueri

1. Penguraian dan terjemahan
2. Optimalisasi
3. Evaluasi







# Manajemen Transaksi

- **Transaksi** adalah kumpulan operasi yang menjalankan satu fungsi logis dalam aplikasi database
- **Komponen manajemen transaksi** memastikan bahwa database tetap dalam kondisi yang konsisten (benar) meskipun terjadi kegagalan sistem (misalnya, listrik mati dan sistem operasi macet) dan kegagalan transaksi.
- **Manajer kontrol** konkurensi mengontrol interaksi di antara transaksi yang bersamaan, untuk memastikan konsistensi database.



# Arsitektur Basis Data

- Basis data terpusat
  - Satu hingga beberapa inti, memori bersama
- Klien-server,
  - Satu mesin server menjalankan pekerjaan atas nama beberapa mesin klien.
- Basis data paralel
  - Banyak memori bersama inti
  - Disk bersama
  - Tidak ada yang dibagikan
- Basis data terdistribusi
  - Distribusi geografis
  - Heterogenitas skema/data



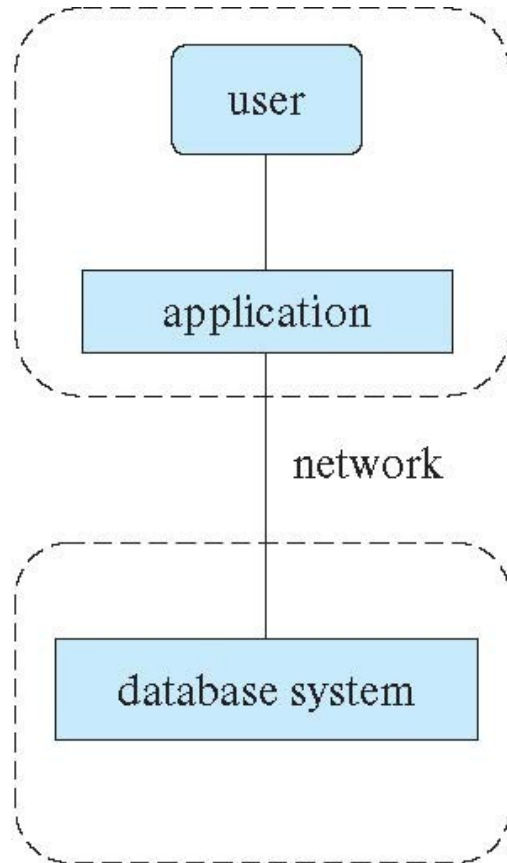
# Aplikasi Basis Data

Aplikasi basis data biasanya dipartisi menjadi dua atau tiga bagian

- **Arsitektur dua** tingkat - aplikasi berada di mesin klien, di mana ia memanggil fungsionalitas sistem basis data di mesin server
- **Arsitektur tiga tingkat** - mesin klien bertindak sebagai ujung depan dan tidak mengandung panggilan basis data langsung.
  - Ujung klien berkomunikasi dengan server aplikasi, biasanya melalui antarmuka formulir.
  - Server aplikasi pada gilirannya berkomunikasi dengan sistem basis data untuk mengakses data.



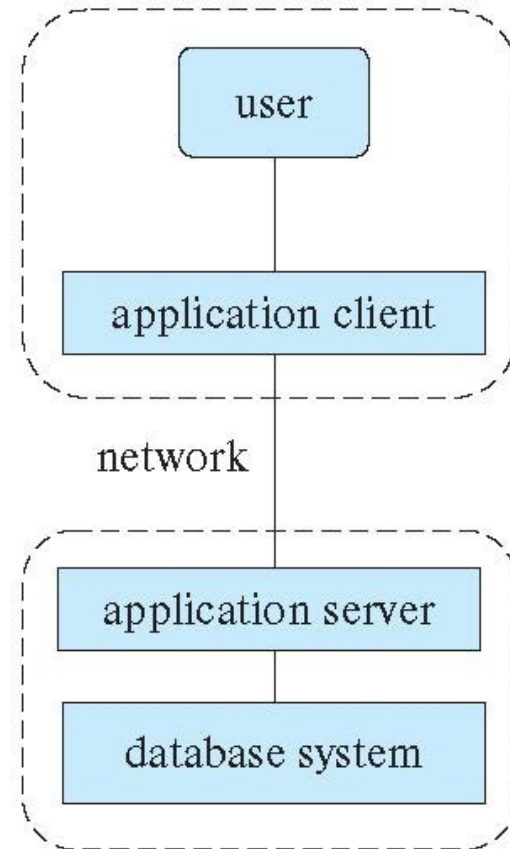
# Arsitektur dua tingkat dan tiga tingkat



(a) Two-tier architecture

client

server



(b) Three-tier architecture



# Pengguna Basis Data

Ada empat jenis pengguna sistem basis data yang berbeda

- Pengguna naif - pengguna yang tidak mahir yang berinteraksi dengan sistem dengan memanggil salah satu program aplikasi yang telah ditulis sebelumnya.
- Pemrogram aplikasi - adalah profesional komputer yang menulis program aplikasi.
- Pengguna yang canggih - berinteraksi dengan sistem tanpa menulis program
  - menggunakan bahasa kueri basis data atau dengan
  - menggunakan alat bantu seperti perangkat lunak analisis data.
- Pengguna khusus --menulis aplikasi basis data khusus yang tidak sesuai dengan kerangka kerja pemrosesan data tradisional. Misalnya, CAD, data grafis, audio, video.



# Administrator Basis Data

Seseorang yang memiliki kendali pusat atas sistem disebut **administrator basis data (DBA)**, yang memiliki fungsi:

- Definisi skema
- Struktur penyimpanan dan definisi metode akses
- Modifikasi skema dan organisasi fisik
- Pemberian otorisasi untuk akses data
- Perawatan rutin
- Mencadangkan basis data secara berkala
- Memastikan ruang disk kosong yang cukup tersedia untuk pengoperasian normal, dan meningkatkan ruang disk sesuai kebutuhan
- Memantau pekerjaan yang berjalan di database dan memastikan bahwa kinerja tidak menurun karena tugas yang sangat mahal yang dikirimkan oleh beberapa pengguna



# Sejarah Sistem Basis Data

- 1950-an dan awal 1960-an:
  - Pemrosesan data menggunakan pita magnetik untuk penyimpanan
    - Kaset hanya menyediakan akses berurutan
  - Kartu berlubang untuk masukan
- Akhir 1960-an dan 1970-an:
  - Hard disk memungkinkan akses langsung ke data
  - Model data jaringan dan hirarkis yang digunakan secara luas
  - Ted Codd mendefinisikan model data relasional
    - Akan memenangkan ACM Turing Award untuk karya ini
    - IBM Research memulai prototipe Sistem R
    - UC Berkeley (Michael Stonebraker) memulai prototipe Ingres
    - Oracle merilis database relasional komersial pertama
  - Pemrosesan transaksi berkinerja tinggi (untuk zamannya)



# Sejarah Sistem Basis Data (Lanjutan)

- 1980s:
  - Prototipe relasional penelitian berkembang menjadi sistem komersial
    - SQL menjadi standar industri
  - Sistem basis data paralel dan terdistribusi
    - Wisconsin, IBM, Teradata
  - Sistem basis data berorientasi objek
- 1990s:
  - Aplikasi pendukung keputusan dan penambangan data yang besar
  - Gudang data multi-terabyte yang besar
  - Munculnya perdagangan Web





# Sejarah Sistem Basis Data (Lanjutan)

- 2000s
  - Sistem penyimpanan data besar
    - Google BigTable, Yahoo PNuts, Amazon,
    - Sistem "NoSQL".
  - Analisis data besar: melampaui SQL
    - Mengurangi peta dan teman-teman
- 2010s
  - SQL dimuat ulang
    - Ujung depan SQL ke sistem Map Reduce
    - Sistem basis data paralel secara besar-besaran
    - Basis data memori utama multi-inti



**Akhir dari Bab 1**



# AKHIR BAB 1