

# Basis Data Relasional

Oleh : Rahmi Hidayati, S.Kom., M.Cs

# MODEL DATA

- ▶ Model data merupakan suatu cara untuk menjelaskan tentang data-data yang tersimpan dalam basis data dan bagaimana hubungan antar data tersebut untuk para pemakai secara logik.
- ▶ Tujuan model data adalah untuk menyajikan data agar mudah dimodifikasi dan dimengerti.



# Jenis Model Data

---

- ▶ Secara garis besar model data dikelompokkan dalam 3 macam yaitu :
  - ▶ Model data berbasis obyek (*object based data model*)
  - ▶ Model data berbasis record (*record based data model*)
  - ▶ Model data fisik (*physical based data model*)



# Model Data Berbasis Obyek (*Object Based Data Model*)

---

Merupakan himpunan Data dan Prosedur atau Relasi yang menjelaskan hubungan Logik antar data dalam suatu Basis Data berdasarkan obyek datanya.

Terdiri Atas :

- *Entity Relationship Model*
- *Semantic Model*
- *Binary Model*



# Model Data Berbasis Record (*Record Based Data Model*)

---

- ▶ Model ini berdasarkan Record atau Rekaman untuk menjelaskan kepada pemakai mengenai hubungan Logik antar data dalam Basis Data.

Terdiri Dari :

- *Hierarchycal Model*
- *Network Model*
- *Relational Model*



# Model Data Fisik

## *(Physical Based Data Model)*

---

Model ini digunakan untuk menguraikan data di tingkat *Internal* atau menjelaskan kepada pemakai bagaimana data-data dalam Basis Data di simpan dalam media penyimpanan secara fisik.

Model ini jarang digunakan karena kerumitan dan kompleksitasnya yang justru menyulitkan pemakai.

Model Ini Terdiri Dari :

- *Unifying Model*
- *Frame Memory*



# Konsep *Relational Model*

---

- ▶ Model Basis Data Relasional atau Model Relasional atau Basis Data Relasional.
- ▶ Model Relasional pertama kali dicetuskan oleh Dr. E.F. Codd di IBM pada tahun 1970 dalam paper dengan judul : "A Relational Model for Large Shared Data Banks," Communications of the ACM, June 1970.
- ▶ Merepresentasikan data pada basis data sebagai kumpulan dari relasi-relasi (*relations*).



# Mengapa model relasi di perlukan untuk perancangan basis data ?

---

Mempunyai piranti komunikasi yang baik antara *user* & *designer*

Model relasional mendefinisikan salah satu kriteria perancangan basis data yang penting yaitu relasi bentuk normal.

Struktur data yang direpresentasikan oleh relasi dapat segera dikonversikan & diimplementasikan ke RDBMS.





# Pengertian Relasi

---

Relasi pertama kali didefinisikan menggunakan teori himpunan.

Cara termudah untuk mendefinisikan sebuah relasi adalah sebagai sebuah tabel dimana data-datanya disimpan dalam baris tabel.



# Basis Data Relasional

---

Basis Data Relasional menggunakan tabel dua dimensi yang terdiri atas baris dan kolom untuk memberi gambaran sebuah berkas data.

- ▶ Data disusun dalam bentuk tabel-tabel
- ▶ Antar dua tabel bisa punya hubungan



# Contoh Model Relasional

NO_MHS	NAMA_MHS
55	Ashadi
56	Rina
57	Budi

KODE_MK	NAMA_MK
DB001	Pengantar Basis Data
DB002	Basis Data Lanjut
PI001	Teknik Multimedia

NO_MHS	KODE_MK	NILAI
55	DB001	A
55	PI001	B
56	DB001	B
57	DB001	A
57	DB002	A

Hubungan

# Contoh Informasi yang Bisa Diperoleh

NO_MHS	NAMA_MHS
55	Ashadi
56	Rina
57	Budi

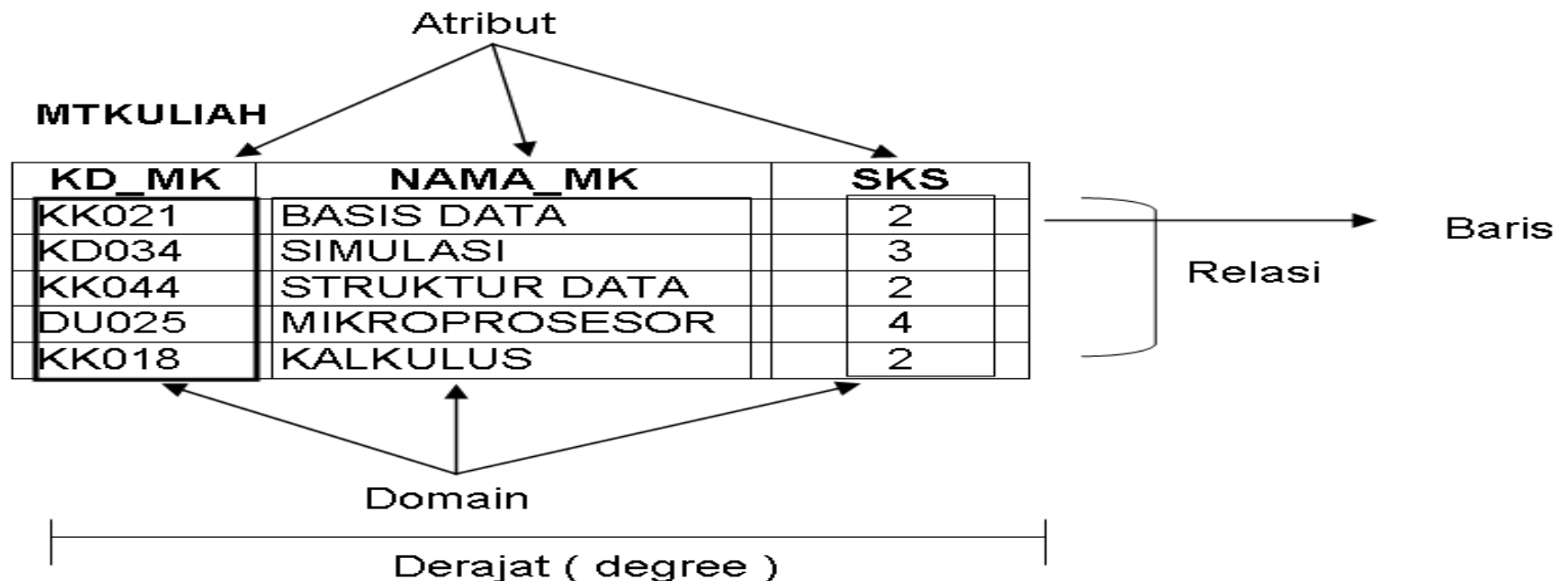
KODE_MK	NAMA_MK
DB001	Pengantar Basis Data
DB002	Basis Data Lanjut
PI001	Teknik Multimedia

NO_MHS	KODE_MK	NILAI
55	DB001	A
55	PI001	B
56	DB001	B
57	DB001	A
57	DB002	A

NAMA_MHS	NAMA_MK	NILAI
Ashadi	Pengantar Basis Data	A
Ashadi	Teknik Multimedia	B
Rina	Pengantar Basis Data	B
Budi	Pengantar Basis Data	A
Budi	Basis Data Lanjut	A

# Istilah dalam Basis Data Relasional

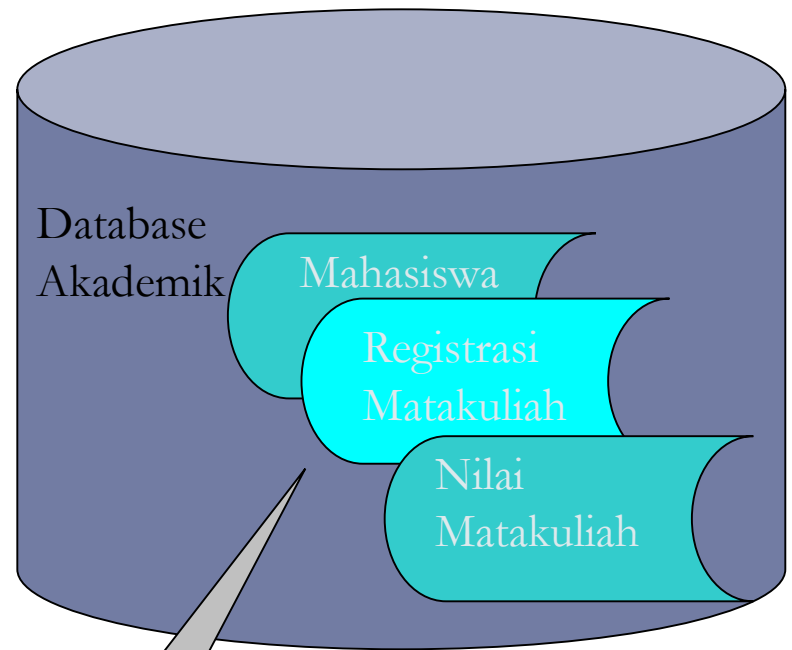
- Relasi** : Sebuah tabel yang terdiri dari beberapa kolom dan beberapa baris
- Atribut** : Kolom pada sebuah relasi
- Tupel** : Baris pada sebuah relasi
- Domain** : Kumpulan nilai yang valid untuk satu atau lebih atribut
- Derajat (*degree*)** : Jumlah atribut dalam sebuah relasi
- Cardinality** : Jumlah tupel dalam sebuah relasi



# Istilah Dasar

---

- ▶ Sebuah *database* terdiri atas sejumlah tabel
- ▶ Tabel sering disebut sebagai **relasi**
- ▶ Setiap tabel menyimpan sekumpulan data atau **baris**

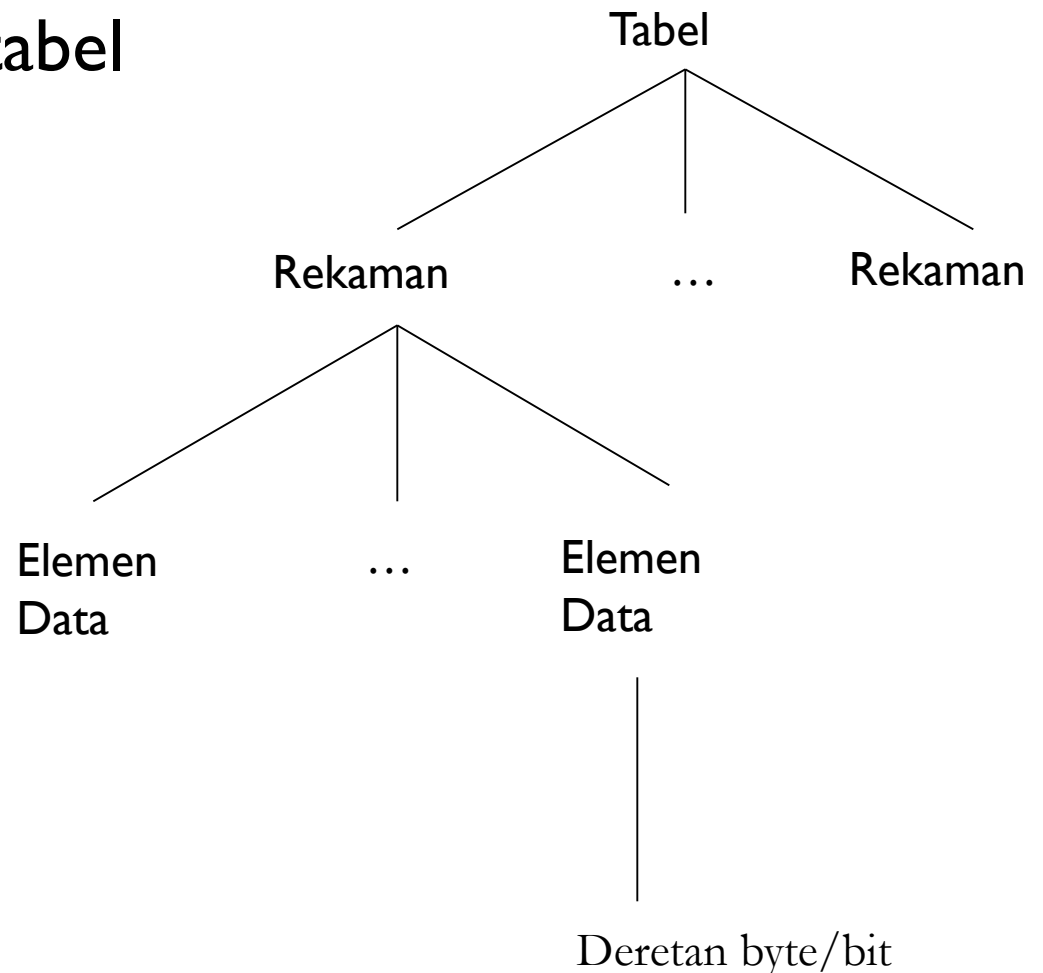


Contoh ada  
3 tabel

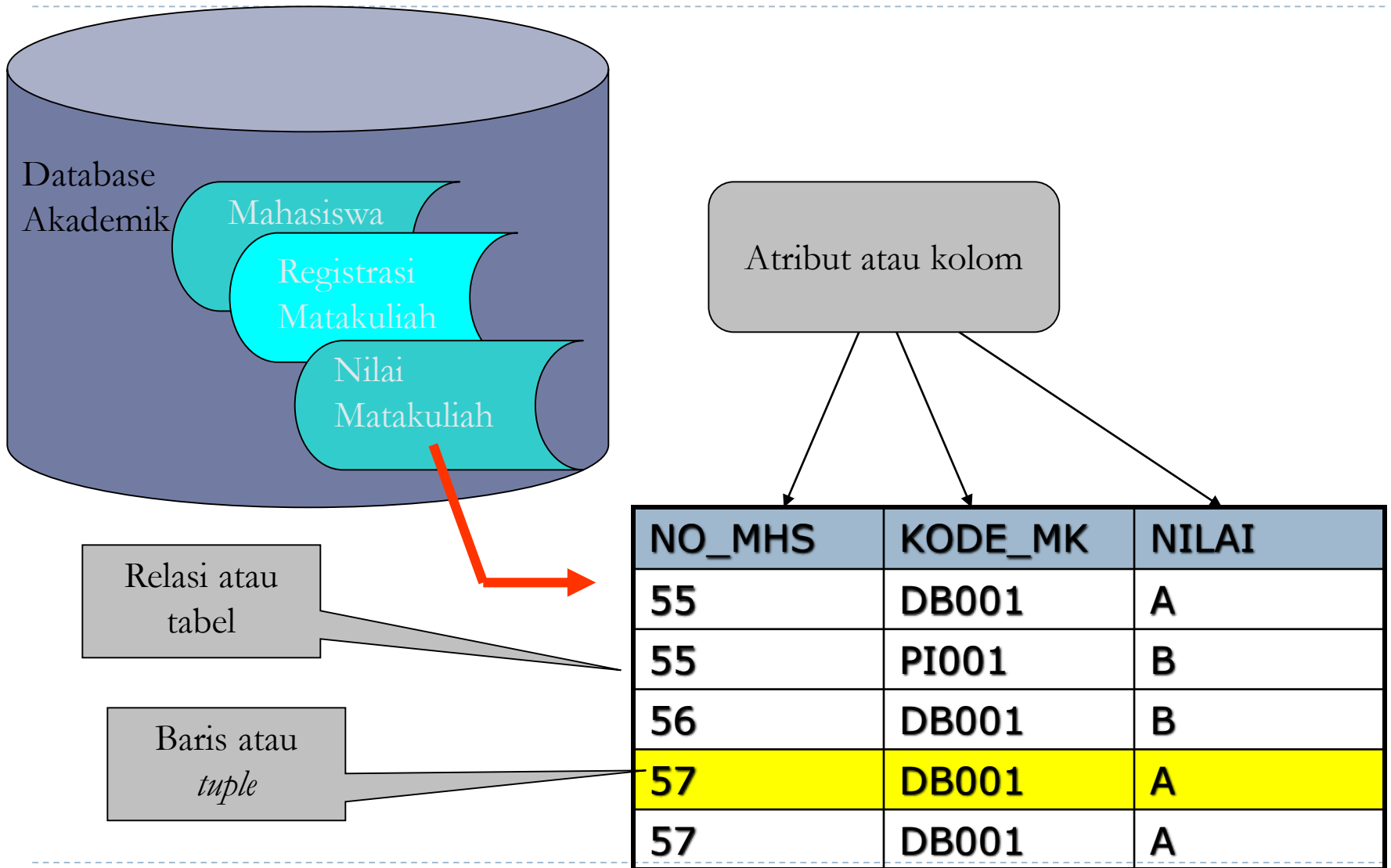
# Istilah Dasar (Lanjutan...)

---

- Data dalam sebuah tabel mengikuti hierarki seperti disamping



# Istilah Dasar (Lanjutan...)





# Perbedaan Istilah

---

ER Model	Relational Model	Database	Traditional Programmer
Entity	Relation	Table	File
Entity Instance	Tuple	Row	Record
Attribute	Attribute	Column	Field
Identifier	Key	Key	Key (or link)



# Karakteristik Dalam Relasi

---


- ▶ Tidak ada baris yang kembar
- ▶ Urutan baris tidak penting
- ▶ Setiap atribut memiliki nama yang unik
- ▶ Letak atribut bebas
- ▶ Setiap atribut memiliki nilai tunggal dan jenisnya sama untuk semua baris



# Pengertian Tidak Ada Baris Kembar

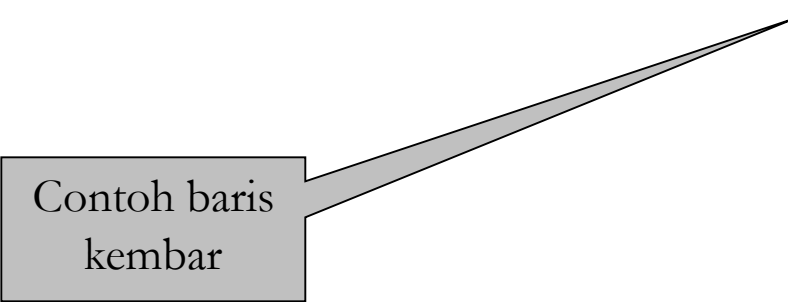
- ▶ Baris punya identitas yang membuat dua buah baris tidak kembar
- ▶ Identitas tersebut dinamakan “primary key”

*Primary key*



NO_MHS	KODE_MK	NILAI
55	DB001	A
55	PI001	B
56	DB001	B
57	DB001	A
57	DB001	B

Contoh baris kembar



# Pengertian Urutan Baris Tidak Penting

NO_MHS	KODE_MK	NILAI
55	DB001	A
55	PI001	B
56	DB001	B
57	DB001	A
57	DB002	A

NO_MHS	KODE_MK	NILAI
57	DB001	A
55	PI001	B
56	DB001	B
55	DB001	A
57	DB002	A



# Setiap Atribut Memiliki Nama Unik

Nama atribut  
unik

NO_SISWA	TES_1	TES_2
12	75	80
13	76	78
14	89	58
15	60	90
16	75	86

TIDAK  
UNIK

NO_SISWA	TES	TES
12	75	80
13	76	78
14	89	58
15	60	90
16	75	86

# Letak Atribut Bebas

NO_SISWA	TES_1	TES_2
12	75	80
13	76	78
14	89	58
15	60	90
16	75	86

TES_2	TES_1	NO_SISWA
80	75	12
78	76	13
58	89	14
90	60	15
86	75	16

Urutan kolom tidak penting

# Setiap Atribut Memiliki Nilai Tunggal

Semua atribut bernilai tunggal

NO_SISWA	TES_1	TES_2
12	75	80
13	76	78
14	89	58
15	60	90
16	75	86

Nilai TIDAK TUNGGAL

NO_SISWA	TES
12	75 80
13	76 78
14	89 58
15	60 90
16	75 86

# Setiap Atribut Berjenis Sama untuk Semua Baris

Semua atribut bernilai tunggal

NO_SISWA	TES_1	TES_2
12	75	80
13	76	78
14	89	58
15	60	90
16	75	86

Jenis tidak sama

NO_SISWA	TES_1	ES_2
12	75	80
13	Tujuh puluh enam	78
14	89	58
15	60	90
16	75	86



# Manfaat Basis Data Relasional

---

1. Bentuknya sederhana
2. Mudah untuk melakukan berbagai operasi data



# Contoh Tabel Keterhubungan

---

**Tabel Mahasiswa**

NIM	NAMA	ALAMAT
H1051021	Dian	Jln. Merdeka
H1051022	Eka	Jln. Imam Bonjol
H1051023	Putra	Jln. Tanjung Sari
H1051024	Iwan	Jln. Sejahtera
H1051025	Rio	Jln. Irian



# RELATIONAL KEY

---

## ▶ **Super key**

Satu atribut atau kumpulan atribut yang secara unik mengidentifikasi sebuah tupel di dalam relasi (satu atau lebih *field* yang dapat dipilih untuk membedakan antara satu *record* dengan *record* lainnya).

Contoh: Untuk tabel Mahasiswa, **Super key**-nya:

- ▶ NIM
- ▶ NAMA (dengan syarat tidak ada nama yang sama)
- ▶ ALAMAT (dengan syarat tidak ada alamat yang sama)
- ▶ NIM + NAMA
- ▶ NIM + ALAMAT
- ▶ NAMA + ALAMAT
- ▶ NIM + NAMA + ALAMAT



## Candidate Key

Kumpulan atribut minimal yang membedakan setiap baris data dalam tabel secara unik.

Untuk bisa menjadi ***candidate key***, suatu atribut harus memenuhi persyaratan sebagai berikut:

- ▶ Untuk satu nilai hanya mengidentifikasikan satu baris dalam satu relasi (unik)
- ▶ Tidak dapat bernilai null
- ▶ ***Super key*** dengan jumlah *field* yang paling sedikit

Maka, ***candidate key*** pada tabel mahasiswa :

NIM, NAMA dan ALAMAT (karena hanya terdiri dari satu *field* saja)



## Primary Key

---

**Candidate key** yang dipilih untuk mengidentifikasi tupel secara unik dalam relasi.

Kunci relasi terdiri dari satu atau lebih atribut-atribut relasi. Agar bisa menjadi sebuah **primary key** sebuah atribut harus memenuhi persyaratan sebagai **candidate key**.

Maka, **primary key** yang dipilih pada tabel mahasiswa adalah NIM (unik, tidak ada NIM yang sama).

- Pemilihan **primary key** dari sejumlah **candidate key** umumnya didasari oleh :
    - **Key** tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan.
    - Jaminan keunikan **key** tersebut lebih baik.
- 



## *Alternate Key*

---

Kunci alternatif dibuat ketika tidak ada satupun atribut dalam sebuah relasi yang bisa mewakili relasi tersebut atau ada yang bisa menjadi ***candidate key*** tetapi tidak cukup efektif untuk digunakan sebagai ***primary key***.

***Candidate key*** yang tidak dipilih sebagai ***primary key*** adalah NAMA dan ALAMAT sebagai ***alternate key***.



## *Foreign Key (FK)*

---

Atribut dengan domain yang sama yang menjadi kunci utama pada sebuah relasi tetapi pada relasi lain atribut tersebut hanya sebagai atribut biasa.

Sebuah FK adalah sekumpulan atribut dalam suatu relasi (misal A) sedemikian sehingga kumpulan atribut ini bukan kunci relasi A tetapi merupakan kunci dari relasi lain.



# Contoh Basis Data Relasional

**Tabel Mahasiswa :**

**Primary key : NoMhs**

**Foreign key : Kode\_Wali**

	Nomhs	Nama	Alamat	Sks	IPK	Kode_wali
▶	151000500	Prima Santoro	Jl.Menu 12 Demangan Yk.	20	2.51	2
	151000501	Priyani	Jl.Demangan Kidul 221 Yk.	18	2.92	2
	151000502	Lufti Yudhianto	Bantengan Lor 34 Yk.	18	3.05	3
	151000503	Sudarno Prawiro	Sagan 76 Yk.	22	2.21	3
	151000504	Yanti Komariah	Jl.Inpres 88 CT XI Yk.	21	3.42	2
	151000505	Sambudi Ratno	Jl.Perkutut 332 Yk.	18	2.02	3

	Kode_wali	Nama	NIP	Pangkat	Jabatan
▶	1	DR. Samtidar	093065531	Lektor	Dosen
	2	Drs.Panca Wahono	094067432	Lektor Kepala	Dosen
	3	Dra.Siti Munawati	094071303	Lektor	Sesjur



# ***Relational Integrity Rules***

---

## **1. Null**

Nilai suatu atribut yang tidak diketahui dan tidak cocok untuk baris (*tuple*) tersebut

## **2. Entity Integrity**

Tidak ada satu komponen *primary key* yang bernilai *null*

## **3. Referential Integrity**

Suatu domain dapat dipakai sebagai kunci primer bila merupakan atribut tunggal pada domain yang bersangkutan



# Bahasa Basis Data

---

*Bahasa Dalam Basis Data  
Relasional*

*Bahasa Query*

*Bahasa Formal*

*Aljabar relasional*

*Kalkulus Relasional*

*Bahasa Komersial*

*QUEL*

*QBE*

*SQL*



# Bahasa Pada Basis Data *Relational*

---

- ▶ Menggunakan bahasa *query* → pernyataan yang diajukan untuk mengambil informasi.
- ▶ Bahasa pada basis data relasional terbagi menjadi dua yaitu :
  1. Bahasa Formal
  2. Bahasa Komersial



# Bahasa Formal

---

- ▶ Bahasa *query* yang diterjemahkan dengan menggunakan simbol-simbol matematis, terbagi menjadi dua yaitu :

- a. **Prosedural**, yaitu pengguna memberi spesifikasi data apa yang dibutuhkan dan bagaimana cara mendapatkannya.

Misal : *Aljabar Relasional*, yaitu dimana *query* diekspresikan dengan cara menerapkan operator tertentu terhadap suatu tabel atau relasi.

- b. **Non Prosedural**, yaitu pengguna menspesifikasikan data apa yang dibutuhkan tanpa menspesifikasikan bagaimana untuk mendapatkannya.

Misal : *Kalkulus Relasional*, dimana *query* menjelaskan set *tuple* yang diinginkan dengan cara menjelaskan predikat *tuple* yang diharapkan, terbagi menjadi dua :

- ▶ Kalkulus Relasional Tupel
  - ▶ Kalkulus Relasional Domain
- 



## Aljabar Relasional

---

- ▶ Kumpulan operasi yang digunakan untuk memanipulasi seluruh relasi.
- ▶ Berdasar teori himpunan : gabungan (***union***), irisan (***intersection***), beda (***difference***) dan hasil kali cartesian (***cartesian product***).
- ▶ Untuk relasi basis data : ***select***, ***project***, ***join*** dan ***division***.



# Bahasa Komersil

---

- ▶ Bahasa *Query* yang dirancang sendiri oleh *programmer* menjadi suatu program aplikasi agar pemakai lebih mudah menggunakannya (*user friendly*).
- ▶ QUEL : Berbasis pada bahasa kalkulus relasional
- ▶ QBE : Berbasis pada bahasa kalkulus relasional
- ▶ SQL : Berbasis pada bahasa kalkulus relasional dan aljabar relasional



# Contoh Basis Data Relasional

---

- ▶ **DB2 → IBM**
- ▶ **ORACLE → Oracle**
- ▶ **SYBASE → Powersoft**
- ▶ **INFORMIX → Informix**
- ▶ **Microsoft Access → Microsoft**



# Model Hirarki atau *Tree*

---

- Biasa juga disebut : **tree structure** (Struktur Pohon “dibalik”), hubungan bertingkat. Dalam model ini elemen-elemen penyusunnya disebut **node**.
  - Dapat berupa rincian data, agregat data dan *record*.
  - Menggunakan pola hubungan orangtua anak.
  - Istilah-istilah yang biasa digunakan dalam Model Hirarki :
    - **Root** : *node* yang memiliki kedudukan paling tinggi dalam hirarki.
    - **Parent** : *node* yang memiliki kedudukan lebih tinggi.
    - **Child** : *node* yang memiliki kedudukan lebih rendah.
    - **Leaves** : *node* yang tidak mempunyai *child*.
- 





# Model Hirarki atau *Tree*

---

Dalam model hirarki ini hanya ada satu **root**, setiap **child** hanya boleh mempunyai 1 **parent** dan **parent** boleh mempunyai > (lebih dari) 1 **child**.

Contoh :

DBMS yang pakai model hirarki : **Information Management System**, dikembangkan oleh IBM dan **Rockwell International Corporation**.

Pada model data hirarki, hubungan antar entitas dinyatakan dalam satu-banyak (**one to many**) atau satu-satu (**one to one**).

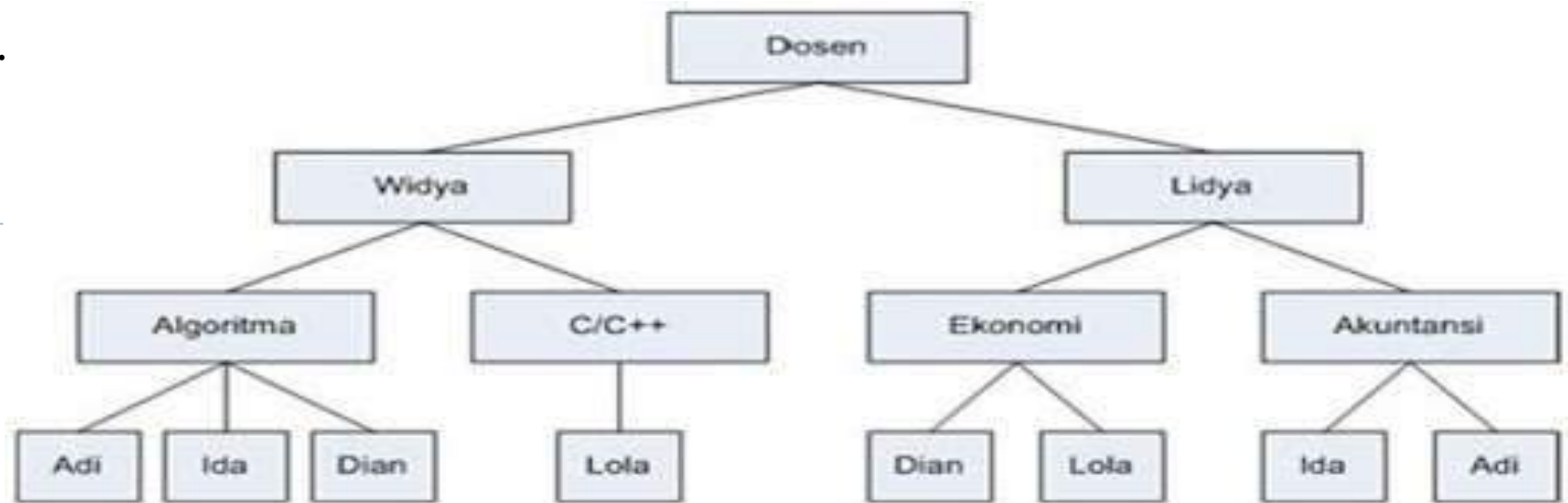
Dalam satu Universitas terdapat banyak Fakultas dan setiap Fakultas terdapat banyak Dosen atau banyak Mahasiswa dan seterusnya.

Tanda panah menunjukkan derajat keterhubungan “banyak”.

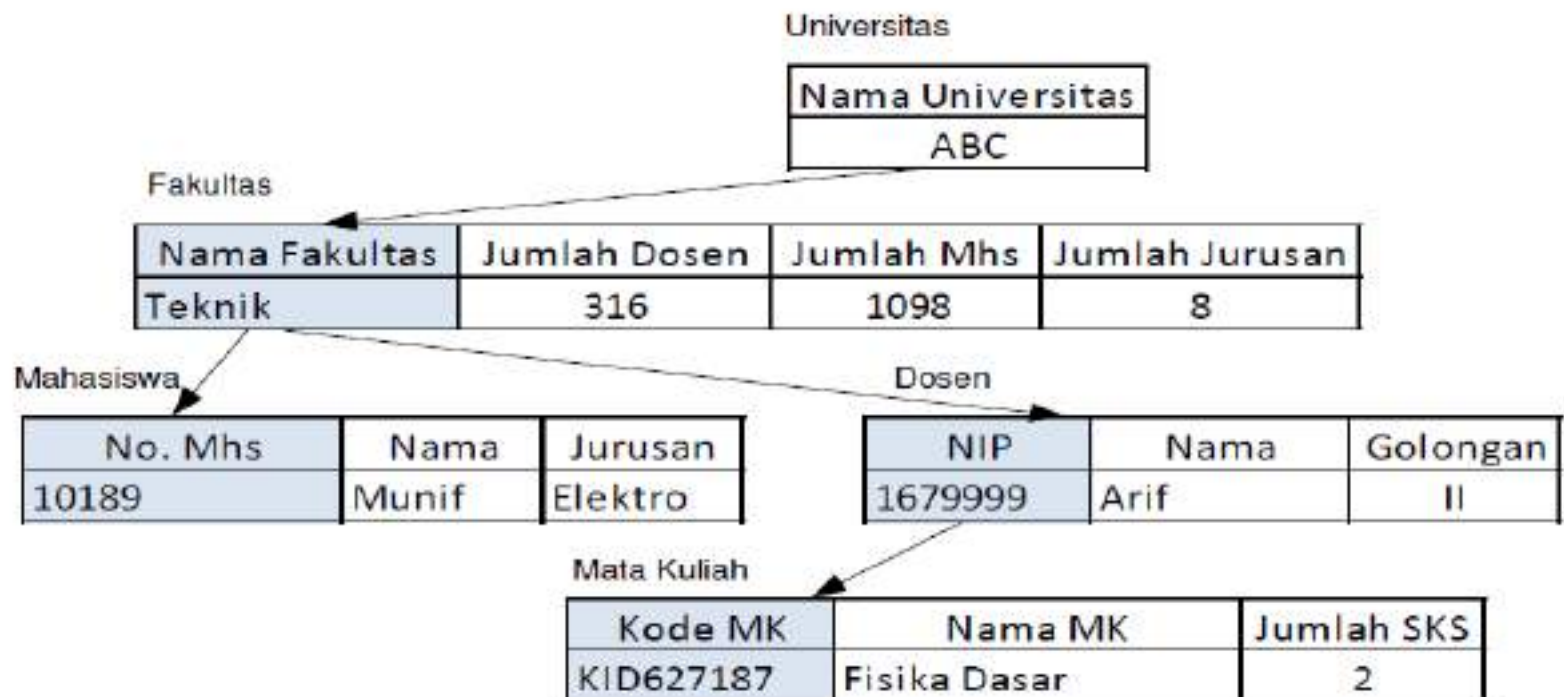
---



A.



B.



# Model Jaringan

Mirip dengan hirarki model, dimana data dan hubungan antar data direpresentasikan dengan **record** dan **links**.

Perbedaannya terletak pada susunan **record** dan **link**nya yaitu **network** model menyusun **record-record** dalam bentuk **graph**.

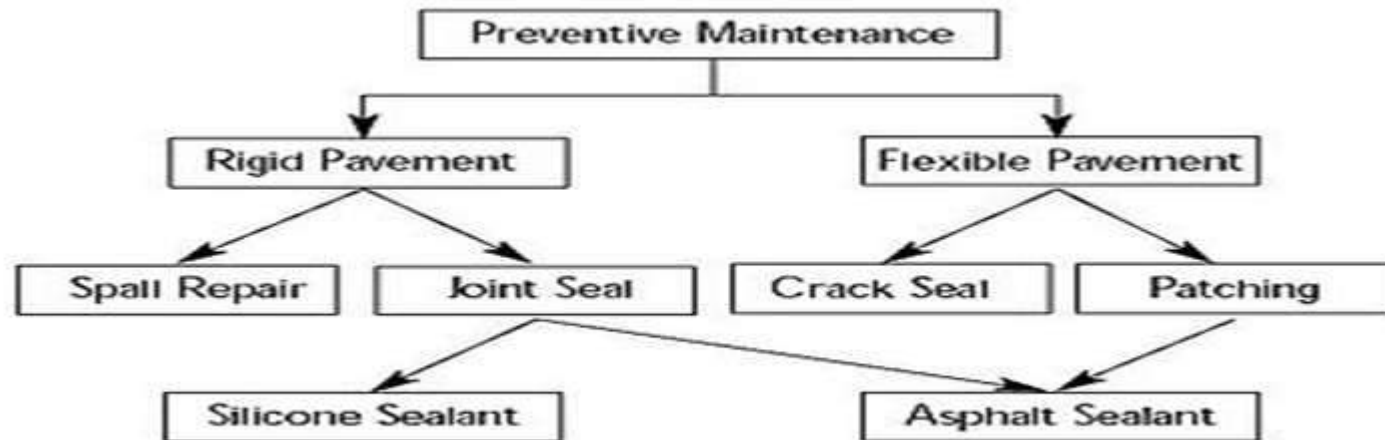
Sebuah **child** dapat mempunyai lebih dari satu **parent**.

Dalam model ini lebih sedikit terdapat data rangkap, namun lebih banyak terdapat hubungan antar entitas, sehingga akan menambah informasi hubungan yang harus disimpan dalam *database*. Hal ini akan menambah volume dan kerumitan dalam penyimpanan berkas data.

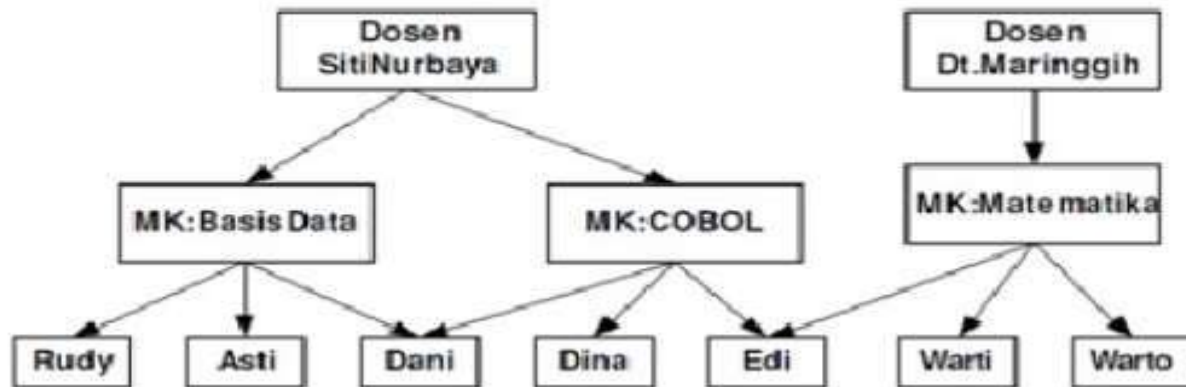
Data dalam model jaringan direpresentasikan dengan sekumpulan **record** (Pascal) dan relasi antara data direpresentasikan oleh **record & link**. **Link** dipandang sebagai **pointer**. **Record-record** diorganisasikan sebagai **graph**.

Contoh : DBMS yang menggunakan model jaringan yaitu CA-IDMS/DB dan **Integrated database management system** yang dibangun oleh **Cullinet Software Inc.**

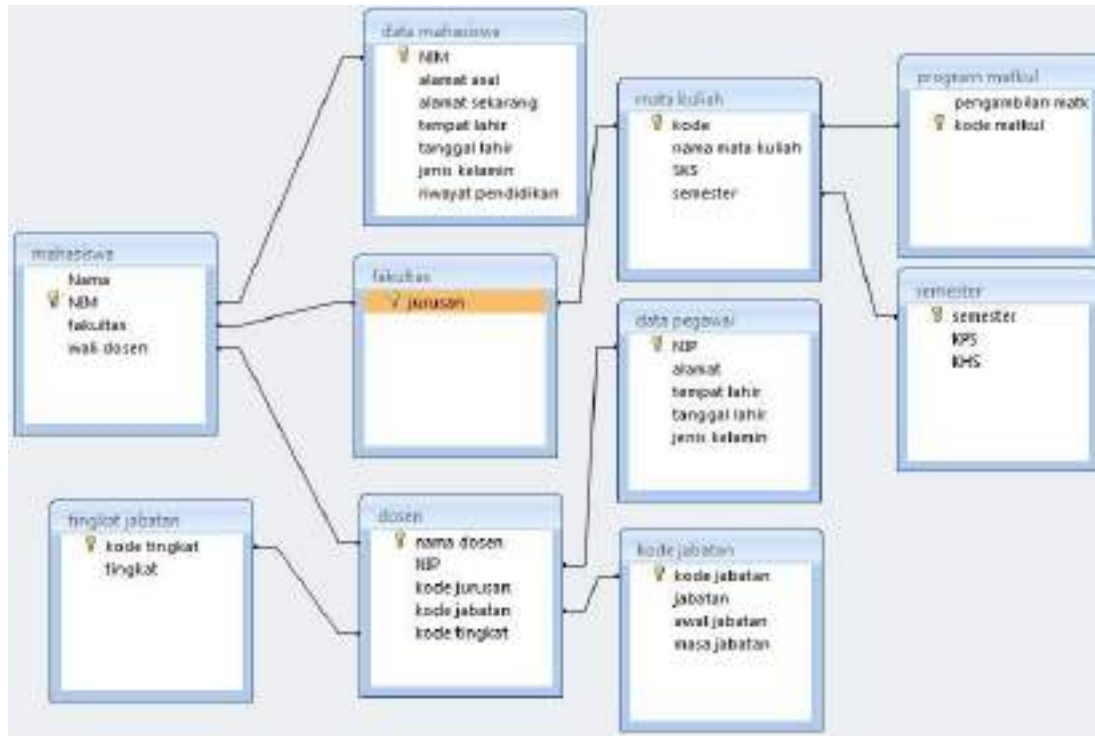
A.



B.



# Model Relasional



Representasi dalam bentuk tabel yang terdapat sejumlah Baris yang menunjuk **record** dan **kolom** yang menunjuk **atribut**.

Model ini banyak digunakan dalam pemodelan dan perancangan Basis Data.

Konsep dan terminologi yang digunakan mirip dengan kondisi nyata yang dihadapi oleh pemakai sehingga mudah dipahami.

# LATIHAN

- ▶ Desainlah tabel untuk penyimpanan informasi pada sebuah toko pakaian *online*, informasi mengenai jenis pakaian yang dijual, informasi pembeli dan informasi pengiriman barang.
- ▶ Buat tabel yang terdiri dari kolom, tipe data, dan keterangan (setiap tabel terdapat minimal 3 baris data).
- ▶ Tentukan key pada setiap tabel dan gambarkan relasi tabel tersebut.
- ▶ Contoh tabel :

No.	Nama Kolom	Tipe Data	Keterangan
1.	Id_MK	Varchar(6)	Primary Key
2.	Nama_MK	Varchar(20)	
3.	SKS	Integer	

Id_MK	Nama_MK	SKS
SK-I01	Basis Data	3
SK-I02	Fisika	3
SK-I03	Multimedia	3

---

TERIMA KASIH

