



# Basis Data

## PENDAHULUAN

Alif Finandhita, S.Kom

# Definisi

- **Basis data :**  
Adalah sekumpulan data persistence yang saling terkait, menggambarkan suatu organisasi(enterprise).
- **Sistem Basis data (DBS):**  
Suatu sistem yang mengelola data dan menyediakan data tersebut apabila dibutuhkan.

# Komponen Dasar Sistem Basis Data

## ■ Data :

- Data disimpan secara terintegrasi (integrated).
- Data dapat dipakai secara bersama-sama (shared).
- Integrated yaitu database merupakan kumpulan dari berbagai macam file dari aplikasi – aplikasi yang berbeda yang disusun dengan cara menghilangkan bagian – bagian yang rangkap (redundant).
- Shared yaitu masing – masing bagian dari database dapat diakses oleh pemakai dalam waktu yang bersamaan untuk aplikasi berbeda.

# Komponen Dasar Sistem Basis Data (2)

## ■ Hardware :

Terdiri dari semua perangkat keras yang digunakan untuk pengelolaan sistem database.

- Peralatan untuk penyimpanan, hard disk, magnetic tape, dll.
- Peralatan input dan output.
- Peralatan komunikasi data.

## ■ Software :

Sebagai perantara (interface) antara pemakai dengan data fisik pada database.

- Database Management System (DBMS)
- Program – program aplikasi & prosedur - prosedur

# Komponen Dasar Sistem Basis Data (3)

## ■ User :

Terbagi menjadi 3 klasifikasi :

- Database Administrator (DBA), orang/team yang bertugas mengelola sistem database secara keseluruhan.
- Programmer, orang/team yang membuat program aplikasi yang dapat mengakses database dengan menggunakan bahasa pemrograman.
- End User, orang yang mengakses database melalui terminal / client dengan menggunakan query language atau program aplikasi yang dibuat oleh programmer.

# Komponen Dasar Sistem Basis Data (4)

## ■ Database Administrator :

- Adalah orang yang memiliki kontrol utama terhadap keseluruhan sistem basis data (mencakup data & program).
- Fungsi :
  - Pendefinisian skema.
  - Pendefinisian struktur penyimpanan & metode akses.
  - Modifikasi skema & organisasi fisik.
  - Pemberian otorisasi bagi pengaksesan data.
  - Mendefinisikan bagian basis data yang mana yang dapat diakses oleh seorang pemakai, termasuk operasi-operasi yang dapat dilakukan.
  - Spesifikasi batasan integritas.

# Komponen Dasar Sistem Basis Data (5)

- **Data Administrator :**
  - Adalah orang yang bertanggung jawab dalam kontrol terhadap data.
  - Fungsi :
    - Menentukan data apa yang harus disimpan.
    - Menentukan aturan dalam penanganan data.

# Sistem Basis Data vs Kertas

- **Kepadatan :**  
Tidak diperlukan jumlah kertas yang sangat banyak untuk menyimpan data dan cukup ringkas.
- **Kecepatan :**  
Data dapat diambil dan dimanipulasi lebih cepat dibandingkan pencarian secara manual dari kertas.
- **Kemudahan :**  
Mengurangi pekerjaan yang menjemukan jika harus berurusan dengan berlembar-lembar kertas.
- **Kekinian :**  
Data yang disimpan adalah data yang akurat sesuai perkembangan (up to date) dan dapat disediakan pada saat yang dibutuhkan.



# Data pada Sistem Basis Data dan Hubungannya

Ada 3 Jenis data pada sistem basis data :

- Data operasional dari suatu enterprise, berupa data yang disimpan di dalam database.
- Data masukan (input data), data dari luar sistem yang dimasukkan melalui peralatan input (keyboard) yang dapat merubah data operasional.
- Data keluaran (output data), berupa laporan melalui peralatan output (screen, printer) sebagai hasil dari dalam system yang mengakses data operasional

# Keuntungan Basis Data

- Terkontrolnya kerangkapan (redundant) dan inkonsistensi data.
- Terpeliharanya keselarasan data.
- Data dapat dipakai secara bersama-sama.
- Memudahkan penerapan standarisasi.
- Memudahkan penerapan batasan-batasan pengamanan.
- Terpeliharanya integritas data.
- Terpeliharanya keseimbangan atas perbedaan kebutuhan data dari setiap aplikasi.
- Program / data independent

# Daftar Istilah yang Digunakan dalam Basis Data

- **Enterprise**, suatu bentuk organisasi.  
Contoh : Universitas -> Data Mahasiswa  
Rumah Sakit -> Data Pasien
- **Entitas**, suatu objek yang dapat dibedakan dari objek lainnya.  
Contoh :
  - Bidang Administrasi Mahasiswa -> Entitas Mahasiswa, pembayaran, buku
  - Bidang Kesehatan -> Entitas Pasien, Dokter, Obat.

# Daftar Istilah yang Digunakan dalam Basis Data (2)

- **Attribute/field**, setiap entitas mempunyai atribut atau suatu sebutan untuk mewakili suatu entitas.

Contoh :

- Entitas Mahasiswa -> field nim, nama, alamat
- Entitas Nasabah -> field no\_rek, nama, alamat

- **Data value (nilai atau isi data)**, data aktual atau informasi yang disimpan pada tiap data elemen atau atribut. Isi dari atribut disebut nilai data.

Contoh :

- Atribut Nama Karyawan -> Adi Saputra, Rian Satria
- Atribut Kota -> Bandung, Bogor, Jakarta.

# Daftar Istilah yang Digunakan dalam Basis Data (3)

- **Record/tuple**, kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entity secara lengkap.  
Contoh :
  - Satu record mewakili satu data/informasi -> nim, nama\_mhs, alamat
- **File**, kumpulan record-record sejenis yang mempunyai panjang elemen sama, attribute yang sama, namun berbeda-beda data valuenya

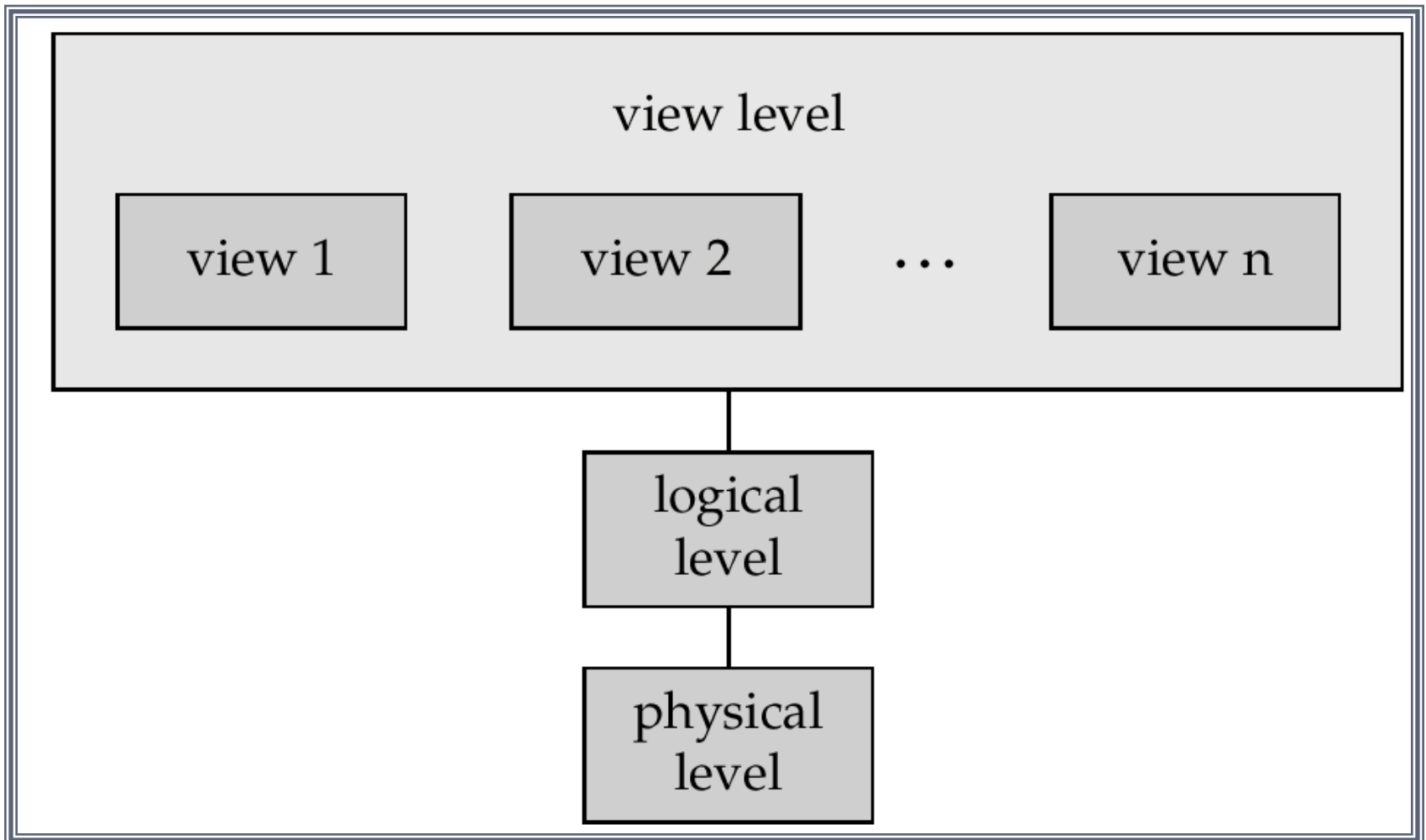
# Daftar Istilah yang Digunakan dalam Basis Data (4)

- **Kunci elemen data**, tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas
- **Database Management System (DBMS)**, kumpulan file yang saling berkaitan bersama dengan program untuk pengelolaannya.

# Abstraksi Data

- **Abstraksi data**, merupakan tingkatan/level dalam hal bagaimana melihat data dalam sebuah sistem basis data.
- Ada 3 level abstraksi data :
  - Level Fisik (Physical Level) – Internal Level
  - Level Logik/Konseptual (Conceptual Level)
  - Level Penampakan (View Level)

# Abstraksi Data (2)





# Abstraksi Data (3)

## ■ **Physical Level – Internal Level**

Merupakan level terendah, yang menunjukkan bagaimana sesungguhnya suatu data/record disimpan. Pada level ini pemakai melihat data sebagai gabungan dari struktur dan datanya itu sendiri.

- Bagaimana data disimpan
- Deskripsi detail struktur data

## ■ **Conceptual/Logical Level**

Menjelaskan suatu data yang disimpan di dalam database, dan hubungannya antara satu data dengan data lainnya.

- Data apa yang tersimpan (Representasi abstrak).
- Relasi antar data.
- Digunakan oleh DBA.

# Abstraksi Data (4)

- **View Level – External Level**

Merupakan level tertinggi, hanya menunjukkan sebagian dari basis data sesuai dengan kebutuhan user, bagi user yang menggunakan terasa sebagai suatu kesatuan yang kompak.

- Mendeskripsikan sebagian basis data.
- Digunakan oleh pengguna.

# Bahasa Basis Data

- Secara garis besar bahasa basis data (database languages) dibagi ke dalam dua bentuk :
  - **Data Definition Language (DDL)**
  - **Data Manipulation Language (DML)**

# Bahasa Basis Data (2)

## Data Definition Language (DDL)

- Dengan bahasa inilah kita dapat membuat tabel baru, membuat index, mengubah tabel, menentukan struktur penyimpanan tabel, dll.
- Hasil dari kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut Kamus Data (Data Dictionary)
- Kamus Data merupakan suatu metadata (superdata) yaitu data yang mendeskripsikan data sesungguhnya. Kamus Data ini selalu diakses dalam suatu operasi basis data sebelum file data yang sesungguhnya diakses.

# Bahasa Basis Data (3)

## Data Manipulation Language (DML)

- Berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. Manipulasi data dapat berupa :
  - Penyisipan/penambahan data baru (insert)
  - Penghapusan data (delete)
  - Pengubahan data (update)
- DML merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data

# Bahasa Basis Data (4)

## Jenis Data Manipulation Language (DML)

- **Prosedural**, yang mensyaratkan agar pemakai menentukan, data apa yang diinginkan serta bagaimana cara mendapatkannya.
- **Non-Prosedural**, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

# Model Data

- **Model Data**, merupakan sekumpulan tools yang digunakan untuk menjelaskan data, hubungan antar data (relationships), skema data, dan batasan data.
- Model Data terdiri dari :
  - Object Based Logical Model
  - Record Based Logical Model
  - Physical Data Model

# Model Data (2)

## ■ Object based Logical Model

- Untuk mendeskripsikan data level konseptual dan view.
- Penstrukturan data dilakukan berbasis objek.
- Contoh : entity-Relationship model, Object-oriented Model.

## ■ Record based Logical Model

- Untuk mendeskripsikan data level Koseptual & view.
- Mendeskrisikan struktur logik keseluruhan basis data.
- Menstrukturkan basis data dalam sejumlah fixed format records.

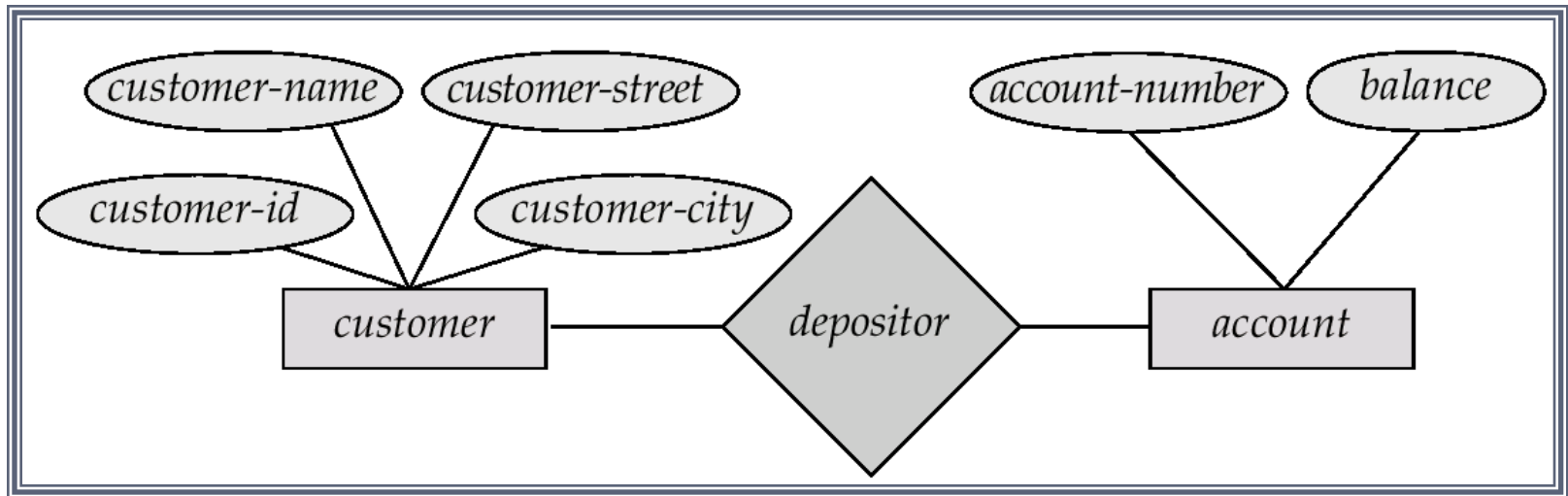
## ■ Physical data Model

- Untuk mendeskripsikan data pada level terendah.



# Model Data (3)

- Model E – R



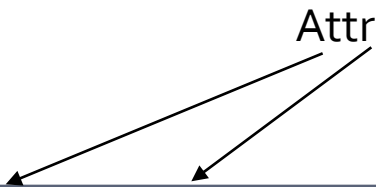
Contoh Diagram Entity Relationship

# Model Data (4)

- Contoh model E – R dari kehidupan sehari – hari :
  - Entitas (Objek) : customers, accounts
  - Hubungan antar entitas :
    - Account A-101 dimiliki oleh seorang customer yang bernama Johnson
    - Relationship mengatur depositor yang terasosiasikan ke customer dengan accountnya

# Model Data (5)

## ■ Model Relasional



<i>Customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>	<i>account-number</i>
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

Contoh data tabular dalam bentuk Model Relasional

# Model Data (6)

<i>customer-id</i>	<i>customer-name</i>	<i>customer-street</i>	<i>customer-city</i>
192-83-7465	Johnson	12 Alma St.	Palo Alto
019-28-3746	Smith	4 North St.	Rye
677-89-9011	Hayes	3 Main St.	Harrison
182-73-6091	Turner	123 Putnam Ave.	Stamford
321-12-3123	Jones	100 Main St.	Harrison
336-66-9999	Lindsay	175 Park Ave.	Pittsfield
019-28-3746	Smith	72 North St.	Rye

(a) The *customer* table

<i>account-number</i>	<i>balance</i>
A-101	500
A-215	700
A-102	400
A-305	350
A-201	900
A-217	750
A-222	700

(b) The *account* table

<i>customer-id</i>	<i>account-number</i>
192-83-7465	A-101
192-83-7465	A-201
019-28-3746	A-215
677-89-9011	A-102
182-73-6091	A-305
321-12-3123	A-217
336-66-9999	A-222
019-28-3746	A-201

(c) The *depositor* table

Contoh Database Relasional

# Instance dan Skema

## ■ Instance :

- Adalah sekumpulan data yang tersimpan di dalam basis data pada suatu waktu tertentu.
- Memiliki frekuensi perubahan yang tinggi.

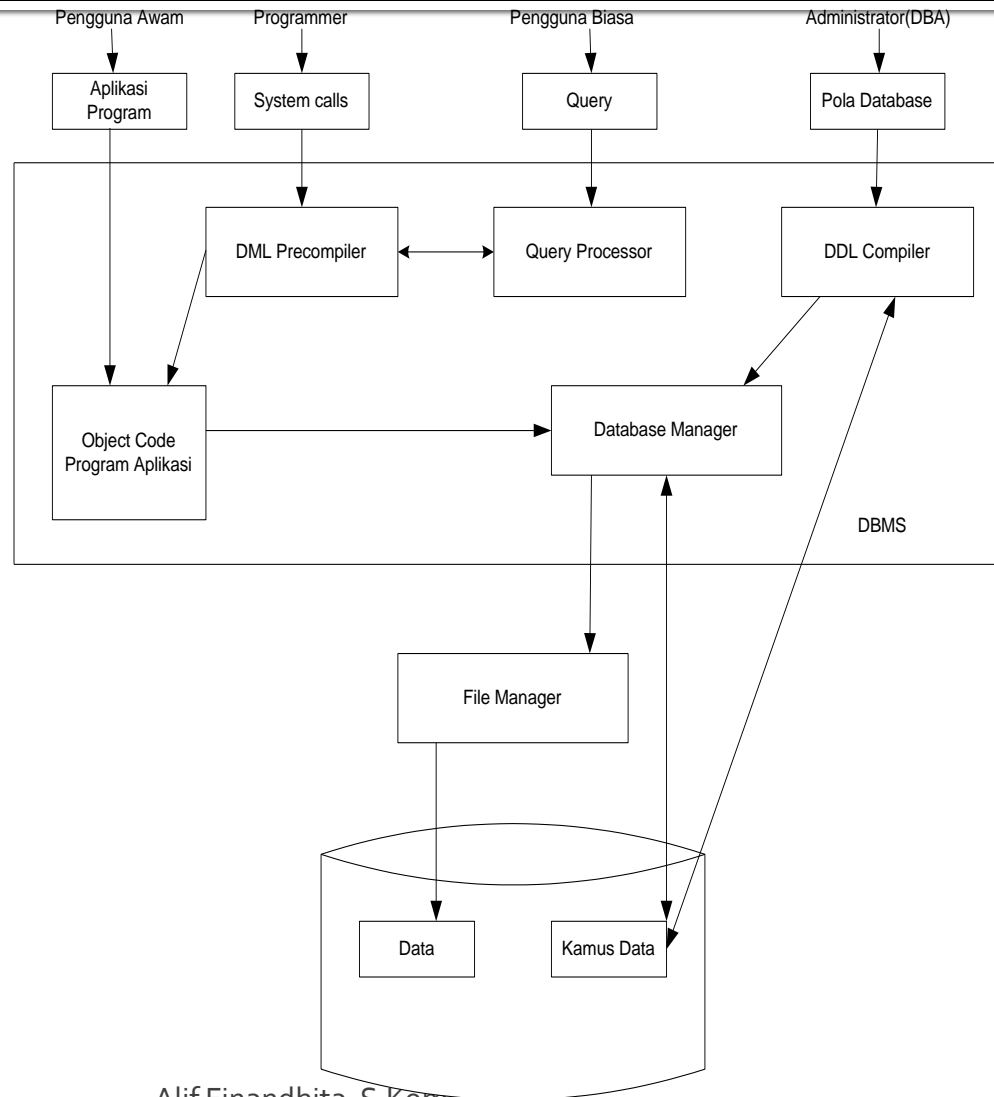
## ■ Skema :

- Adalah desain keseluruhan basis data.
- Memiliki frekuensi perubahan yang sangat rendah.
- Umumnya, sebuah basis data akan memiliki satu skema fisik.
- Idependensi data : kemampuan untuk mendefinisikan skema pada suatu level tanpa mempengaruhi skema di level berikutnya.

# Struktur Sistem Basis Data

- **File Manager**
  - Mengelola space & struktur data.
- **Database Manager**
  - Menyediakan antarmuka dengan data fisik.
- **Query Processor**
  - Menterjemahkan query ke instruksi yang dimengerti Database Manager.
- **DML Precompiler**
  - Mengkonversi perintah DML menjadi calls normal.
- **DDL Compiler**
  - Mengkonversi perintah DDL menjadi metadata.

# Struktur Sistem Basis Data (2)



# Sistem Manajemen Basis Data (DBMS)

- **DBMS** adalah perangkat lunak yang menangani semua mekanisme pengaksesan database. Mempunyai fasilitas membuat, mengakses, memanipulasi dan memelihara basis data.
- Tujuan DBMS adalah menyediakan lingkungan yang mudah dan aman untuk penggunaan dan perawatan database.



# Sistem Manajemen Basis Data - DBMS

## ■ Fungsi DBMS

- **Data Definition**, DBMS harus dapat mengolah pendefinisian data.
- **Data Manipulation**, DBMS harus dapat menangani permintaan dari pemakai untuk mengakses data.
- **Data Security & Integrity**, DBMS harus dapat memeriksa security dan integritas data yang didefinisikan oleh DBA.

# Sistem Manajemen Basis Data - DBMS (2)

## ■ Fungsi DBMS

- **Data Recovery & Concurrency**, DBMS harus dapat menangani kegagalan – kegagalan pengaksesan database yang dapat disebabkan oleh kesalahan sistem, kerusakan disk, dsb.
- **Data Dictionary**, DBMS harus dapat menyediakan data dictionary (kamus data).
- **Performance**, DBMS harus dapat menangani unjuk kerja dari semua fungsi seefisien mungkin.