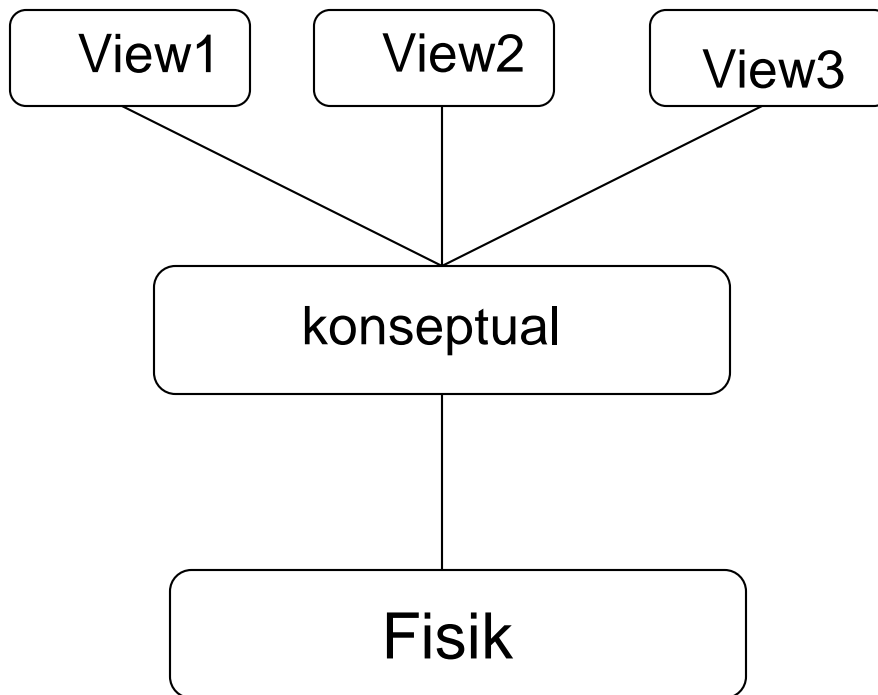


Abstraksi Data

Abstraksi Data : *Merupakan tingkatan dalam bagaimana melihat data dalam Sistem Basis Data*



Level Penampakan : user yang menikmati sebagian dari Basis Data

Menggambarkan data apa yg sebenarnya (secara fungsional) disimpan dlm Basis Data dan hubungannya dg data lain

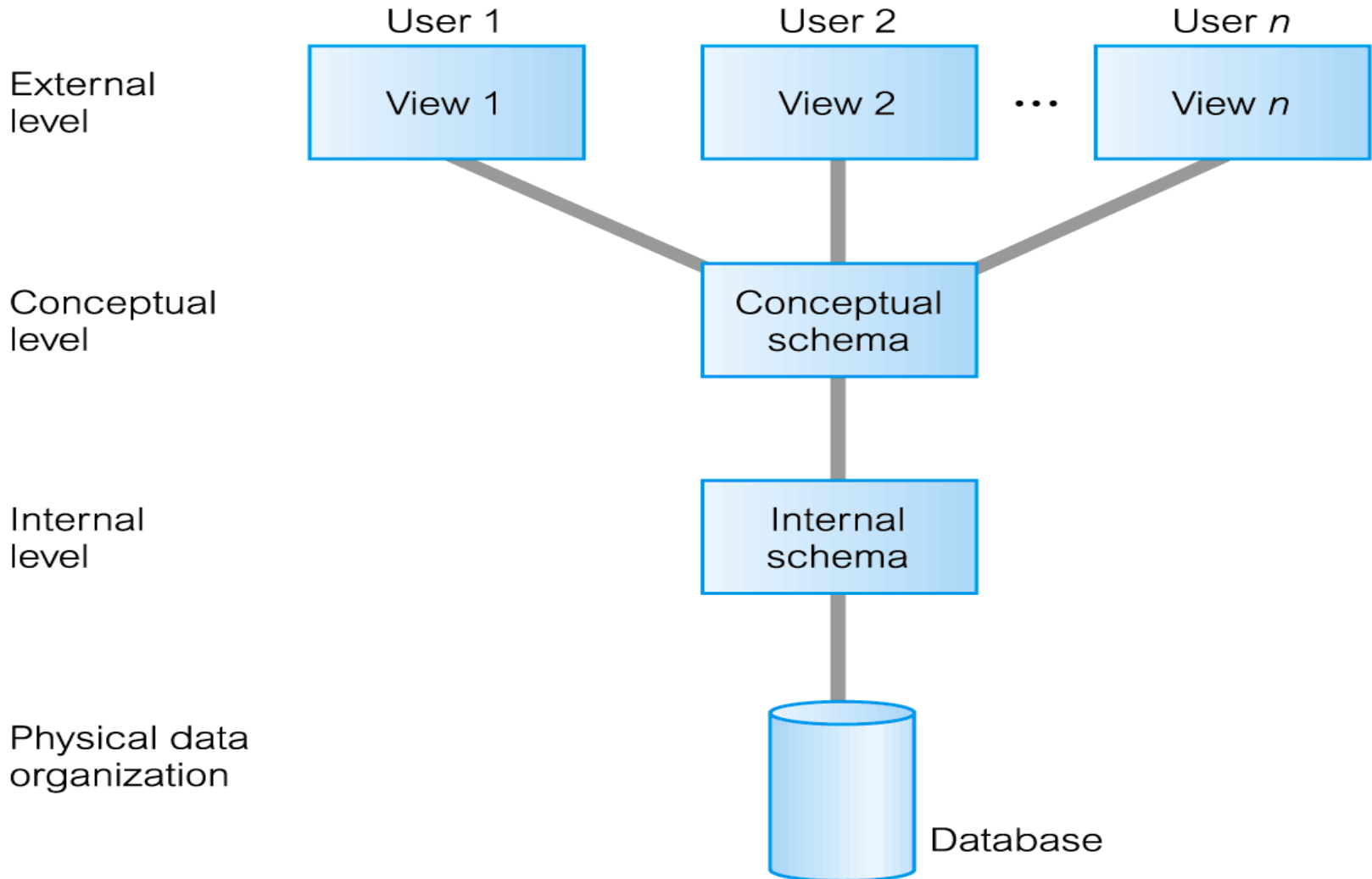
Bagaimana sesungguhnya suatu data disimpan



Abstraksi Data

- Keunggulan DBMS adalah menyembunyikan kompleksitas dari struktur data.
- Abstraksi Data terdiri dari tiga tingkatan:
 1. Level View/External Level
 2. Level Konseptual/Conceptual Level
 3. Level Fisik/Physical atau Internal Level

Three-Level Architecture/Abstraksi Data





Three-Level Architecture

- External Level

- Users' view of the database.
- Describes that part of database that is relevant to a particular user.

- Conceptual Level

- Community view of the database.
- Describes what data is stored in database and relationships among the data.



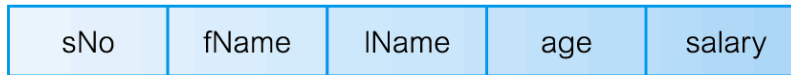
Three-Level Architecture

- Internal Level

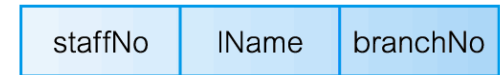
- Physical representation of the database on the computer.
- Describes how the data is stored in the database.

Differences between Three Levels of Architecture

External view 1



External view 2



Conceptual level



Internal level

```
struct STAFF {  
    int staffNo;  
    int branchNo;  
    char fName [15];  
    char lName [15];  
    struct date dateOf Birth;  
    float salary;  
    struct STAFF *next;           /* pointer to next Staff record */  
};  
index staffNo; index branchNo; /* define indexes for staff */
```



✓ **Conceptual view:**

merupakan pandangan yang berkaitan dengan permasalahan data-data apa saja yang diperlukan untuk disimpan dalam basis data dan penjelasan mengenai hubungan antar data yang satu dengan lainnya. Conceptual view dapat disetarakan dengan schema, dilakukan [database administrator](#)

✓ **Physical view:**

merupakan bentuk implementasi dari conceptual view, yaitu pandangan tentang bagaimana data disimpan dalam media penyimpan data

✓ **User view:**

dapat disejajarkan dengan sub-schema



Sistem Basis Data

Bahasa Basis Data : *Merupakan cara berinteraksi pemakai dg basis data yang sesuai dengan aturan ditetapkan oleh pembuat DBMS*

Bagian Basis Data :

- Data Definition Language (DDL)

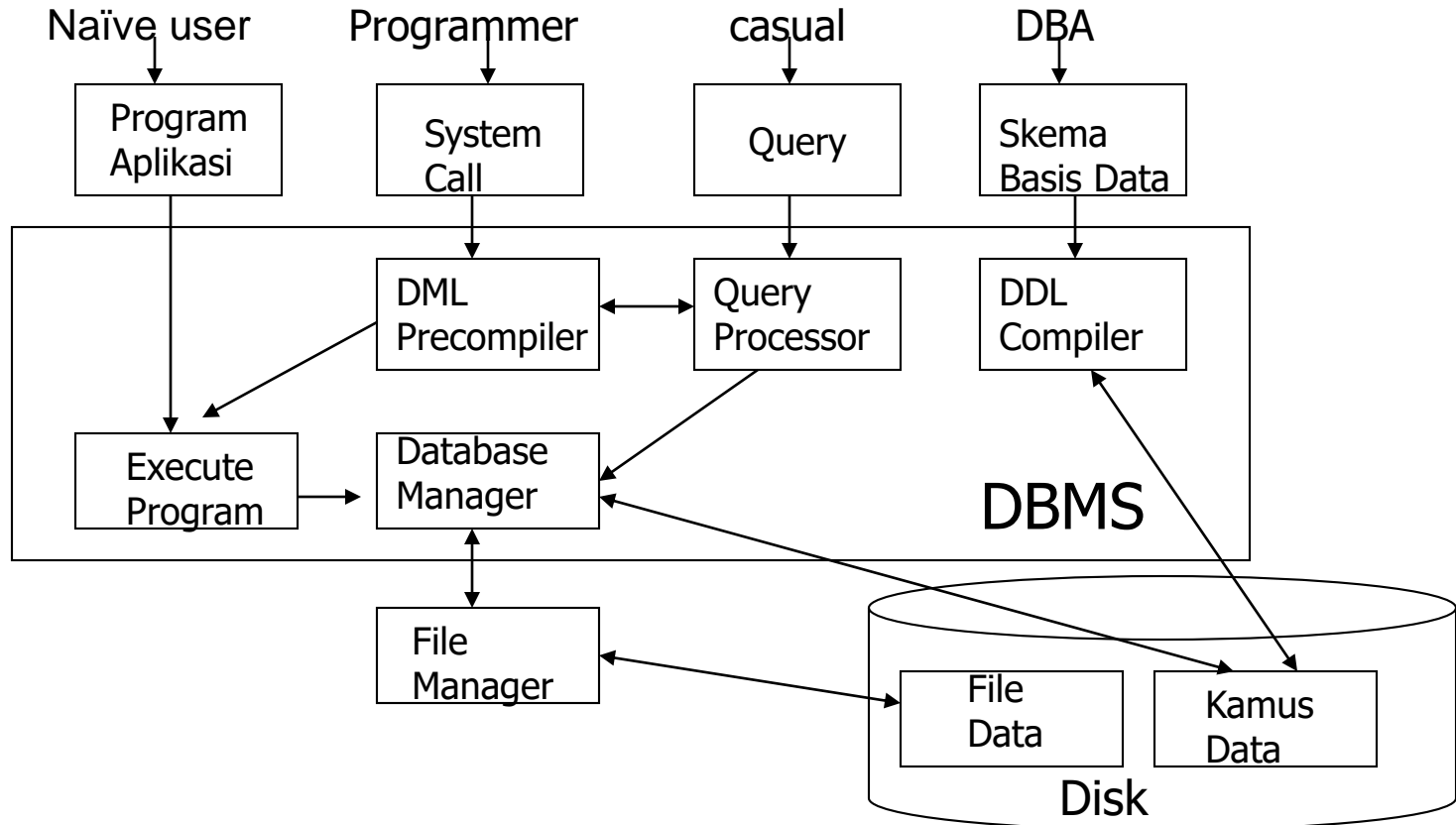
Bahasa yg digunakan untuk menggambarkan basis data secara keseluruhan, Operasi yang dapat dijalankan adalah membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur. Hasil kompilasi dari DDL ini adalah Kamus Data (Data Dictionary)

- Data Manipulation Language (DML)

Bahasa untuk melakukan manipulasi dan pengambilan data pada suatu basis data, seperti : penyisipan, penghapusan dan update.

Sistem Basis Data

Struktur Sistem Keseluruhan :





Sistem Basis Data

Keterangan :

File manager : mengelola alokasi ruang dalam disk dan struktur data dalam disk

Database manager : menyediakan interface antara low level Dengan prog. Aplikasi dan query

Query processor : menterjemahkan perintah dlm query Language ke perintah low level yang dimengerti oleh Database manager

DML precompiler : mengkonversi perintah DML yang ada Di prog. Aplikasi ke pemanggilan prosedur normal dlm Bahasa induk

DDL compiler : mengkonversi perintah-perintah DDL Kedalam sekumpulan tabel yang mengandung metadata. Tabel ini kemudian disimpan dalam kamus data



Model Data Dalam SBD

Model Data : Adalah kumpulan perangkat konseptual untuk menggambarkan Data, hubungan antar data, semantik dan batasan data.

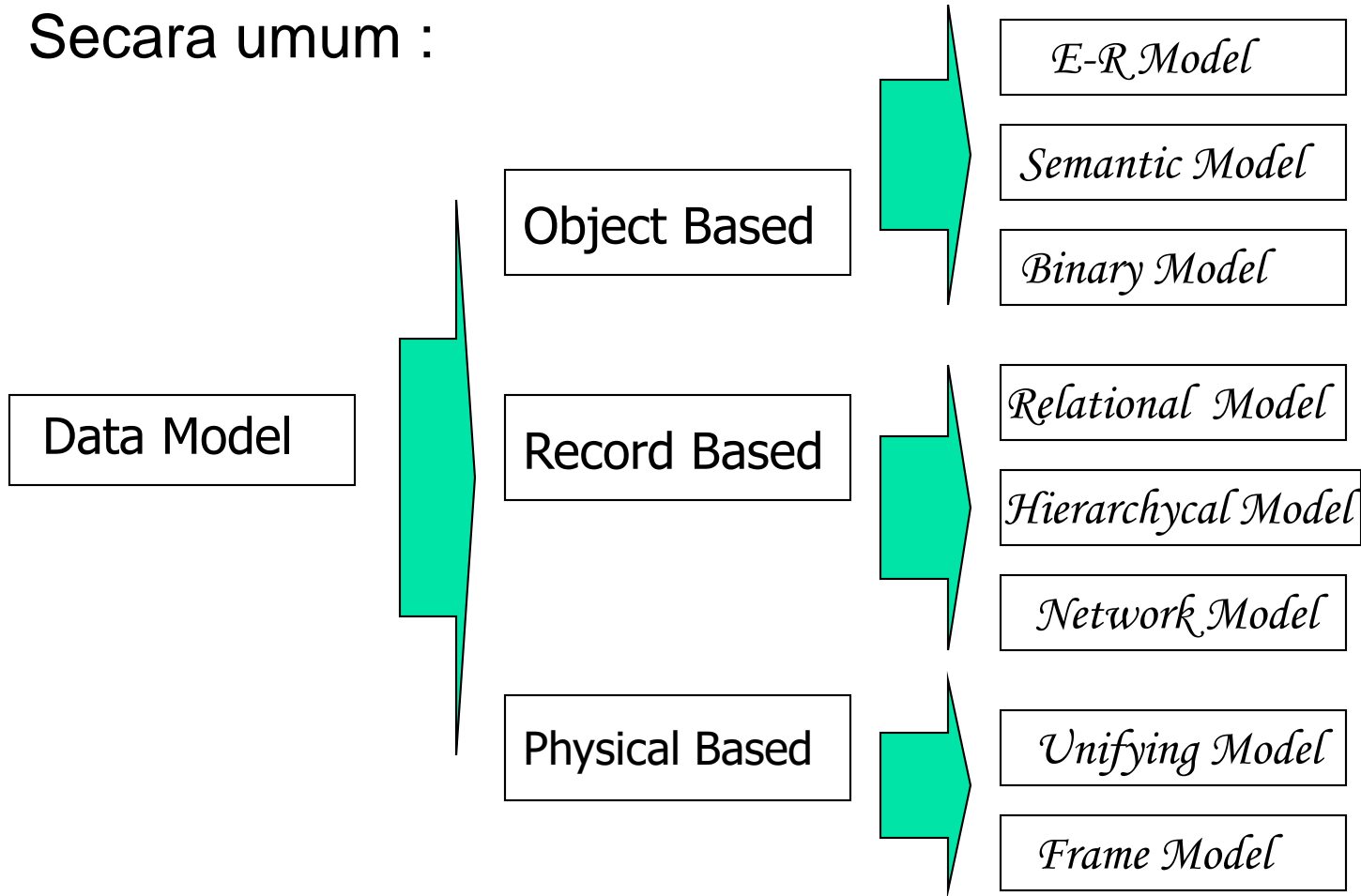
Model data biasanya digunakan untuk perancangan basis data, yang disebabkan karena kelangkaan data / fakta yang dimiliki

Henry F. Korth : Ada 2 kelompok model data, yaitu :

- ❖ Model data berbasis object
- ❖ Model data berbasis record

Model Data Dalam SBD

Secara umum :

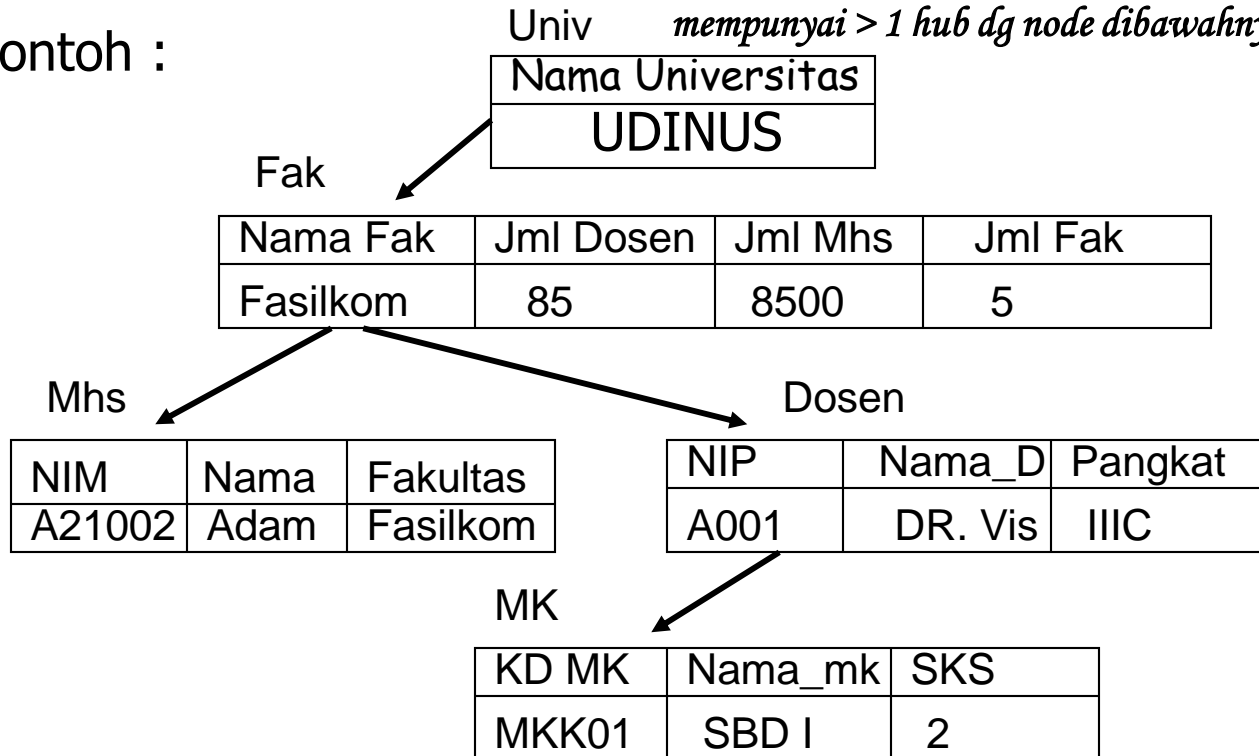


Model Data Dalam SBD

Model Data Hirarkis :

Menjelaskan hub logik antar data dlm bentuk hub bertingkat. Elemen penyusun disebut node, node tertinggi: root. Suatu node yg lebih rendah hanya diijinkan memp, 1 hub dg node yg lebih tinggi, disebut parent. Parent dapat mempunyai > 1 hub dg node dibawahnya.

Contoh :

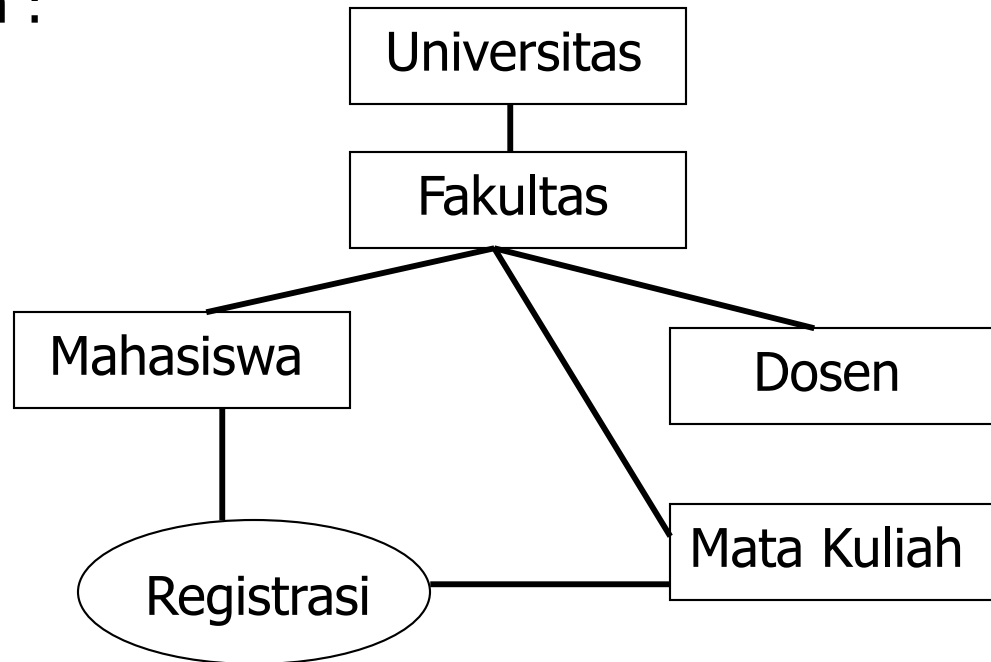


Model Data Dalam SBD

Model Data Jaringan :

Distandarisasi oleh Data Base Task Group (DBTG) tahun 1971. Hampir sama seperti hirarkis, tetapi dalam model jaringan suatu node dibawahnya bisa memp hub dengan > 1 node diatasnya

Contoh :





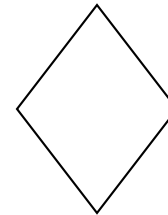
Model Data Dalam SBD

E-R model : *Model yang menjelaskan hub antar data dalam basis data berdasarkan suatu persepsi bahwa real world terdiri dari obyek-obyek dasar yang mempunyai relasi antara obyek-obyek tersebut*

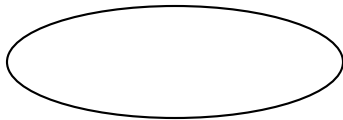
Simbol-simbol :



entitas



Relasi / hubungan



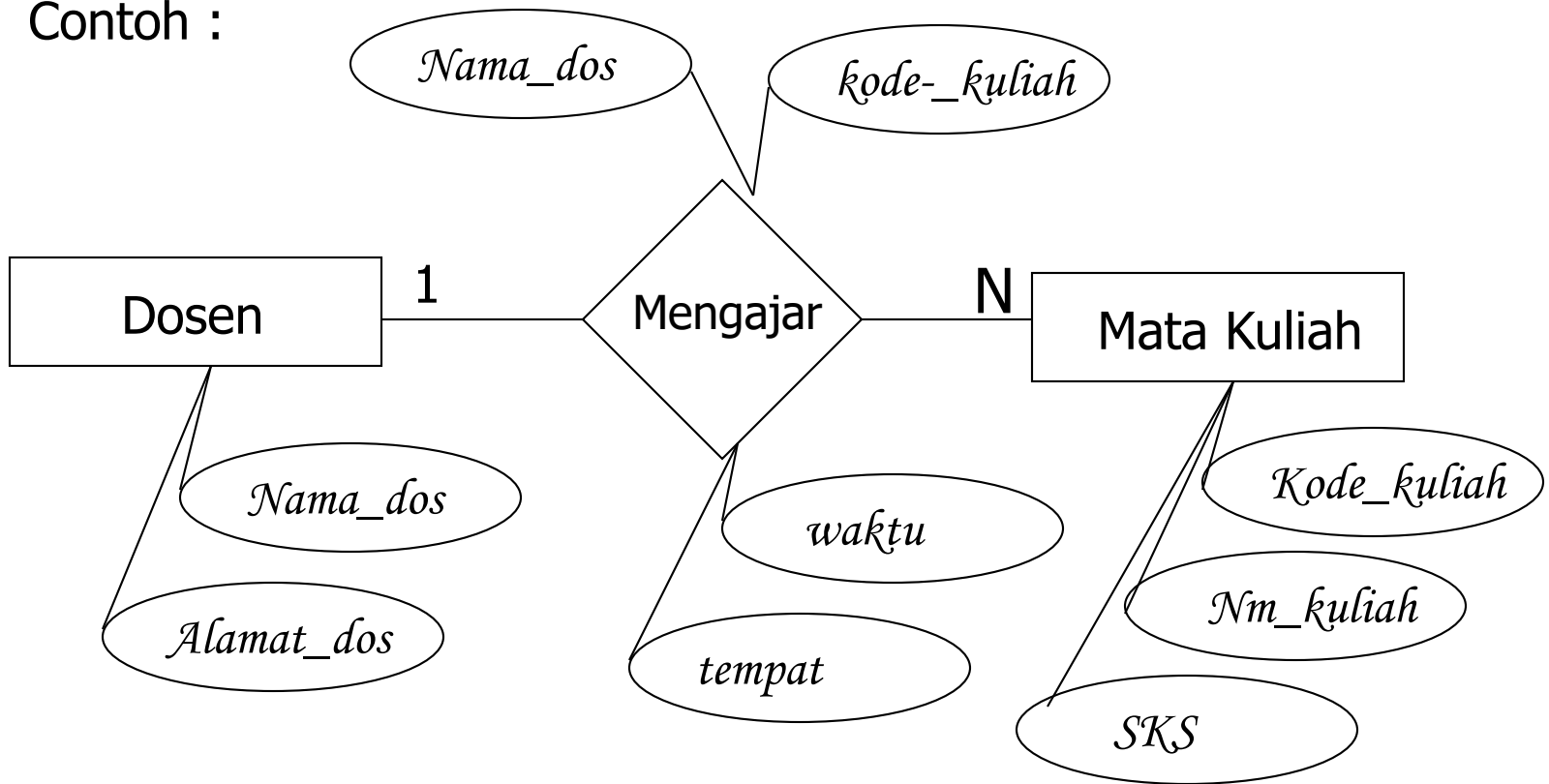
atribut



Garis hubung

Model Data ERD Dalam SBD

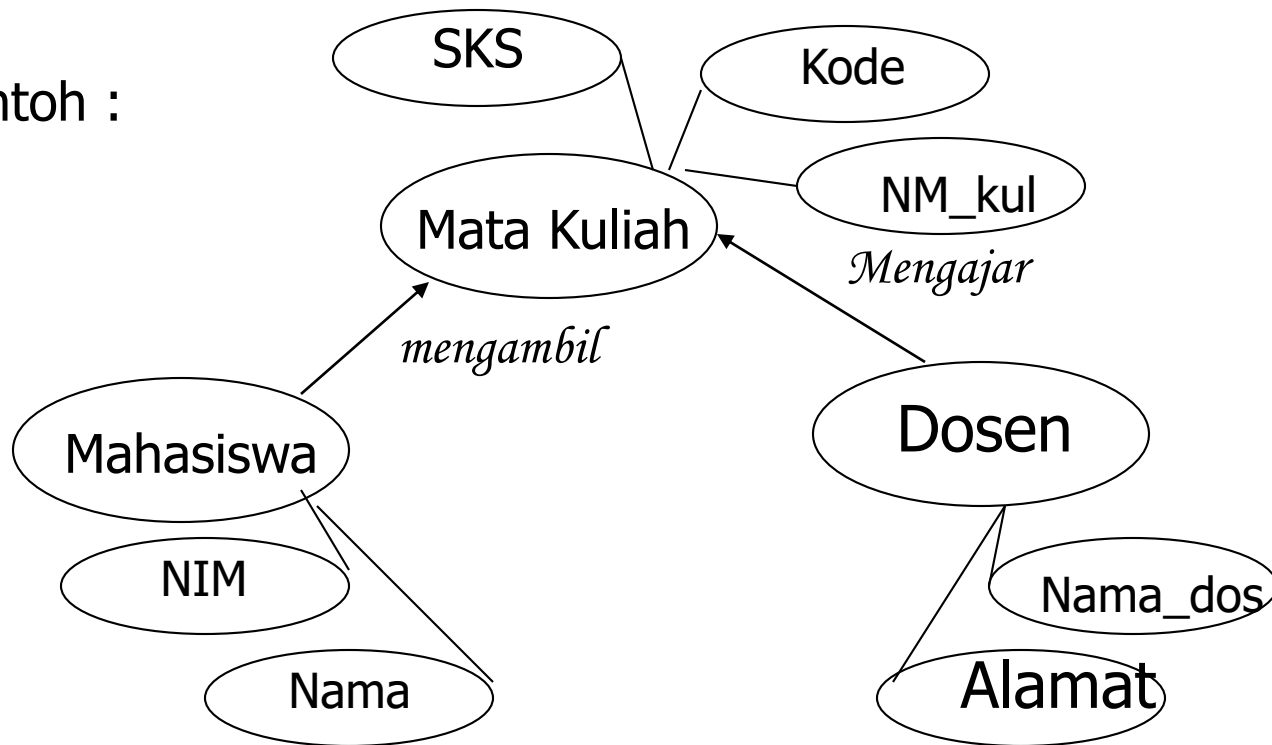
Contoh :



Model Data Dalam SBD

Model Data Semantic : *Hampir sama seperti E-R Model, tetapi relasi antar obyek dasar dinyatakan dengan kata-kata (semantik).*

Contoh :





Model Data Relasional

Disebut juga : Model Relasional atau Basis Data Relasional (ditemukan oleh : E.F. Codd)

Menunjukkan suatu cara yang digunakan untuk mengelola data secara fisik dalam memori sekunder dan bagaimana bentuk relasi dari keseluruhan data dalam sistem yang sedang ditinjau

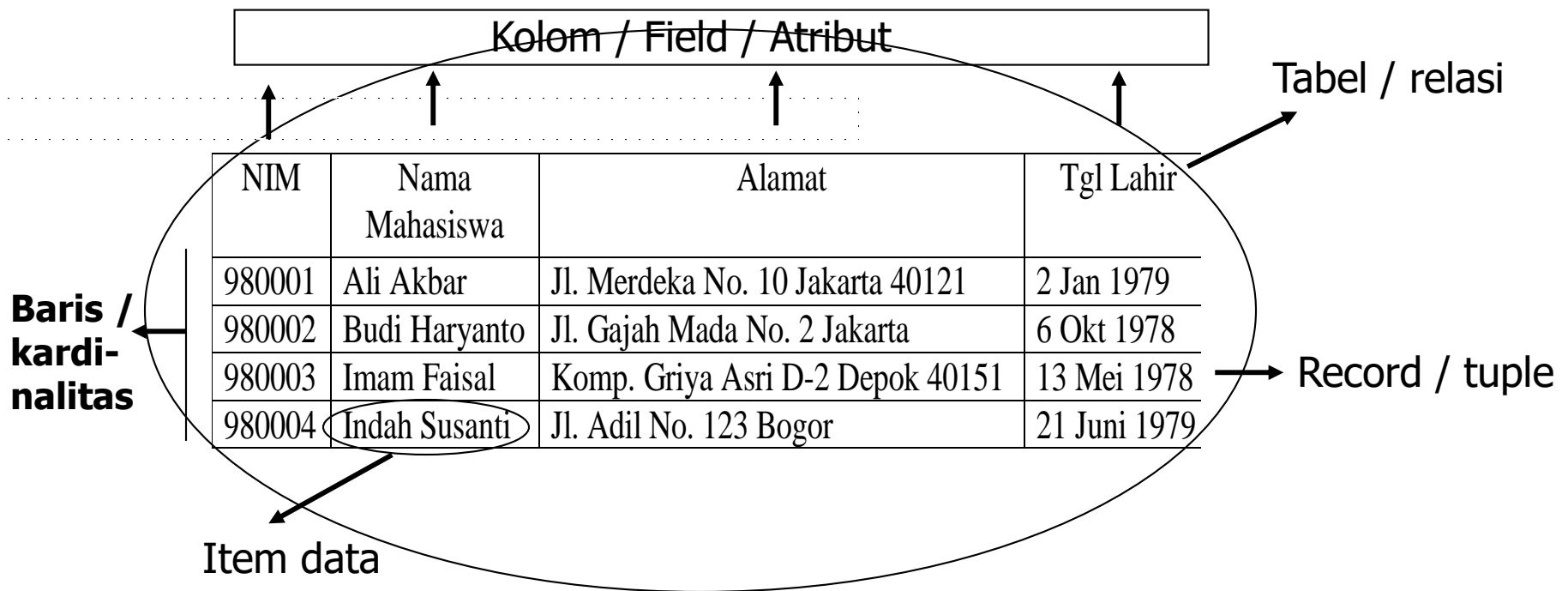
Kelebihan :

- ‡ *Dapat mengakomodasi berbagai kebutuhan pengelolaan basis data yg ada di dunia nyata (real word)*
- ‡ *Pencarian data dari suatu tabel atau banyak tabel dapat dilakukan dengan cepat*
- ‡ *Merupakan model yang paling sederhana sehingga mudah untuk dipahami*

Model Data Relasional

Basis Data akan disebar / dipilah ke dalam tabel dua dimensi

Contoh :





Model Data Relasional

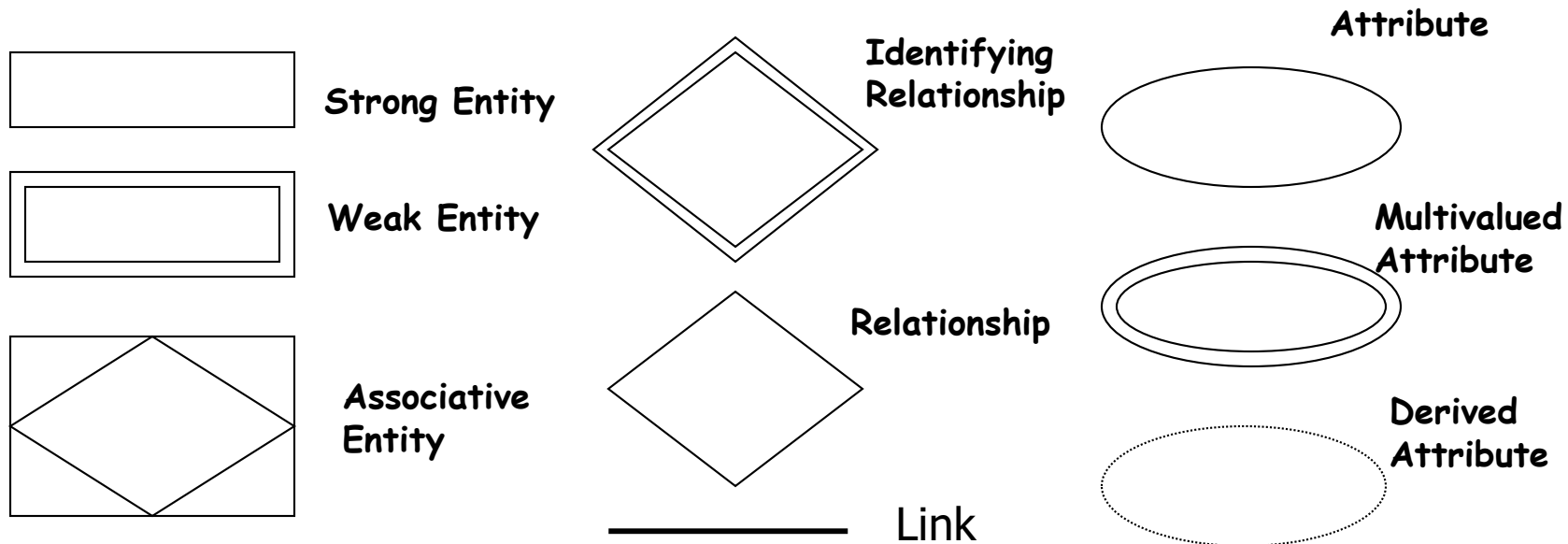
Istilah-istilah:

- \$ RDMS (relational Database Management System) adalah DBMS yang bermodelkan relasional
- \$ Kardinalitas adalah jumlah tupel suatu relasi (tabel)
- \$ Derajad / degree / arity adalah jumlah atribut suatu relasi
- \$ Domain adalah Himpunan / batasan nilai yang berlaku bagi suatu atribut
misalnya : Domain atribut nilai adalah A, B, C, D, dan E
Domain tidak sama dengan tipe data suatu atribut
- \$ Istilah relasi biasanya untuk membicarakan struktur logis sedangkan istilah tabel biasanya untuk basis data fisik

E-R Model (Model Keterhubungan Entitas)

Semesta data di dunia nyata ditransformasikan ke dalam sebuah diagram dengan memanfaatkan perangkat konseptual disebut dengan ERD (Entity Relationship Diagram).

Simbol / Notasi E-R Diagram :





E-R Model (Model Keterhubungan Entitas)

Entity (Entitas)

Merupakan individu yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lainnya (individu : manusia, tempat, obyek, kejadian, konsep). Biasanya berhub. Dg baris dlm sebuah tabel).

Entity Sets (Himpunan Entitas) :

Sekelompok entitas yang sejenis dan berada dalam lingkup yang sama

Contoh :

Himpunan Entitas : Pelanggan

Entitas : Budiman, Suherman dll

Himpunan Entitas : Mobil

Entitas : Mobil Suzuki, Mobil Honda dll

Himpunan Entitas : Mahasiswa

Entitas : Ali, Budi, Iman dll



E-R Model (Model Keterhubungan Entitas)

Contoh :



NIM	Nama Mahasiswa	Alamat	Tgl Lahir
980001	Ali Akbar	Jl. Merdeka No. 10 Jakarta 40121	2 Jan 1979
980002	Budi Haryanto	Jl. Gajah Mada No. 2 Jakarta	6 Okt 1978
980003	Imam Faisal	Komp. Griya Asri D-2 Depok 40151	13 Mei 1978
980004	Indah Susanti	Jl. Adil No. 123 Bogor	21 Juni 1979

Entitas 1
Entitas 2
Entitas 3
Entitas 4

Himpunan Entitas



E-R Model (Model Keterhubungan Entitas)

Atribut (attribute / Properties)

Merupakan karakteristik dari sebuah entitas (biasanya berhubungan dengan field dalam sebuah tabel). Penentuan atribut bagi suatu entitas didasarkan pada relevansinya terhadap entitas tersebut.

Atribut Kunci / Identifikasi :

Merupakan atribut pengidentifikasi entitas yang paling unik untuk semua entitas dalam himpunan entitas

Contoh : Atribut NIM pada Himp. Entitas mahasiswa

Atribut Deskriptif :

Merupakan atribut lain selain atribut kunci yang berfungsi sebagai penjelasan terhadap entitas dalam himpunan entitas

Contoh : Atribut nama, alamat, tgl_lahir pada Himp. Entitas MHS



E-R Model (Model Keterhubungan Entitas)

Ada tiga macam kunci :

® **Superkey**

Adalah satu / lebih atribut yg dapat membedakan entitas satu dengan lainnya dalam himp entitas

® **Candidate Key**

Merupakan kumpulan atribut minimal yang dapat membedakan entitas satu dengan lainnya dalam himp entitas.

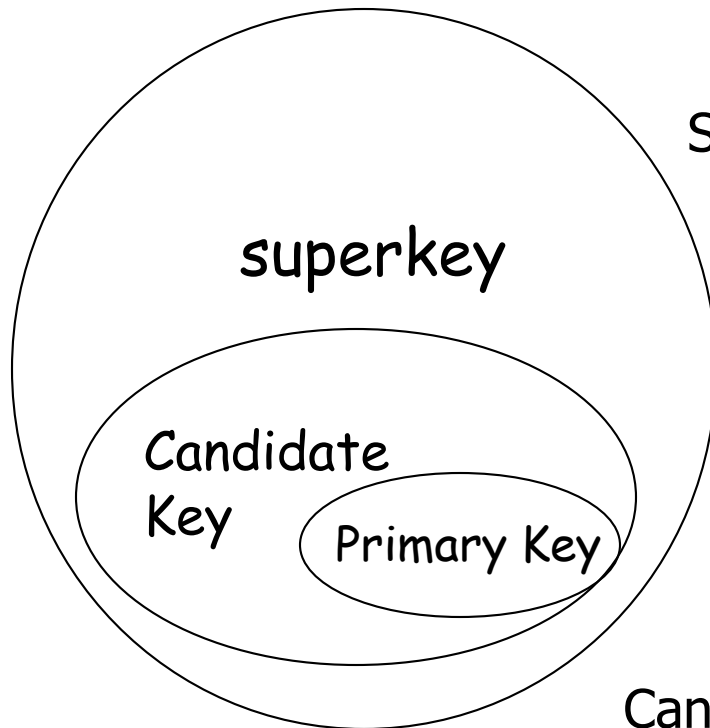
® **Primary Key**

Salah satu dari candidate key yang digunakan sebagai pengidentifikasi suatu entitas dalam himp entitas.

E-R Model (Model Keterhubungan Entitas)

Contoh :

No_KTP	No_SIM	Nama	Alamat



Superkey : No_KTP+No_SIM+Nama+Alamat

No_KTP+No_SIM+Nama

No_KTP+No_SIM+alamat

No_KTP+No_SIM

No_KTP+nama

No_KTP+alamat

No_SIM+ nama

No_SIM+ alamat

No_KTP

No_SIM

Primary Key:

No_KTP atau No_SIM
tergantung kebutuhan

Candidate Key:

No_KTP

No_SIM



E-R Model (Model Keterhubungan Entitas)

Dasar pemilihan Primary Key :

1. Key sering digunakan sebagai acuan
2. Key lebih ringkas
3. Key adalah unik

Atribut Sederhana (Simple Attribute) :

atribut atomik yg tidak dapat di pilah lagi

Atribut Komposit (Composite Attribute) :

atribut atomik yg dapat di pilah lagi

Contoh :

Atribut nama : atribut sederhana (nilai sudah paling kecil / atomik)

Atribut alamat : atribut komposit, karena masih dapat dipilah-pilah lagi menjadi atribut : jalan, kota dan kode_pos



E-R Model (Model Keterhubungan Entitas)

Atribut bernilai banyak (multivalued attribute) :

Merupakan atribut yang dapat bernilai lebih dari 1 nilai yang sejenis

Atribut bernilai tunggal (Single-valued attribute) :

Merupakan atribut yang hanya mempunyai satu nilai

Contoh :

NIM	Nama	Alamat	Hobi
98001	Rudi	Jl. Seroja	Renang Nonton
98002	Wati	Dago Raya	Tidur

NIM, Nama dan Alamat :
atribut bernilai tunggal

Hobi : atribut bernilai banyak

Atribut Turunan (Derived attribute) :

Merupakan atribut yang nilainya diperoleh dari pengolahan atau diturunkan dari atribut / tabel lain



E-R Model (Model Keterhubungan Entitas)

Contoh :

NIM	Nama	Alamat	Angkatan	IPK
98001	Andi	Jl. X	1998	3.2
99011	Susi	Jl. Y	1999	3.0

Angkatan, IPK :

Atribut turunan

Atribut harus bernilai (Mandatory Attribute) :

Merupakan atribut-atribut yang harus diisi nilainya

Atribut tidak harus bernilai (Non Mandatory Attribute / Null) :

Merupakan atribut-atribut yang nilainya boleh dikosongi

E-R Model (Model Keterhubungan Entitas)

Relasi (Relationship) :

Digunakan untuk menunjukkan hubungan antar entitas

Himpunan Relasi (Relationship Sets) :

Merupakan kumpulan semua relasi diantara entitas

Contoh :

Mahasiswa

NIM	Nama	...
98001	Andi	...
98003	Rudi	...
98013	Susi	...

Mata Kuliah

Kode_kul	Nama_kul	sks
A01	Pancasila	2
A03	Internet I	2
A02	Network I	2

Dari tabel-tabel diatas, dapat dilihat bahwa terdapat hubungan / relasi antara himp entitas mahasiswa dengan mata kuliah.

--> Andi mempelajari mata kuliah Internet I

--> Rudi mempelajari mata kuliah Internet I dan Network I

E-R Model (Model Keterhubungan Entitas)

Sehingga apabila dimodelkan dengan E-R Diagram :

