

## **BAB I** **PENDAHULUAN : SISTEM BASIS DATA**

### **Apa yang disebut dengan Sistem Manajemen Basis Data (Database Management System) ?**

- Himpunan data yang terintegrasi
- Model yang menggambarkan dunia nyata
  - Entiti (contoh : mahasiswa, mata kuliah)
  - Relasi (contoh : Keanu mengambil mata kuliah dengan kode CS564)
- Sistem Manajemen Basis Data adalah paket perangkat lunak yang didesain untuk melakukan penyimpanan dan pengaturan basis data.

### **Mengapa menggunakan DBMS ?**

- Independensi (kemandirian) data dan akses yang efisien
- Mereduksi waktu pengembangan aplikasi
- Integritas dan keamanan data
- Administrasi keseragaman data
- Akses bersamaan dan perbaikan dari terjadinya crashes (tabrakan dari proses serentak).

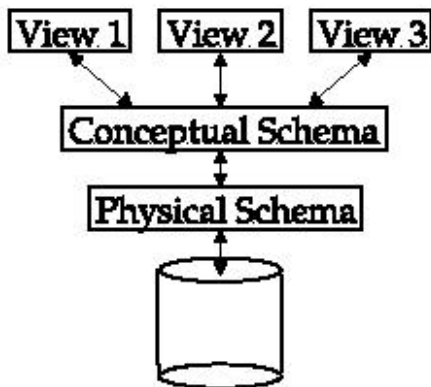
### **Mengapa mempelajari Basis Data ?**

- Perpindahan dari komputasi ke informasi
- Himpunan elemen data semakin banyak dan beragam
  - Perpustakaan digital, video interaktif, Human genome project, EOS project
  - Kebutuhan untuk memperluas DBMS
- DBMS mencakup bidang ilmu lain
  - Sistem operasi, bahasa pemrograman, teori komputasi, AI, logika, multimedia

### **Model Data**

- Model data adalah himpunan konsep untuk menggambarkan suatu data
- Skema adalah deskripsi umum dari himpunan data, dengan menggunakan model data
- Model relasional adalah model data yang paling banyak digunakan saat ini.
  - Konsep utama : relasi, pada dasarnya adalah sebuah table dengan baris dan kolom.  
Tiap relasi memiliki skema, yang menggambarkan kolom atau fields

### Level Abstraksi



- Memiliki beberapa tinjauan (views), skema konseptual tunggal (logical) dan skema fisik.
    - Menggambarkan bagaimana cara user melihat data
    - Skema konseptual mendefinisikan struktur logika
    - Skema fisik menggambarkan file dan indeks yang digunakan
- ⇒ Skema didefinisikan menggunakan DDL (Data Definition Language), data dimodifikasi dengan menggunakan DML (Data Management Language).

### Contoh : Basis Data Universitas

- Skema Konseptual :
  - Students (sid: string, name : string, login : string, age : integer, gpa : real)
  - Courses(cid : string, cname : string, dcredits : integer)
  - Enrolled(sid : string, cid : string, grade : string)
- Skema Fisik
  - Relasi yang disimpan sebagai file yang belum terurut
  - Indeks pada kolom pertama file students
- Skema Eksternal
  - Course\_info(cid : string, enrollment : integer)

### **Kebebasan Data**

- Aplikasi disekat dari bagaimana data disimpan dan distrukturkan
- Kebebasan data secara logika : Perlindungan dari perubahan struktur logika suatu data
- Kebebasan data secara fisik : Perlindungan dari perubahan struktur fisik suatu data
  - ⇒ Kebebasan data adalah salah satu keuntungan utama dari penggunaan DBMS

### **Kontrol Proses Serentak**

- Eksekusi bersamaan dari beberapa user program, merupakan ukuran performansi yang utama dari suatu DBMS.
  - Dikarenakan akses ke disk cukup sering, dan relatif lebih lambat, maka perlu untuk menjaga waktu CPU pada beberapa user program yang bekerja secara serentak.
- Interleaving pada user program yang berbeda dapat menyebabkan ketidakkonsistenan, misal : field check dihapus pada saat account balance dihitung.
- DBMS memastikan bahwa masalah seperti diatas tidak akan terjadi seperti halnya sistem yang menggunakan single-user.

### **Transaksi : Eksekusi dari Program Basis Data**

- Konsep kuncinya adalah transaksi, merupakan urutan atomic dari aksi basis data (baca/tulis).
- Tiap transaksi dijalankan sampai selesai, dan basis data harus tetap dijaga agar tetap dalam keadaan yang konsisten.
  - User dapat menentukan beberapa batasan integritas (integrity constraint) yang sederhana pada data, dan kemudian DBMS akan mengusahakan batasan ini.
  - Selain dari itu, DBMS tidak benar-benar mengerti semantic dari data (contoh : tidak tahu cara menghitung bunga bank, atau tidak tahu cara menetapkan nilai jual barang yang ada di gudang ... misalnya).
  - Sehingga untuk memastikan transaksi berlangsung secara konsisten, seluruhnya berada pada tanggung jawab user.

### **Penjadwalan dari Transaksi yang berlangsung Serentak**

- DBMS memastikan bahwa eksekusi dari  $\{T_1, \dots, T_n\}$  adalah sama dengan eksekusi serial  $T_1' \dots T_n'$ 
  - Sebelum membaca/menulis obyek, transaksi meminta lock pada obyek dan menunggu sampai DBMS memberikan lock. Semua lock akan dilepas (direlease) pada akhir dari transaksi

- Ide : Jika aksi dari T1 (missal menulis X) menyebabkan kejadian Tj (missal membaca X), salah satu dari mereka , missal akan T1, akan mendapatkan lock pada X pertama kali dan Tj akan menunggu sampai aksi T1 diselesaikan.
- Apa yang terjadi jika TJ telah memiliki lock kemudian setelah itu T1 meminta (request) lock Y ? (Deadlock akan terjadi , maka T1 atau Tj harus di-abort atau di-restart).

### **Memastikan sifat Atomik**

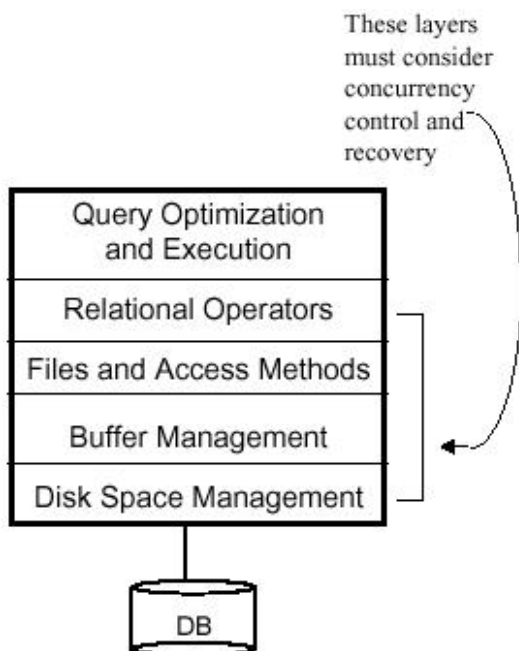
- DBMS memastikan sifat atomic (semuanya atau tidak ada property).
- Ide : Simpan log (history) semua aksi yang dijalankan oleh DBMS pada saat menjalankan himpunan aksi X.
  - Sebelum perubahan dibuat terhadap basis data, maka log entry harus berkorespondensi dengan lokasi yang aman (system operasi seringkali tidak mendukung secara penuh hal seperti ini).
  - Setelah terjadi crash (tabrakan), akibatnya sebagian transaksi yang telah dieksekusi tidak lagi dikerjakan dengan menggunakan log.

### **Basis Data membuat segalanya menjadi lebih mudah**

- Pemakai akhir dan vendor DBMS
- Programmer aplikasi basis data
  - Misal : webmaster yang canggih
- Administrator Basis Data (Database Administrator)
  - Desain skema Logikal/Fisikal
  - Menangani pengamanan dan kewenangan
  - Ketersediaan data, perbaikan crash
  - Penyesuaian oleh Basis Data

*Yang penting adalah mengerti bagaimana cara DBMS mengerjakan semua itu*

### **Struktur DBMS**



- Biasanya DBMS memiliki arsitektur layer
- Gambar diatas tidak menunjukkan kontrol proses serentak (concurrency control) dan komponen perbaikan (recovery).
- Gambar tersebut adalah salah satu dari beberapa kemungkinan arsitektur : tiap system memiliki variasinya sendiri.

**Ringkasan :**

- DBMS digunakan untuk pemeliharaan himpunan query yang sangat banyak.
- Keuntungan dari penggunaan DBMS meliputi perbaikan dari system yang crash, akses serentak, pengembangan aplikasi secara cepat, keamanan dan integritas data.
- Level abstraksi memberikan kebebasan data.
- Umumnya DBMS memiliki arsitektur per layer
- DBMS adalah salah satu bidang yang paling banyak diminati dalam ilmu komputer.

