

MODEL DATA & SKEMA

Dosen: Moh. Ali Albar, S.T., M.Eng.



LESSONS

1. Model Basis Data

2. Model Data & Schema

3. Arsitektur Sistem Database & Kebebasan Data

4. Bahasa & Interface Database



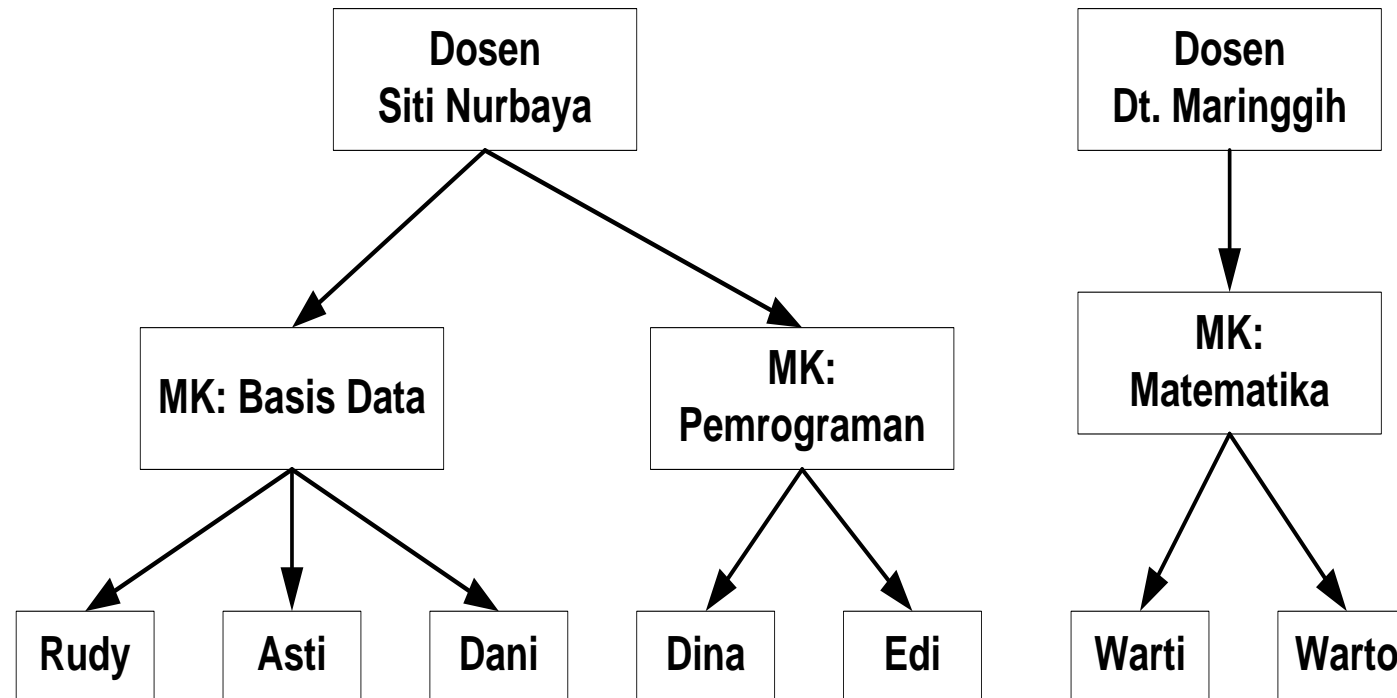
MODEL BASIS DATA

- Model Hierarkis
- Model Jaringan
- Model Relasional

MODEL HIERARKIS

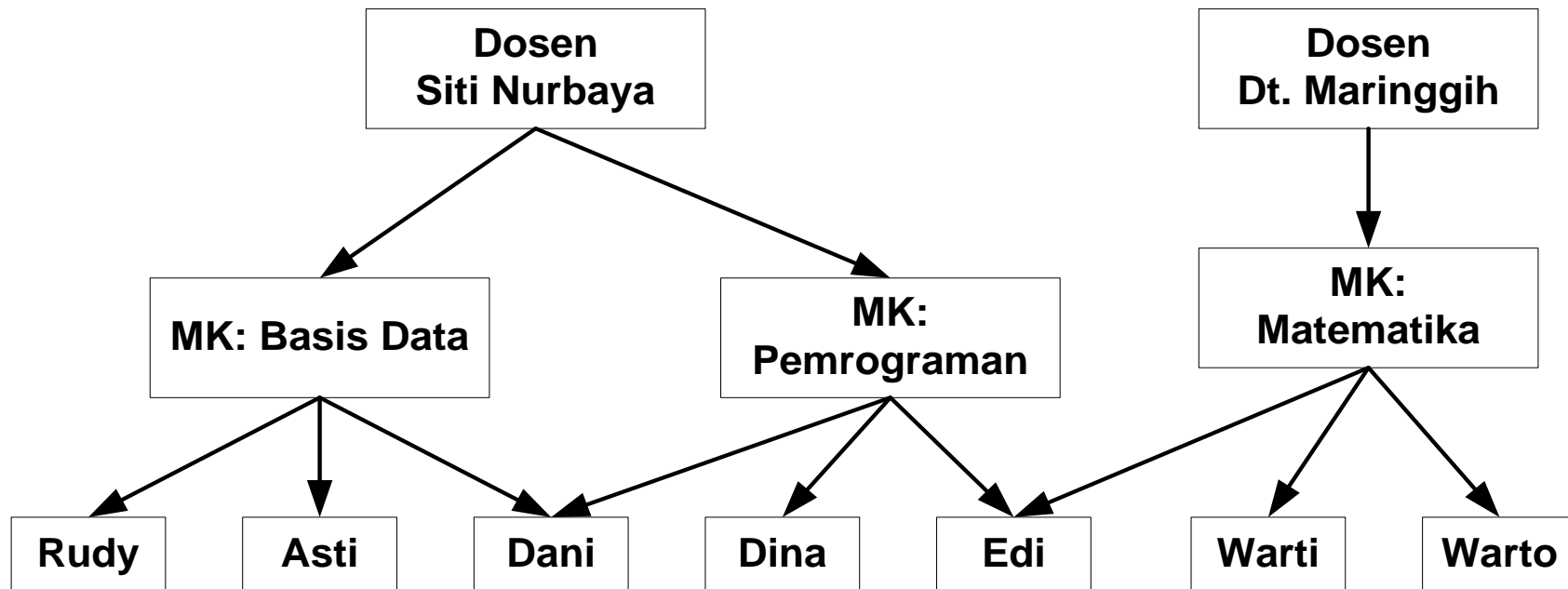
- ✓ Biasa disebut dengan model **pohon** (*tree model*) → menggunakan pola hubungan orangtua-anak (*parent-child*). **1 anak** hanya memiliki **1 orang tua**
- Setiap simpul menyatakan sekumpulan medan (*field*).
- Suatu simpul (*node*) yang terhubung ke simpul pada level di bawahnya → **orangtua** (*parent*).
- Simpul yang dibawahinya orangtua → **anak** (*child*).
- Simpul yang tak memiliki anak → **daun** (*leaf*).
- Hubungan orangtua dan anak → **cabang** (*branch*).

SKEMA MODEL HIERARKIS



MODEL JARINGAN

- Satu simpul anak bisa memiliki **lebih dari satu** simpul orangtua, orangtua → pemilik (owner) dan anak → anggota.





MODEL RELASIONAL

- Merupakan model yang paling sederhana sehingga mudah digunakan dan dipahami, serta paling populer saat ini
- Model ini berisikan sekumpulan tabel berhubungan (memiliki relasi), dimana relasi dirancang untuk menghilangkan kemubaziran data.

SKEMA MODEL RELASIONAL

NIM	<u>Nama_mhs</u>
K001	Rita
K002	Rina
K003	Rini
K004	Rani
K005	Rika

Tb_mhs

<u>Kd_konsen</u>	<u>Konsentrasi</u>
S1	Sistem Basis Data
S2	Sistem Berbasis Web
S3	Sistem Cerdas

Tb_konsen

NIM	<u>Kd_konsen</u>	<u>NIP</u>
K001	S1	D010
K002	S1	D010
K003	S2	D020
K004	S3	D030
K005	S2	D020

Tb_minat_mhs

<u>NIP</u>	<u>Nama_dosen</u>
D010	Dani
D020	Doni
D030	Dion

Tb_dosen

PERBEDAAN SKEMA DAN SUB-SKEMA

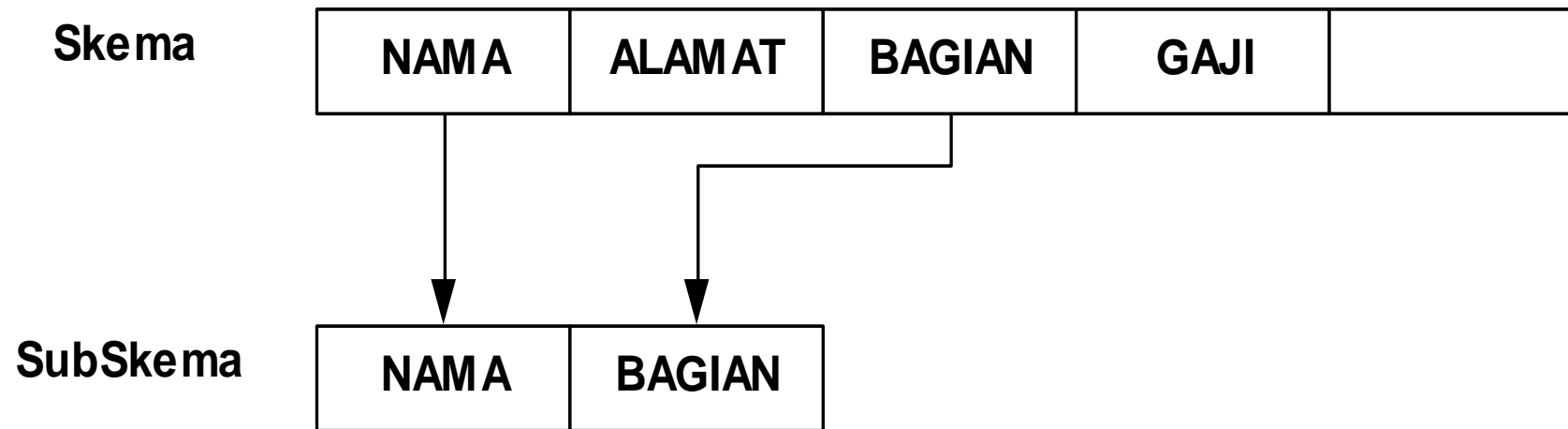
- Skema

Deskripsi hubungan logik antar data dalam basis data secara **lengkap**, termasuk di dalamnya nama dan deskripsi dari **semua** atribut dan record.

- Sub-Skema

Deskripsi **terpisah** dari dari atribut, record, dan batasan nilai yang akan digunakan oleh sebuah program aplikasi

PERBEDAAN SKEMA VS SUB-SKEMA





LESSONS

1. Model Basis Data
- 2. Model Data & Schema**
3. Arsitektur Sistem Database & Kebebasan Data
4. Bahasa & Interface Database

INTRODUCTION

- Arsitektur database telah mengalami evolusi dari sistem monolitik hingga modular berupa client/server.
- Dasar dari arsitektur DBMS client/server adalah sistem secara fungsional didistribusikan antara dua jenis modul, yaitu **modul client** dan **modul server**.
- **Modul Client**: secara khusus dirancang agar proses dapat berjalan pada sisi komputer client (workstation/PC).
- Proses yang berjalan pada sisi **client** adalah berbagai **program aplikasi** serta **interface user** dalam mengakses database.
- Oleh karena itu, pada modul **client** ini menangani **interaksi user** dan menyediakan **interface yang user-friendly** agar dapat digunakan dengan mudah oleh user/client (menu, form, report, dll).

INTRODUCTION

- **Modul Server** : secara khusus dirancang agar proses dapat berjalan pada sisi komputer server.
- Modul Server biasanya menangani data storage, data access, searching, serta fungsi-fungsi lainnya yang harus dijalankan pada sisi server.

MODEL DATA & SCHEMA

- Salah satu karakteristik dasar utama dari sistem database adalah tersedianya **level abstraksi data**.
- **Level abstraksi data** menyembunyikan detail struktur database sesungguhnya dari pengguna database (user), yang mana mereka memang tidak terlalu peduli / penting / membutuhkan informasi tersebut.
- Untuk itu DBMS harus memiliki **model data (data model)**.
- **Model data** : adalah kumpulan konsep yang digunakan dalam mendeskripsikan struktur database (tipe data, relasi, constraint/batasan yang menjaga data).
- Kebanyakan Model data berupa **operasi-operasi dasar** seperti operasi untuk mendapatkan data (**query**), **update** data pada database, **delete** data, **modifikasi**, dll.
- Umumnya operasi dalam model data menentukan aspek dinamis / tingkah laku dari database.
- Model data dapat berupa:
 - Operasi Umum: insert, update, delete dan query
 - Operasi yang ditentukan sendiri: menghitung nilai IPK dari objek/entitas mahasiswa, dll.

MODEL DATA & SCHEMA...

Kategori Model Data

- **Model data konseptual (level tinggi)**

Pemodelan data berdasarkan pendekatan sisi user.

- **Model data fisik (level rendah)**

Mendeskripsikan detail database bagaimana data itu disimpan.

- **Model data implementasi (representasi) / View**

Memberikan tampilan khusus bagi sisi user yang menyembunyikan detail data yang tersimpan dalam database.

- Model data **konseptual**: dapat berupa entitas, atribut dan relasional.

1. **Entitas** : gambaran dari object / konsep database, misal. Pegawai, project, mahasiswa, dosen. Dan memiliki atribut.
2. **Atribut** : informasi yang melekat pada entitas, seperti entitas mahasiswa memiliki atribut: nama, NRP, alamat, telp, dll.
3. **Relasi** : hubungan antar dua hingga tiga entitas yang saling ber-relasi. Misal. Dosen mengajar Mahasiswa.

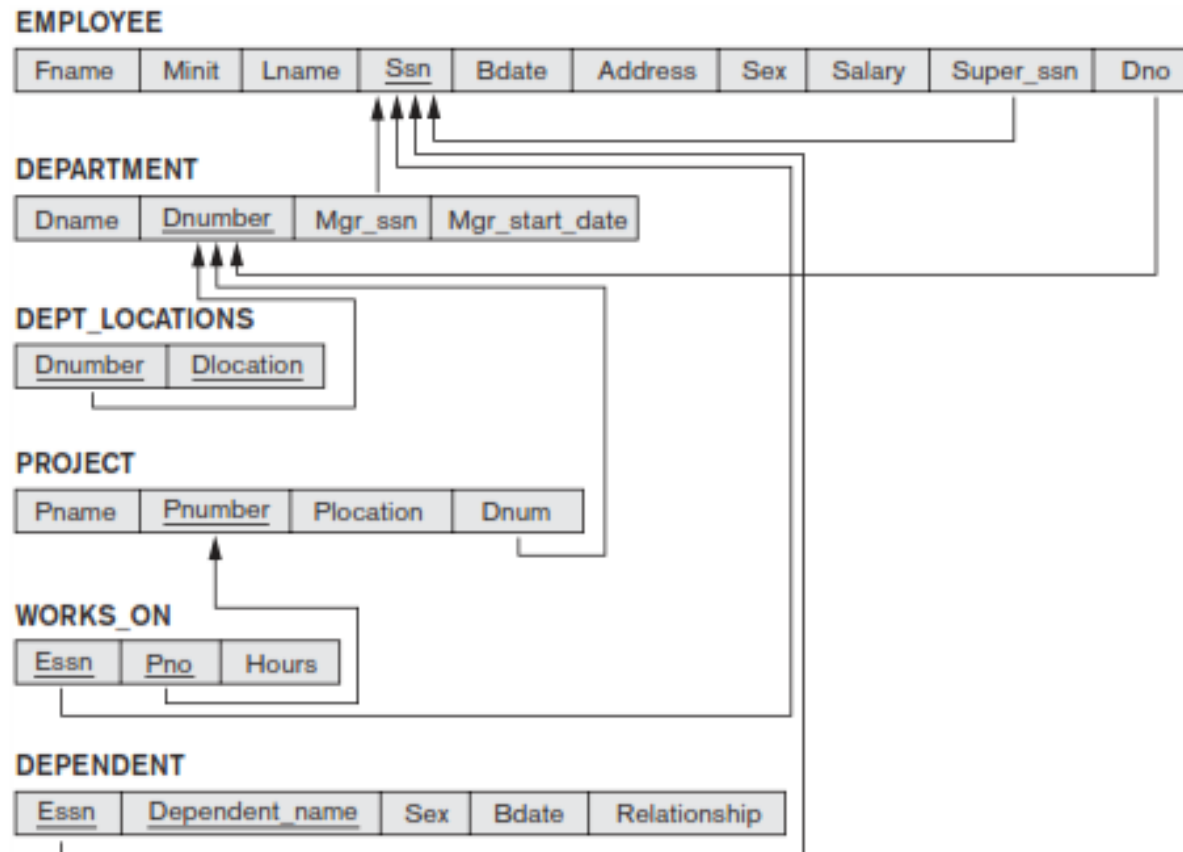
MODEL DATA & SCHEMA...

Skema, Instance dan State Database

- Dalam model data, sangatlah penting untuk dapat membedakan antara **deskripsi database** dan **data** dalam database itu sendiri.
- **Deskripsi database** disebut juga **skema database**.
- Skema database ditentukan selama dalam **perancangan database** dan biasanya tidak selalu mengalami perubahan.
- Dalam skema database, digambarkan dalam bentuk **schema diagram (diagram skema)** yang menampilkan struktur setiap tipe record data pada tiap-tiap entitas, constraint yang terjadi tiap entitas.

MODEL DATA & SCHEMA...

Contoh Diagram Skema Database Relasional





LESSONS

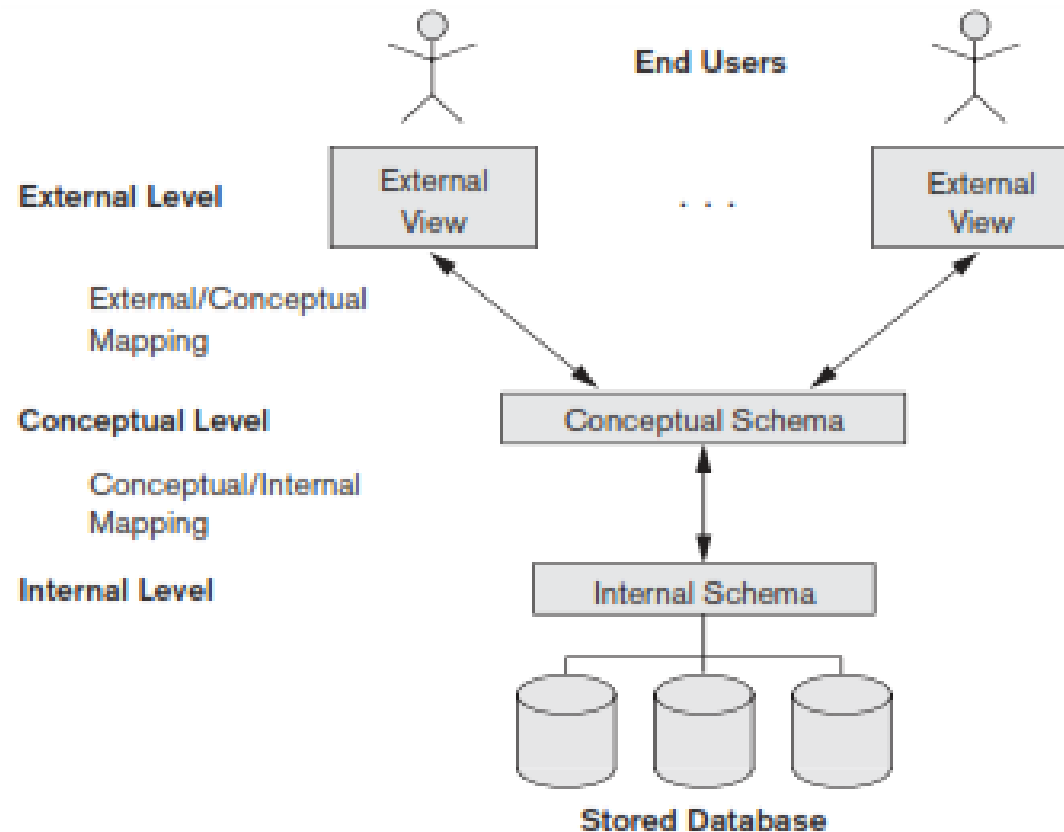
1. Model Basis Data
2. Model Data & Schema
- 3. Arsitektur Sistem Database & Kebebasan Data**
4. Bahasa & Interface Database

ARSITEKTUR SISTEM DATABASE & KEBEBASAN DATA

- Tiga dari empat karakteristik penting database, yaitu
 1. Adanya penyekatan antara program dan data (kebebasan data-program dan operasi-program),
 2. Dukungan multiple view, serta
 3. Menggunakan katalog untuk menyimpan deskripsi database (skema).
- Berdasarkan tiga karakteristik utama database diatas dapat memberikan **gambaran arsitektur sistem database**.
- Arsitektur sistem database dapat digambarkan dalam bentuk **arsitektur tiga-skema (three-schema architecture)**.
- **Tujuan** dari gambaran arsitektur tiga-skema tersebut adalah untuk memisahkan antara sisi aplikasi user dengan **struktur fisik database**.

ARSITEKTUR SISTEM DATABASE & KEBEBASAN DATA

Arsitektur Sistem Database : (gambaran abstraksi data)



ARSITEKTUR SISTEM DATABASE & KEBEBASAN DATA

1. **Level Internal**, memiliki **skema internal**, menggambarkan struktur penyimpanan secara **fisik database**. Skema internal menggunakan model data fisikal dan menggambarkan detail lengkap data storage dan access path database.
2. **Level Konseptual**, memiliki **skema konseptual**, menggambarkan struktur keseluruhan database bagi user. Level konseptual menyembunyikan detail dari struktur penyimpanan fisik dalam mendeskripsikan entitas, tipe data, relasi, constraint dan operasi user.

Biasanya representasi **model data** digunakan dalam **menggambarkan skema konseptual** ketika sistem database dibuat.

3. **Level External / View**, mencakup sejumlah skema eksternal / user views. Level ini menggambarkan pengaksesan database pada group user tertentu dan menyembunyikan sebagian data dalam database dari group user tertentu pula.

ARSITEKTUR SISTEM DATABASE & KEBEBASAN DATA

Data Independence (Kebebasan Data)

- Arsitektur tiga-skema selanjutnya dapat digunakan untuk menjelaskan konsep dari **data independence**.
- **Data independence**, dapat digambarkan sebagai berikut: jika terjadi perubahan skema pada satu level tertentu maka tidak akan mempengaruhi / merubah skema pada level selanjutnya yang lebih tinggi.

ARSITEKTUR SISTEM DATABASE & KEBEBASAN DATA

Ada dua jenis data independence:

- **Logical data independence**

kemampuan dalam merubah **skema konseptual tanpa harus mengubah skema eksternal / program aplikasi**. Jadi, kita dapat merubah skema konseptual untuk menambah database (menambah tipe record / item data), untuk merubah constraint, atau mengurangi database (menghapus record / item data)

- **Physical data independence**

kemampuan dalam merubah **skema internal tanpa harus merubah skema konseptual**.



LESSONS

1. Model Basis Data
2. Model Data & Schema
3. Arsitektur Sistem Database & Kebebasan Data
- 4. Bahasa & Interface Database**

BAHASA DAN INTERFACE DATABASE

- DBMS merupakan **perantara** antara user dengan database.
- Cara komunikasi diatur dalam suatu **bahasa khusus** yang telah ditetapkan oleh DBMS. Contoh: SQL, dBase, QUEL, dsb.
- Bahasa database, dibagi dalam 2 bentuk:
 1. Data Definition Language (**DDL**)
 2. Data Manipulation Language (**DML**)

DDL

Data Definition Language (DDL)

- Digunakan dalam **membuat** tabel baru, indeks, **mengubah** tabel, **menentukan** struktur tabel, dsb.
- Hasil dari kompilasi perintah DDL berupa kumpulan tabel yang disimpan dalam file khusus: **Kamus Data (Data Dictionary)**.
- **Data Dictionary**: merupakan **metadata (superdata)**, yaitu data yang mendeskripsikan data sesungguhnya. Data dictionary ini akan selalu diakses dalam suatu operasi database sebelum suatu file data yang sesungguhnya diakses.

DML

Data Manipulation Language (DML)

- Digunakan dalam **memanipulasi** dan **pengambilan** data pada database.
- Manipulasi data, dapat mencakup:
 1. **Pemanggilan** data yang tersimpan dalam database (**query**)
 2. **Penyisipan/penambahan** data baru ke database (**Insert**)
 3. **Pengubahan** data pada database (**Update**)
 4. **Penghapusan** data dari database (**Delete**)

BAHASA BASIS DATA

Terdapat dua (2) jenis DML:

- **Prosedural**

Menghendaki user untuk **menspesifikasikan** data apa yang diperlukan dan bagaimana cara mendapatkan data itu.

Contoh: bahasa C/C++, PL/SQL, dsb.

- **Nonprosedural**

Menghendaki user untuk menspesifikasikan data apa yang dibutuhkan, **tanpa harus** menspesifikasikan bagaimana cara mendapatkan data tersebut.

Contoh: SQL

Contoh bahasa prosedural → dBase	<pre> USE mahasiswa SET FILTER TO t4lahir = 'Makassar' GO TOP DO WHILE .NOT. EOF() DISPLAY nama SKIP ENDDO USE </pre>	Langkah sangat detail dan terlalu teknis
Contoh bahasa non-prosedural → SQL	<pre> SELECT nama FROM mahasiswa WHERE t4lahir = 'Makassar'; </pre>	Singkat dan tidak perlu langkah yang bersifat prosedural

```

create or replace function native_comp_1 return number as
    v_avg_n    number := 0;
    v_total_n number := 0;
BEGIN
    for i in 1..10000000 loop
        v_total_n := v_total_n + i;
        v_avg_n   := v_total_n/i;
    end loop;
    return(v_avg_n);
END;

create or replace function native_comp_2 return number as
    cursor c1 is
        select *
        from residents;
BEGIN
    for c1_rec in c1 loop
        null;
    end loop;
    return(null);
END;

create or replace function native_comp_3 return number as
    v_salary_n number := 0;
    cursor c1 is

```


BAHASA DAN INTERFACE DATABASE

DBMS Interface

DBMS menyediakan interface yang user-friendly:

- Interface Menu untuk Web Client / Browsing database (menghapus record / item data)
- Interface Form
- GUI
- Natural Language Interface
- Interface untuk User Parametric
- Interface untuk DBA.

kemampuan dalam merubah skema internal tanpa harus merubah skema konseptual.

SISTEM MANAJEMEN BASIS DATA (DBMS)

- Merupakan **perangkat lunak** yang didesain untuk melakukan penyimpanan dan pengaturan basis data
- DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pengendalian keakuratan data, dll.

Mengapa Menggunakan DBMS ?

DBMS diperlukan untuk :

- a. Independensi data dan akses yang efisien
- b. Mereduksi waktu pengembangan aplikasi
- c. Integritas dan keamanan data
- d. Administrasi keseragaman data
- e. Akses bersamaan dan perbaikan dari terjadinya *crash*

KOMPONEN UTAMA DBMS

- Perangkat keras (PC) : Tipe prosesor, kapasitas memory, kapasitas media penyimpanan data.
- Data :
 - ✓ Terpadu ➔ data saling terkait (dependensi data)
 - ✓ Berbagi data ➔ dipakai sejumlah pengguna dalam waktu yang bersamaan (sistem multi-user)
- Perangkat lunak : *Software DBMS*

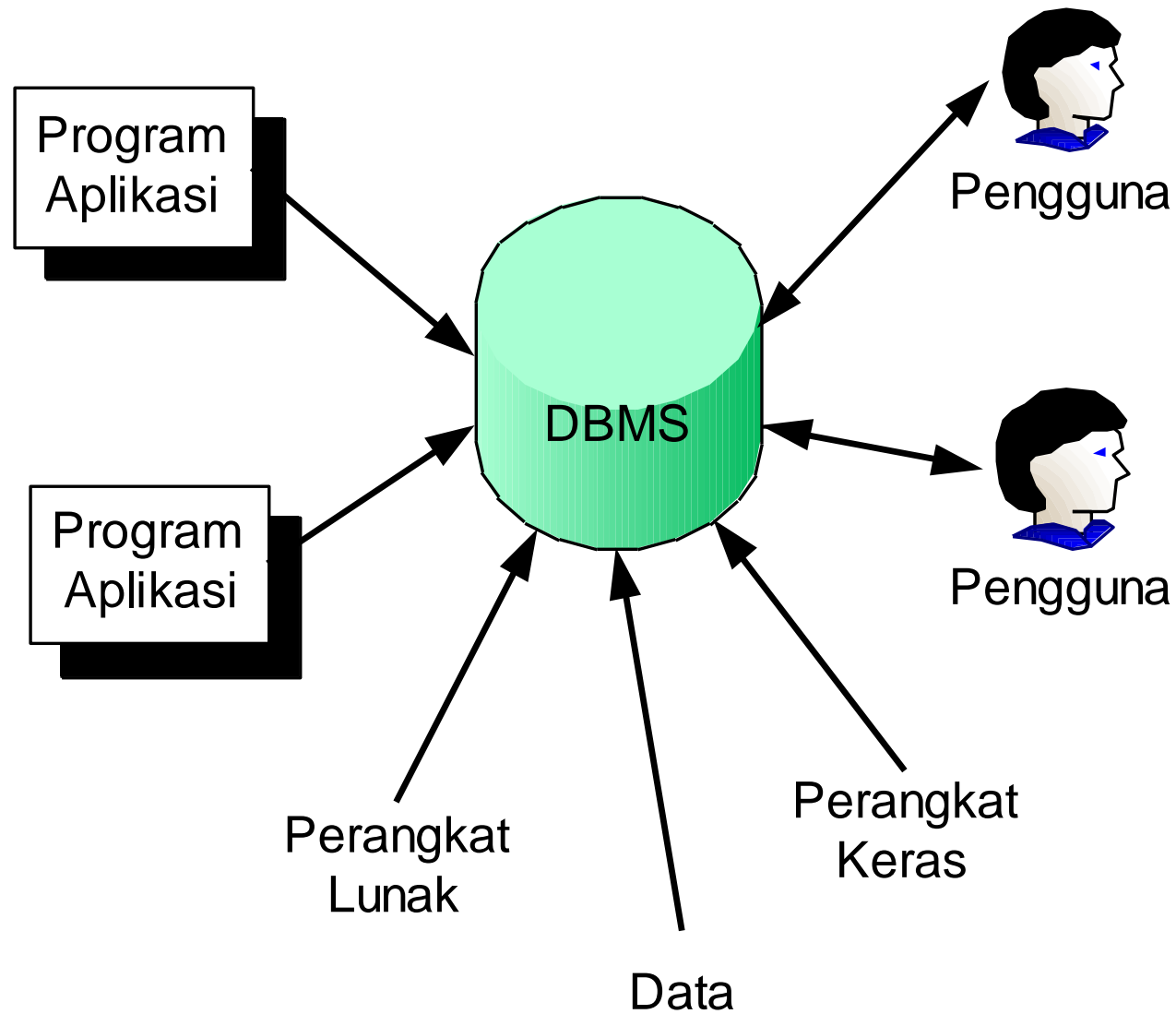
KOMPONEN UTAMA DBMS (LANJUTAN)

➤ Pengguna

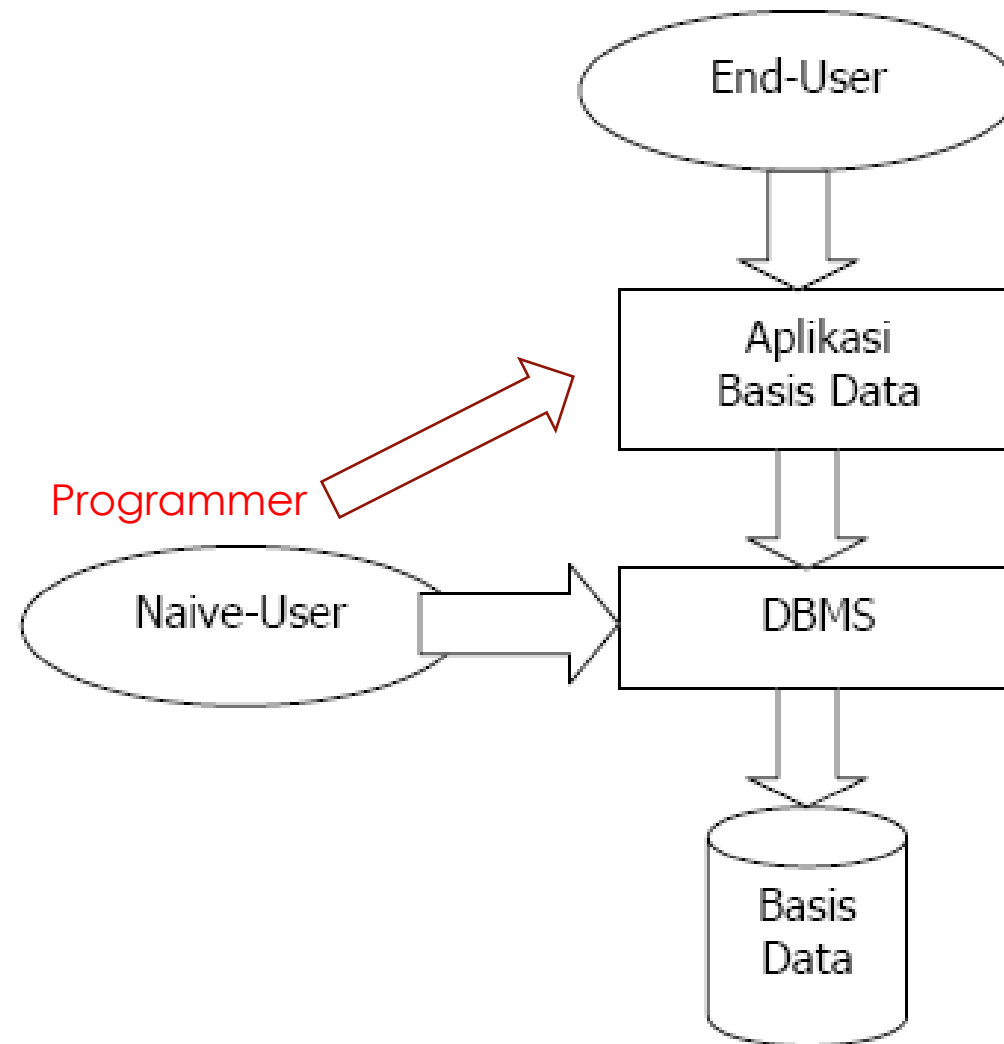
✓ Kategori pengguna

1. Pengguna akhir (*End User*)
2. Pemrogram aplikasi (*Application Programmer*)
3. Administrator Database (*Database Administrator*)

Keterhubungan DBMS



ARSITEKTUR SISTEM MANAJEMEN BASIS DATA (DBMS)





CONTOH DBMS

- DB2 dari IBM
- MySQL
- Microsoft SQL Server
- Microsoft Visual Foxpro
- Oracle

UJI PEMAHAMAN

1. Apa perbedaan antara DML dan DDL?
2. Jelaskan kelebihan dan kekurangan pemodelan basis data menggunakan model jaringan dan relational
3. Sebuah basis data hendak dibuat untuk menggambarkan hubungan antara kamar sebuah rumah sakit, pasien yang dirawat di suatu kamar, serta dokter yang bertugas memeriksa pasien di suatu kamar. Gambarkan model jaringan dan relational untuk basis data tersebut



SELES AI