

# **BASIS DATA (BS203)**

## **Pengenalan Basis Data**

[k\\_doroedi@yahoo.com](mailto:k_doroedi@yahoo.com)

fb: NDoro Edi

# Tujuan Pembelajaran

- Dapat menjelaskan karakteristik basis data
- Dapat menjelaskan keuntungan basis data dibandingkan dengan sistem berkas/file

# Outline

- Beberapa Definisi
- Data vs. informasi vs. metadata
- Definisi & Sejarah Basis Data
- Basis Data vs. Sistem Berkas
- *Database Management System* (DBMS)
- Komponen utama DBMS dan penggunaannya (*user*)
- Abstraksi data
- Model basis data
- Skema vs. instansi
- Bahasa basis data: SQL (DDL & DML)
- Struktur sistem keseluruhan

# Beberapa Definisi

## Data

- Rekaman mengenai fenomena/ fakta yang ada atau yang terjadi.
- Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol)

## Informasi

- Data yang telah diolah dan sehingga dapat digunakan untuk dasar pengambilan keputusan.
- Hasil analisis dan sintesis terhadap data.
- Data yang telah diorganisasikan ke dalam bentuk yang sesuai dengan kebutuhan seseorang (grafik, dll)

## Basis Data

- kumpulan data yang terorganisir berdasarkan suatu struktur hubungan

## Metadata

- Data mengenai data
- Data yang mendeskripsikan karakteristik/properti dari data lain.

## DBMS

- paket perangkat lunak yang digunakan untuk pembuatan dan pemeliharaan basis data

## Sistem Basis Data

- Perangkat lunak DBMS bersama basis data, kadang-kadang disertai aplikasi

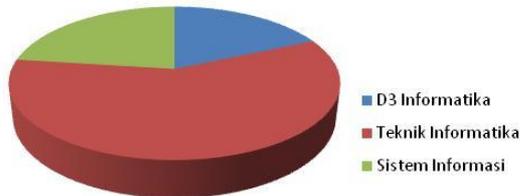
# Contoh data, informasi, metadata

D3 Teknologi Informasi	100
Teknik Informatika	320
Sistem Informasi	125

**Data**

## Informasi

Jumlah Mhs yg mengambil Basis Data  
Semester Ganjil 2007-2008

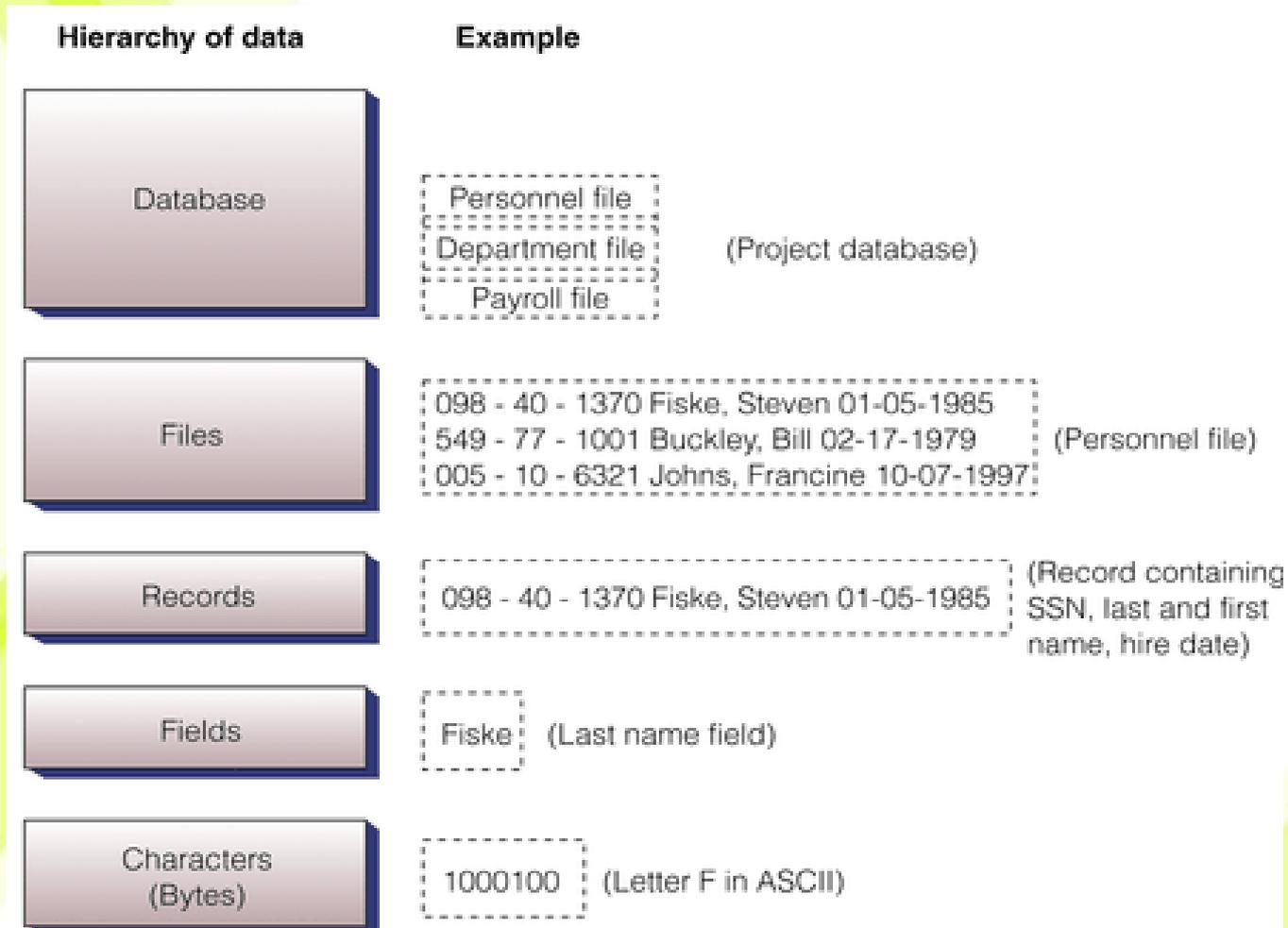


Jurusan	Jumlah Mhs yg mengambil MK Basis Data
D3 Teknologi Informasi	100
Teknik Informatika	320
Sistem Informasi	125

Nama	Tipe	Panjang	Deskripsi
Jurusan	Alphanumeric	30	Nama jurusan
Jumlah Mhs	Integer	4	Jumlah orang yang mengambil MK Basis Data

**Metadata**

# Hirarki Data



# Definisi Basis Data

- Kumpulan data yang saling berhubungan yang merefleksikan fakta-fakta yang terdapat di suatu organisasi.
- Kumpulan informasi/berkas data yang terorganisasi ke dalam tatacara yang khusus

# Sejarah Basis Data

Pada saat awal penerapan sistem komputer, sekelompok rekaman disimpan pada sejumlah berkas secara terpisah. Sistem ini disebut sebagai **sistem pemrosesan berkas** (*File Processing System*).

# Evolusi Teknologi Basis Data

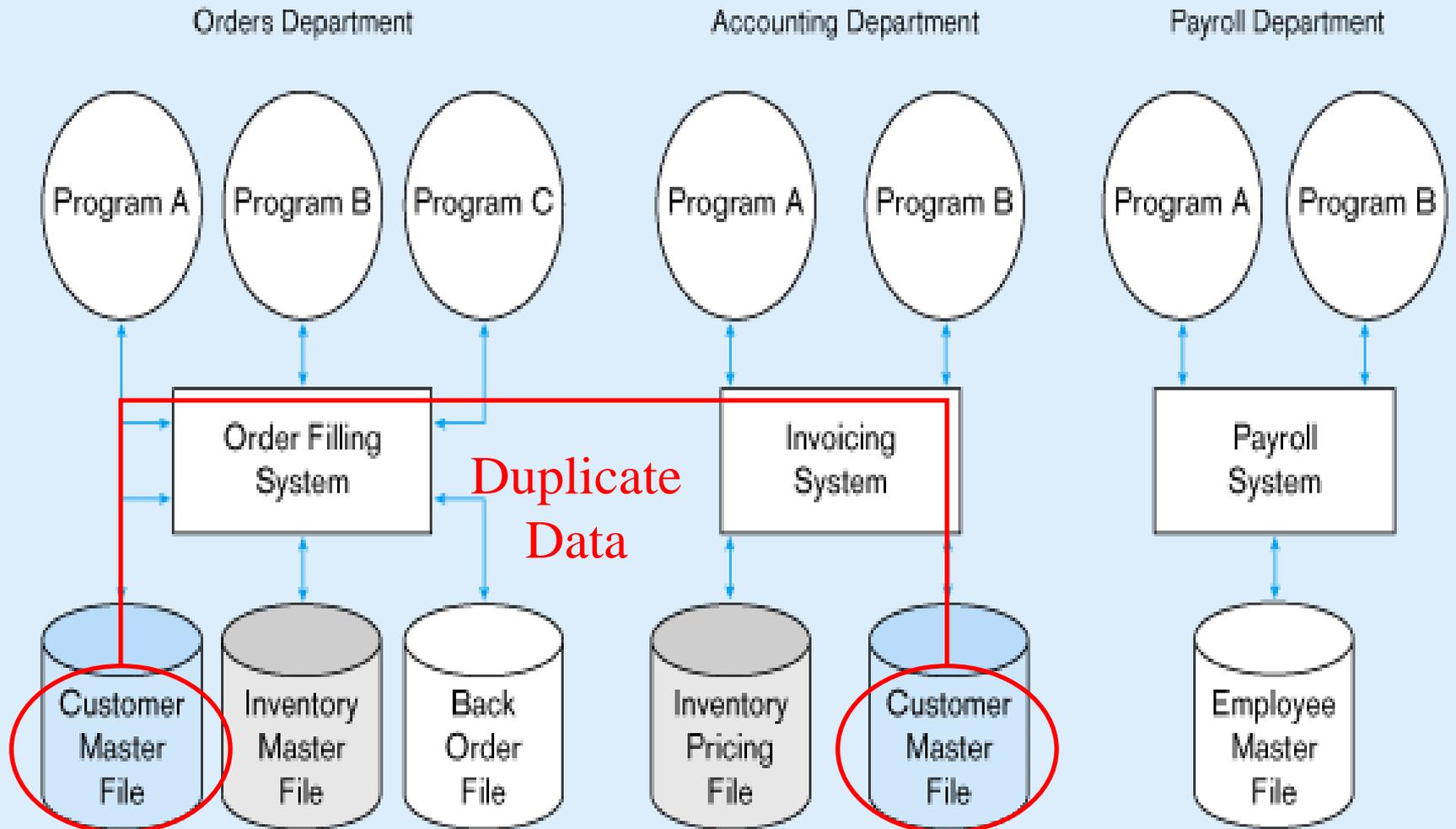
<b>Masa</b>	<b>Perkembangan Basis Data</b>
1960-an	<ul style="list-style-type: none"><li>• Sistem pemrosesan berkas</li><li>• DBMS</li><li>• Layanan informasi secara <i>online</i> berbasis manajemen teks</li></ul>
1970-an	<ul style="list-style-type: none"><li>• Penerapan sistem pakar pada sistem pendukung pengambilan keputusan</li><li>• Basis data berorientasi objek</li></ul>
1980-an	<ul style="list-style-type: none"><li>• Sistem hypertext, yang memungkinkan untuk melihat basis data secara acak menurut suatu kata kunci (sebagaimana yang diterapkan pada Internet)</li></ul>
1990-an	<ul style="list-style-type: none"><li>• Sistem basis data cerdas</li><li>• Sistem basis data multimedia cerdas</li></ul>

# Sistem Pemrosesan Berkas

## **Contoh:**

- Program komputer yang digunakan di perusahaan retail.
- Mereka memerlukan paling sedikit tiga aplikasi:
  - Order Filling
  - Invoicing, and
  - Payroll.

# Sistem Pemrosesan Berkas



# Kekurangan Sistem Pemrosesan Berkas

- Kemubaziran data atau duplikasi data. Setiap program aplikasi menggunakan data tersendiri.
- Keterbatasan berbagi data.
- Ketidakkonsistenan data (integritas berkurang).
- Kekurangluwesannya dalam hal pengembangan atau perubahan.

# Komponen Utama Sistem Basis Data

1. **Perangkat keras** (prosesor, memori, dan harddisk) untuk memproses dan menyimpan data
2. **Sistem Operasi** (program yang mengaktifkan/memfungsikan sistem komputer, mengendalikan seluruh sumber daya dalam komputer, dan melakukan operasi-operasi dasar dalam komputer)

# Komponen Utama Sistem Basis Data

## 3. Data

- **Terpadu** (integrated) berarti bahwa berkas-berkas data saling terkait. Dengan demikian, kemubaziran dapat dihindari.
- **Berbagi data** (shared) berarti bahwa data dapat dipakai oleh sejumlah pengguna dalam waktu bersamaan (multiuser).

## 4. Perangkat lunak (DBMS)

## 5. Pengguna

# SOLUSI

Sistem pemrosesan basis data dimaksudkan untuk mengatasi kelemahan-kelemahan yang ada pada sistem pemrosesan berkas. Sistem ini dikenal dengan sebutan **DBMS** (*Database Management System*).

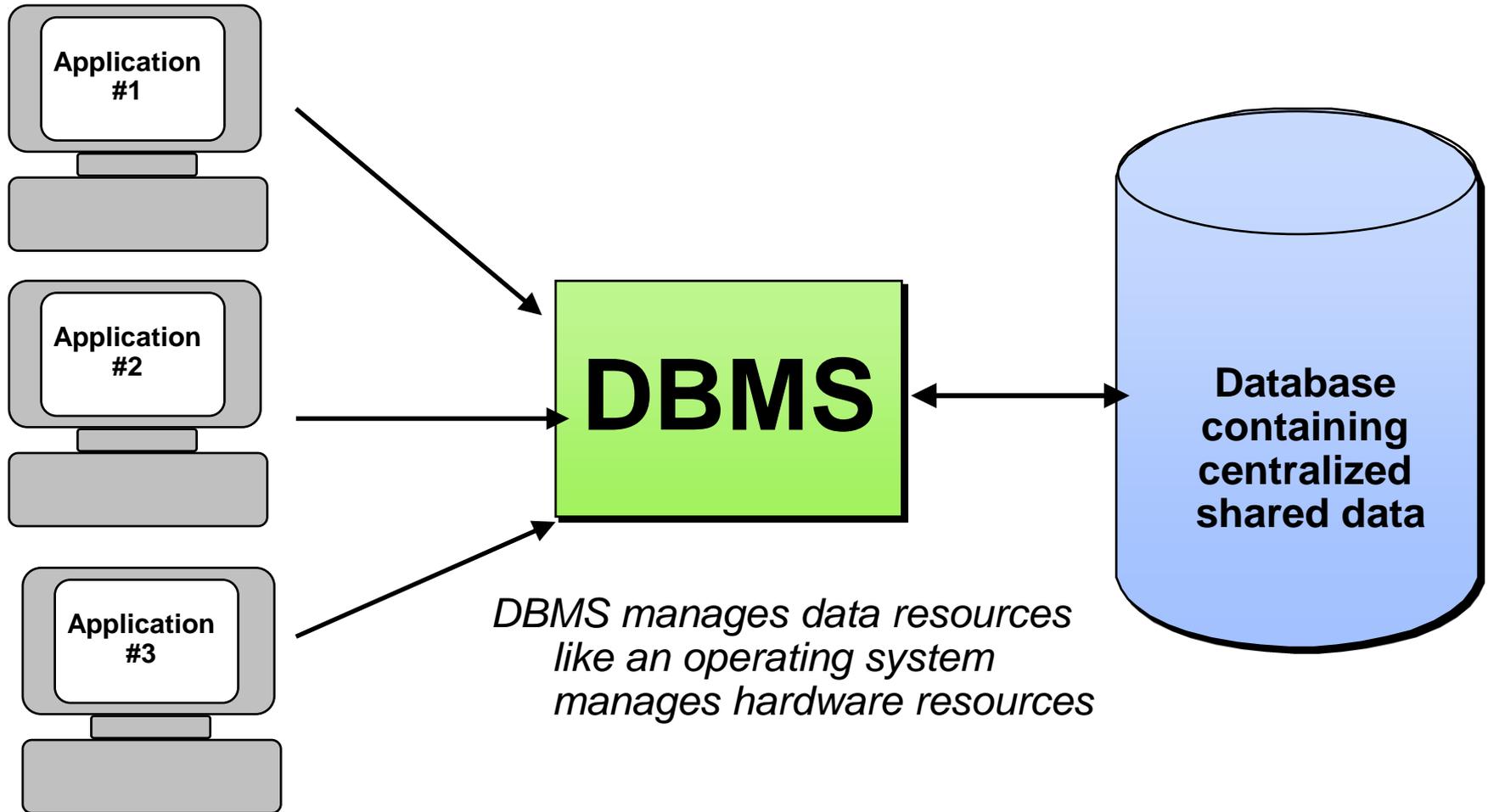
# DBMS: *Database Management System*

- **Definisi:** Suatu kumpulan data yang saling berkaitan dan seperangkat program untuk mengakses data tersebut.
- **Tujuan:** DBMS menyediakan suatu cara untuk menyimpan, mengubah, dan mengambil informasi pada basis data dengan cara yang nyaman dan efisien.
- **Contoh** DBMS yang ada di pasaran: Oracle, Microsoft SQL Server, IBM DB2, MySQL, PostgreSQL, etc.

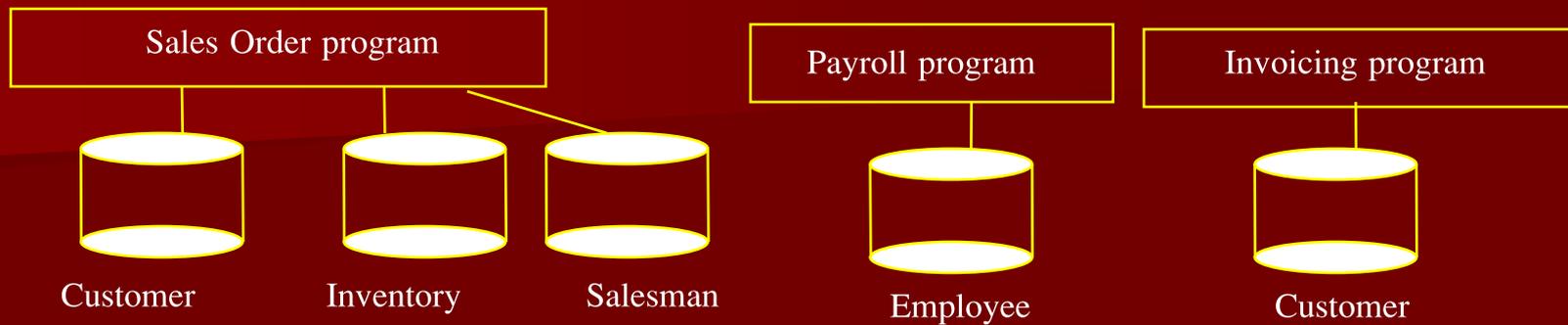
# Database Management System (DBMS)

Secara umum, DBMS diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi, dan memperoleh data/informasi dengan praktis dan efisien.

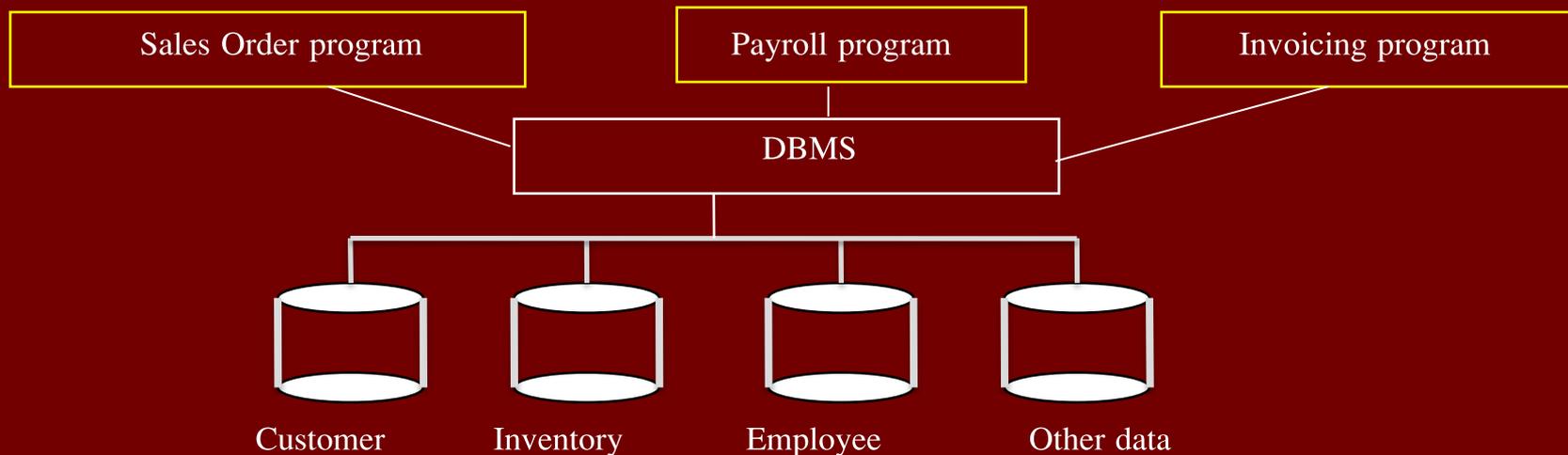
# Database Management System



# Sistem Pemrosesan Berkas Vs. DBMS



## Sistem Pemrosesan Berkas



**Basis Data sebagai koleksi file-file yang saling berhubungan**

# Keuntungan DBMS

- Kemubaziran data berkurang
- Integritas (kekonsistenan) data
- Independensi data
  - Perubahan struktur berkas tidak mempengaruhi program, dan juga sebaliknya. Independensi data juga bisa berarti bahwa data bersifat tidak tergantung pada data lain.
- Berbagi data

# Komponen DBMS

## 1. Storage Manager

- Menyediakan **antarmuka** antara program aplikasi dengan data yang disimpan di basis data.
- **Menterjemahkan** perintah-perintah DBMS menjadi perintah-perintah file-system di sistem operasi.

## 2. Query Processor, terdiri atas:

- Interpreter bahasa pendefinisian data (DDL)
- Compiler bahasa manipulasi data (DML)
- Mesin pengevaluasi *query*

# Pengguna (User)

- **Programmer Aplikasi**
  - Berinteraksi melalui *Data Manipulation Language* (DML), yang disertakan dalam program yang ditulis dalam bahasa pemrograman induk (seperti C++, C, Pascal, dll)
- **User Mahir (Casual User)**
  - Berinteraksi dengan sistem tanpa menulis modul program. Mereka menyatakan query (untuk akses data) dengan bahasa query yang telah disediakan oleh suatu DBMS.
- **User Umum (End User/Naïve User)**
  - Berinteraksi melalui pemanggilan satu program aplikasi permanen (*executable program*) yang telah disediakan/ditulis sebelumnya.

# Pengguna (User)

- **User Khusus (Specialized User)**
  - Menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus, seperti untuk aplikasi AI, Sistem pakar, dll. yang bisa saja mengakses basis data dengan/tanpa DBMS yang bersangkutan.
- **Database Administrator (DBA)**
  - Bertanggung jawab terhadap pengelolaan basis data. Tugas DBA antara lain: mendefinisikan basis data, menentukan isi basis data, menentukan sekuritas basis data, memantau kinerja sistem, merencanakan *backup* dan *recovery*, dan mengikuti perkembangan produk.

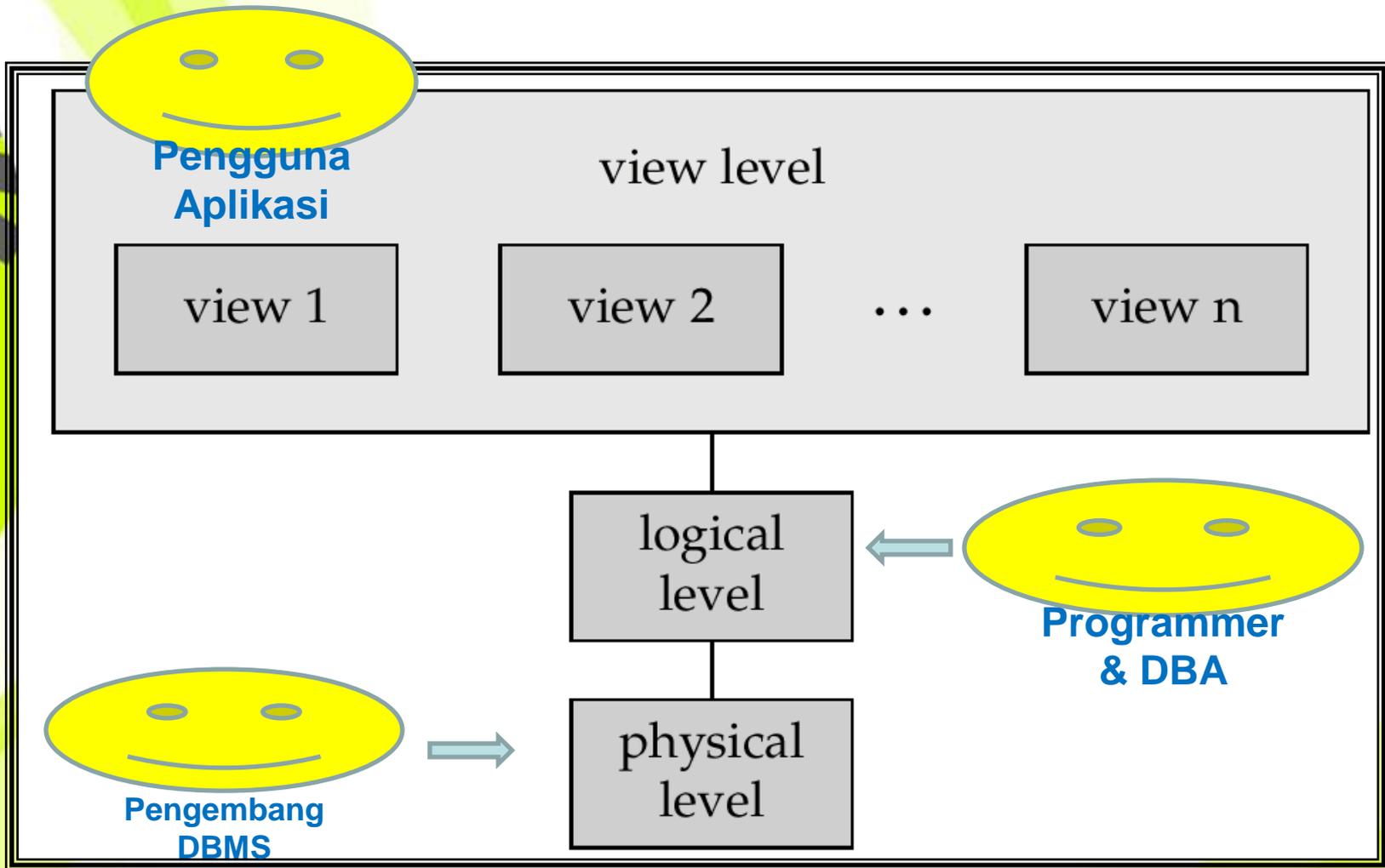
# DBA: *Database Administrator*

- DBA adalah pengelola DBMS (orang yang bertanggung jawab mengelola DBMS).
- Tugasnya meliputi:
  - Mendefinisikan skema basis data
  - Mendefinisikan struktur penyimpanan
  - Mendefinisikan metode pengaksesan data
  - Memodifikasi basis data
  - Memberikan hak akses kepada para pengguna
  - Melakukan pemeliharaan (backup, upgrade, monitoring)
- Tersedia sertifikasi internasional dari Oracle, Microsoft, & IBM.

# Abstraksi Data

- Tujuan DBMS adalah untuk menyediakan kepada *user* suatu pandangan 'abstrak' dari data yang dikelolanya.
- Artinya, sistem basis data menyembunyikan teknik-teknik detail penyimpanan dan pemrosesan data tersebut.
- Untuk itu, sistem tersebut akan menyembunyikan detail tentang bagaimana data disimpan dan dipelihara. Karena itu, seringkali data yang terlihat oleh pemakai sebenarnya berbeda dengan yang tersimpan secara fisik.
- Abstraksi data merupakan tingkatan/level dalam bagaimana melihat data dalam sebuah sistem basis data.

# Level Abstraksi data



# Abstraksi Data

- **Level Fisik:** menggambarkan **bagaimana** data sesungguhnya disimpan di **disk** (urusannya **pengembang DBMS**)
- **Level Logik:** menggambarkan **data apa** yang disimpan pada basis data dan **bagaimana hubungan antar data** tersebut. (urusannya **programmer** dan **admin basis data**)
- **Level View:** menggambarkan **hanya sebagian** dari basis data yang ada. (urusannya **pengguna aplikasi**, misal teller di bank)

# Abstraksi Data

- **Level Fisik (Physical Level)**

Lapis terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan. Pada lapis inilah struktur data dijabarkan secara rinci

- **Level Logik/Konseptual (Conceptual Level)**

Lapis ini menjabarkan data apa saja yang sesungguhnya disimpan pada basis data, dan juga menjabarkan hubungan-hubungan antardata. Level ini biasa dipakai oleh DBA.

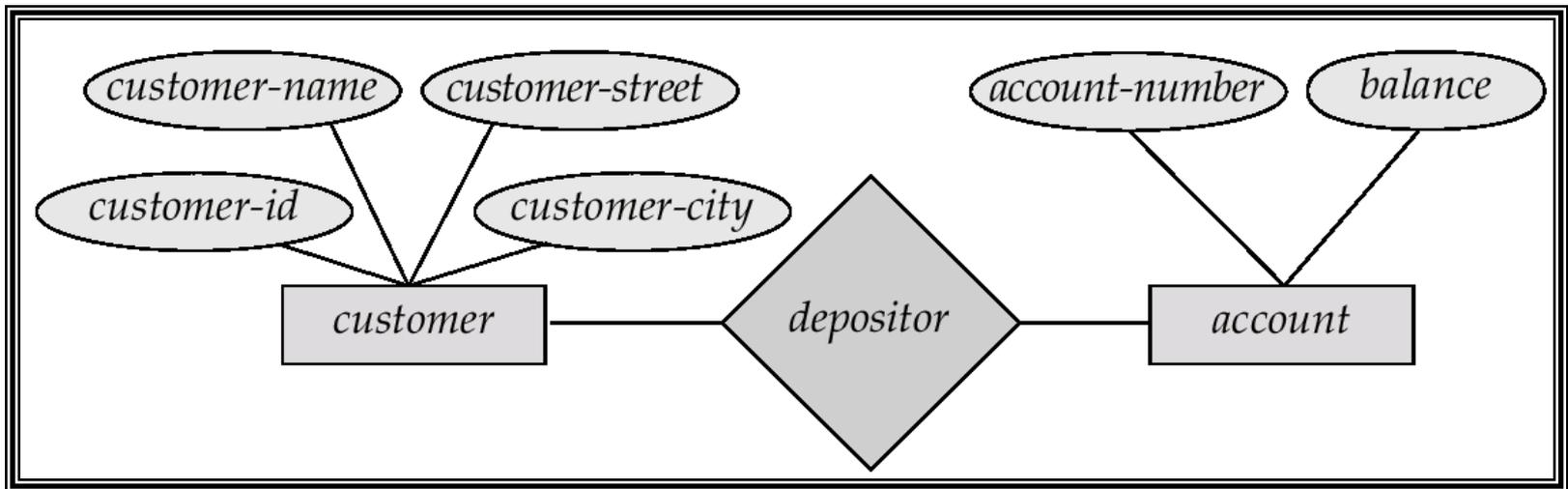
- **Level Penampakan (View Level)**

Lapis tertinggi pada abstraksi data. Pada lapis ini pengguna hanya mengenal struktur data yang sederhana, yang berorientasi pada kebutuhan pengguna. Data yang dikenal oleh masing-masing pengguna bisa berbeda-beda dan barangkali hanya mencakup sebagian dari basis data.

# Model Basis Data

- Model basis data menyatakan hubungan antar rekaman (record) yang tersimpan dalam basis data.
- Istilah lain: **struktur data logis**.
- Ada beberapa model basis data, yaitu:
  - o **Entity-Relationship Model**
  - o **Relational Model**
  - o Object-oriented Model
  - o Semi-structured Model
  - o Older models: network model and hierarchical model

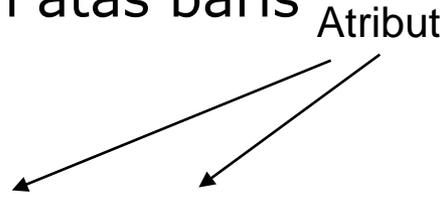
# Entity-Relationship Model



Contoh skema di dalam Entity-Relationship model

# Relational Model

Model ini menggunakan sekumpulan tabel berdimensi dua (yang disebut relasi atau tabel), dengan masing-masing relasi tersusun atas baris (record/ tuple) dan kolom (atribut).



<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 Model Relasional

<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

# Skema Vs. Instansi

- **Skema** adalah deskripsi lengkap tentang struktur medan, rekaman, dan hubungan data pada basis data.
  - Skema dapat dibayangkan sebagai suatu kerangka yang tidak tergantung nilai. Berbagai nilai yang valid dapat diterapkan pada kerangka.
  - Skema bersifat relatif tetap, sementara nilai-nilainya dapat berubah dari suatu saat ke saat yang lain.
  - Tugas utama skema adalah menjabarkan struktur basis data, seperti nama basis data, nama seluruh berkas pada basis data, nama rekaman dan medan, dll.
- **Instansi** adalah nilai yang terdapat pada basis data pada suatu saat.

# Bahasa Basis Data: SQL

- SQL = Structured Query Language

SQL terdiri atas DCL, DDL, dan DML

C=Control, D=Definition, M=Manipulation

- Sistem Basis Data menyediakan 'bahasa pendefinisi data' (DDL) untuk membuat skema basis data dan 'bahasa manipulasi data' (DML) untuk mengolah data (tambah, hapus, ubah).
- DDL = Data Definition Language
- DML = Data Manipulation Language

# DDL & DML

- **Contoh DDL:** membuat tabel 'rekening'

```
CREATE TABLE rekening  
    ( nomor CHAR(10),  
      saldo INTEGER)
```

- **Contoh DML:** menampilkan alamat 'nasabah' yang bernama 'ani':

```
SELECT alamat  
FROM nasabah  
WHERE name = 'ani'
```

# Data-Definition Language (DDL)

- Menspesifikasikan struktur/skema basis data.
- Bahasa untuk membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur penyimpanan tabel, dsb.
- Hasil dari kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut **Kamus Data** (Data Dictionary).
- Kamus data adalah suatu metadata dan akan selalu diakses dalam suatu operasi basis data sebelum suatu file data yang sesungguhnya diakses.

# Data-Manipulation Language (DML)

- Berguna untuk menambah, menghapus, dan mengubah data di suatu basis data.
- DML merupakan bahasa yang bertujuan memudahkan pemakai untuk mengakses data sebagaimana direpresentasikan oleh model data.
- Ada 2 jenis DML:
  - **Prosedural**, yang mensyaratkan agar pemakai menentukan data apa yang diinginkan serta bagaimana cara mendapatkannya.
  - **Nonprosedural**, yang membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

# Struktur Sistem Keseluruhan

DBMS terbagi atas modul-modul yang masing-masing memiliki tanggung jawab dalam membentuk struktur sistem keseluruhan:

- **File Manager**, yang mengelola alokasi ruang dalam disk dan struktur data yang dipakai untuk merepresentasikan informasi yang tersimpan dalam disk. Lebih difokuskan pada efisiensi dan efektivitas penyimpanan.
- **Database Manager**, yang menyediakan interface antara data *low-level* dengan program aplikasi dan query yang diberikan ke sistem.

# Struktur Sistem Keseluruhan

- **Query Processor**, yang menerjemahkan perintah-perintah dalam *query language* ke perintah *low-level* yang dapat dimengerti oleh *database manager* dan mentransformasikan permintaan *user* ke bentuk yang lebih efisien sehingga *query* menjadi lebih efektif.
- **DML Precompiler**, yang mengkonversi perintah DML yang ditambahkan dalam sebuah program aplikasi ke pemanggilan prosedur normal dalam bahasa induk.
- **DDL Compiler**, yang mengkonversi perintah-perintah DDL ke dalam sekumpulan tabel yang mengandung *metadata*.

# Struktur Sistem Keseluruhan

