



BASIS DATA

# NORMALISASI

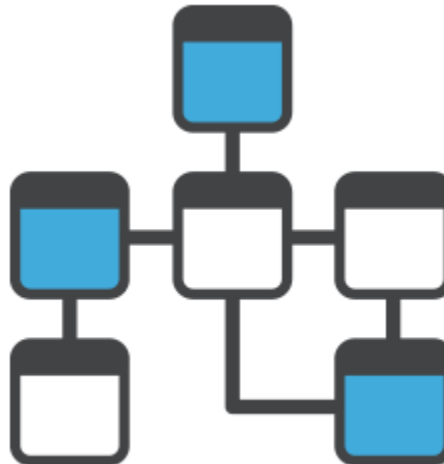
*TIM AJAR BASIS DATA JTI-POLINEMA*



# TUJUAN PERKULIAHAN:

1. Review **normalisasi**, **normalisasi pertama (1NF)**., **normalisasi kedua (2NF)**, **normalisasi ketiga (3NF)**.
2. Memahami aturan **normalisasi Boyce Codd Normal Form (BCNF)**
3. Memahami aturan **normalisasi keempat (4NF)**
4. Memahami aturan **normalisasi kelima (5NF)**

# 1. Review Normalisasi, 1 NF, 2 NF, 3 NF





# NORMALISASI

- **Normalisasi** merupakan sebuah teknik dalam logical desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redudansi).
- **Normalisasi** adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan.



# FUNGSI NORMALISASI

- Normalisasi dilakukan sebagai uji coba pada suatu relasi secara berkelanjutan untuk menentukan apakah relasi sudah baik
- Kondisi relasi yang baik adalah dapat dilakukan proses **insert, update, delete, dan modifikasi** pada satu atau beberapa atribut tanpa mempengaruhi integritas data dalam relasi tersebut
- Dalam perancangan basis data, normalisasi berperan sebagai :
  - Menganalisa skema relasi yang didasarkan pada primary keys dan *functional dependencies* antara atribut-atribut.
  - Satu urutan test, Bila suatu test gagal, maka relasi yang menyalahi test harus **didekomposisi** menjadi sejumlah relasi yang masing-masing memenuhi kaidah normalisasi.

# SUATU RANCANGAN DATABASE DISEBUT BURUK JIKA :



- Data yang sama tersimpan di beberapa tempat (file atau record)
- Ketidakmampuan untuk menghasilkan informasi tertentu
- Terjadi kehilangan informasi
- Terjadi adanya **redudansi (pengulangan)** atau **duplikasi** data sehingga memboroskan ruang penyimpanan dan menyulitkan saat proses updating data



Maka butuh dilakukan **normalisasi**



# TUJUAN NORMALISASI

- Untuk menghilangkan kerangkapan data
- Untuk mengurangi kompleksitas
- Untuk mempermudah pemodifikasian data

# KONSEP KETERGANTUNGAN FUNGSIONAL (FUNCTIONAL DEPENDENCY) (1)



Kondisi **Functional Dependency (FD)** dinyatakan dalam kondisi berikut

$$A \rightarrow B$$

A secara fungsional menentukan B

B secara fungsional tergantung pada A

Syarat **FD** ini terjadi jika minimal dua baris pada suatu tabel dengan nilai A yang sama, memiliki nilai B yang juga sama

$$r_1(A) = r_2(A) , \text{ maka } r_1(B) = r_2(B)$$



# KETERGANTUNGAN FUNGSIONAL (FUNCTIONAL DEPENDENCY) (2)



- Contoh FD pada tabel Nilai

nama_kul	nim	nama_mhs	indeks_nilai
Basisdata	163010015	Betha Susanti	A
Matematika	163010015	Betha Susanti	
Bahasa inggris	163010025	Kyla Nuri M.	B
IMK	163010033	Mega Rinasa	
Matematika	163010033	Mega Rinasa	C
Basisdata	163010035	Tera Akbar	A

## FD

$nim \rightarrow nama\_mhs$   
 $nama\_kul, nim \rightarrow indeks\_nilai$

## Non FD

$nama\_kul \nrightarrow nim$   
 $nim \nrightarrow indeks\_nilai$

- Non FD dapat digunakan untuk membantu mendapatkan FD dari seluruh tabel

# KETERGANTUNGAN FUNGSIONAL (*FUNCTIONAL DEPENDENCY*) (3)



Functional Dependency dari tabel nilai

➤  **$\text{nim} \rightarrow \text{nama\_mhs}$**

Karena untuk setiap nilai nim yang sama, maka nilai nama\_mhs juga sama.

➤  **$\{\text{nama\_kul}, \text{nim}\} \rightarrow \text{indeks\_nilai}$**

Karena atribut indeks\_nilai tergantung pada nama\_kul dan nim secara bersama-sama. Dalam arti lain untuk nama\_kul dan nim yang sama, maka indeks\_nilai juga sama, karena nama\_kul dan nim merupakan key (bersifat unik).

➤  **$\text{nama\_kul} \nrightarrow \text{nim}$**

➤  **$\text{nim} \nrightarrow \text{indeks\_nilai}$**

## BENTUK-BENTUK NORMALISASI

Bentuk Tidak Normal

↓ Menghilangkan perulangan group

Bentuk Normal Pertama (1 NF)

↓ Menghilangkan Ketergantungan parsial

Bentuk Normal Kedua (2NF)

↓ Menghilangkan Ketergantungan Transitif

Bentuk Normal Ketiga (3NF)

↓ Menghilangkan anomali-anomali hasil dari  
ketergantungan fungsional

Bentuk Normal Boyce-Codd (BCNF)

↓ Menghilangkan ketergantungan Multivalue

Bentuk Normal Keempat (4NF)

↓ Menghilangkan anomali-anomali yang tersisa

Bentuk Normal Kelima

# NORMALISASI PERTAMA (1ST NORMAL FORM)



- **Aturan :**

- Tidak adanya atribut multivalued, atribut komposit atau kombinasinya.
- Mendefinisikan atribut kunci.
- Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi)

# LANGKAH PADA NORMALISASI PERTAMA (1ST NORMAL FORM)



1. Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi) → **hilangkan *merge* kolom atau baris jika ada**
2. Mendefinisikan atribut kunci → **definisikan *primary key*** (bisa satu atau kumpulan dari banyak atribut)
3. Tidak adanya atribut multivalue, atribut komposit atau kombinasinya → jika ada **atribut multivalue dekomposisikan menjadi table baru**; dan jika ada **atribut composite maka pecah menjadi atribut yang berbeda** → tentukan **primary key** dan **foreign key** dari table yang baru

# CONTOH 1 (ATRIBUT MULTI-VAL)



Misal data mahasiswa sbb:

Nim	Nama	Hobi
12020001	Heri Susanto	Sepak bola, membaca komik, berenang
12020013	Siti Zulaiha	Memasak, program komputer
12020015	Dini Susanti	Menjahit, membuat roti

Atau

Nim	Nama	Hobi 1	Hobi 2	Hobi 3
12020001	Heri Susanto	Sepak bola	membaca komik	berenang
12020013	Siti Zulaiha	Memasak	Program komputer	
12020015	Dini Susanti	Menjahit	membuat	

Tabel-tabel di atas tidak memenuhi syarat 1NF

# CONTOH 1 (ATRIBUT MULTI-VALUE) 1NF



Tentukan primary key berdasarkan ketergantungan fungsional :

<u>Nim</u>	Nama	Hobi
12020001	Heri Susanto	Sepak bola, membaca komik, berenang
12020013	Siti Zulaiha	Memasak, program komputer
12020015	Dini Susanti	Menjahit, membuat roti

Atau

<u>Nim</u>	Nama	Hobi 1	Hobi 2	Hobi 3
12020001	Heri Susanto	Sepak bola	membaca komik	berenang
12020013	Siti Zulaiha	Memasak	Program komputer	
12020015	Dini Susanti	Menjahit	membuat	

# CONTOH 1 (SAMB...)

## 1NF



Karena terdapat multivalued di atribut hobi maka didekomposisi :

### ➤ Tabel Mahasiswa

<u>Nrp</u>	<u>Nama</u>
12020001	Heri Susanto
12020013	Siti Zulaiha
12020015	Dini Susanti

### ➤ Tabel Hobi

<u>Nrp</u>	<u>Hobi</u>
12020001	Sepakbola
12020001	membaca komik
12020001	Berenang
12020013	Memasak
12020013	mrogram komputer
12020015	Menjahit
12020015	membuat roti



Memenuhi  
bentuk 1NF



# CONTOH 2 (COMPOSITE)



## JadwaKuliah

Kodekul	NamaKul	Dosen	Kelas	Jadwal
---------	---------	-------	-------	--------

- Dimana nilai pada atribut jadwal berisi gabungan antara Hari dan Jam.
- Jika asumsi hari dan jam memegang peranan penting dalam sistem basis data, maka atribut Jadwal perlu dipisah sehingga menjadi JadwalHari dan JadwalJam sbb:



Memenuhi 1  
NF

## JadwaKuliah

<u>Kodekul</u>	NamaKul	Dosen	Kelas	JadwalHari	JadwalJam
----------------	---------	-------	-------	------------	-----------

# NORMALISASI KEDUA (2ND NORMAL FORM)



- **Aturan :**

- Sudah memenuhi dalam bentuk normal kesatu (1NF)
- Semua atribut bukan key primer hanya boleh memiliki ketergantungan (functional dependency) pada atribut key primer
- Jika ada **ketergantungan parsial** maka atribut tersebut harus dipisah pada tabel yang lain

- **\*Ket. Tambahan :**

Ketergantungan parsial ---> hanya tergantung pada sebagian key primer

# CONTOH 2NF



Dengan menggunakan acuan tabel universal pada slide sebelumnya,

Tabel berikut memenuhi 1NF tapi tidak termasuk 2NF:

nim	nama_mhs	alamat_mhs	kode-kul	Nama_kul	sks	Indeks_nilai
-----	----------	------------	----------	----------	-----	--------------

- Tidak memenuhi 2NF, karena {nim, kode\_kul} yang dianggap sebagai key primer sedangkan:

{nim, kode_kul}	→	mhs_nama
{nim, kode_kul}	→	mhs_alamat
{nim, kode_kul}	→	mk_nama
{nim, kode_kul}	→	mk_sks
{nim, kode_kul}	→	indeks_nilai

- Tabel di atas perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF



## Contoh Functional Dependency 2NF adalah :

{nim, kode_kul}	→	indeks_nilai	(FD1)
nim	→	{nama_mhs, alamat_mhs}	(FD2)
kode_kul	→	{nama_kul, sks}	(FD3)

## Maka Dekomposisi tabel pada contoh 2NF adalah :

FD1	( <u>nim</u> , <u>kode_kul</u> , indeks_nilai)	→ Tabel nilai
FD2	( <u>nim</u> , nama_mhs, alamat_mhs)	→ Tabel mahasiswa
FD3	( <u>kode_kul</u> , nama_kul, sks)	→ Tabel mataKuliah

**Jangan lupa untuk mendefinisikan foreign key**

# NORMALISASI KETIGA (3RD NORMAL FORM)



- **Aturan :**
  - Sudah berada dalam bentuk normal kedua (2NF)
  - Tidak ada ketergantungan transitif (dimana atribut bukan key primer tergantung pada atribut bukan key primer lainnya).
- **Dalam bentuk FD :**
  - $X \rightarrow Y$  maka **X** adalah **super key**
  - $X \rightarrow A$ , dengan membolehkan A sebagai bagian dari primary key



# CONTOH 3NF

Berdasarkan informasi pada tabel universal pada slide sebelumnya, jika pada tabel mahasiswa, atribut alamat sesuai dengan kebutuhan dibagi informasi jalan, kota, provinsi, dan kodepos, maka tabel mahasiswa dapat dituliskan sebagai berikut :

nim	nama_mhs	alamat_jalan_mhs	alamat_kota_mhs	alamat_provinsi_mhs	alamat_kodepos_mhs
-----	----------	------------------	-----------------	---------------------	--------------------

- Tabel tersebut sudah memenuhi 2NF, tapi tidak memenuhi 3NF:
- karena masih terdapat atribut non primary key (yakni **alamat\_kota\_mhs** dan **alamat\_provinsi\_mhs**) yang memiliki ketergantungan terhadap atribut non primary key yang lain (yakni **alamat\_kodepos\_mhs**):

**Sehingga tabel tersebut perlu didekomposisi menjadi:**

mahasiswa ( <u>nim</u> , nama_mhs, alamat_jalan_mhs, alamat_kodepos_mhs)
kodepos (alamat_kodepos_mhs, alamat_jalan_mhs, alamat_provinsi_mhs, alamat_kota_mhs)



# PENJELASAN CONTOH 3NF

- Pada kodepos (alamat\_kodepos\_mhs, alamat\_jalan\_mhs, alamat\_provinsi\_mhs, alamat\_kota\_mhs) terdapat FD
  - **alamat\_jalan\_mhs, alamat\_provinsi\_mhs, alamat\_kota\_mhs** → alamat\_kodepos\_mhs
  - alamat\_kodepos\_mhs → **alamat\_provinsi\_mhs, alamat\_kota\_mhs**

NB :

**Merah : Superkey**

**Hitam : bukan superkey**

**Biru : Bagian Primary Key**

# CONTOH



Tabel berikut memenuhi 2NF, tapi tidak memenuhi 3NF:

## Mahasiswa

Nrp	Nama	Alm_Jalan	Alm_Kota	Alm_Provinsi	Alm_Kodepos
-----	------	-----------	----------	--------------	-------------

- karena masih terdapat atribut *non primary key* (yakni **alm\_kota** dan **alm\_Provinsi**) yang memiliki ketergantungan terhadap atribut *non primary key* yang lain (yakni **alm\_kodepos**):

alm\_kodepos → {alm\_Provinsi, alm\_kota}

- Sehingga tabel tersebut perlu didekomposisi menjadi:

Mahasiswa (Nrp, nama, alm\_jalan, alm\_kodepos)

Kodepos (alm\_kodepos, alm\_provinsi, alm\_kota)



# INFORMASI



Tabel-tabel yang memenuhi kriteria normalisasi ketiga, sudah siap diimplementasikan. Sebenarnya masih ada lagi bentuk normalisasi yang lain; BCNF, 4NF, 5NF, hanya saja jarang dipakai. **Pada kebanyakan kasus, normalisasi hanya sampai 3NF.**

# PEMBAHASAN LATIHAN 1 : NORMALISASI DATA



NoProyek	NamaProyek	NoPegawai	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
		Peg02	Paula	B	900.000
		Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
		Peg12	Sita	B	900.000
		Peg14	Yusni	B	900.000

Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama.

Field-field tabel di atas yang merupakan group berulang :  
NoPegawai, NamaPegawai, Golongan, BesarGaji.

# NORMALISASI PERTAMA



Solusinya hilangkan duplikasi dengan mencari ketergantungan parsial. menjadikan field-field menjadi tergantung pada satu atau beberapa field. Karena yang dapat dijadikan kunci adalah *NoProyek* dan *NoPegawai*, maka langkah kemudian dicari field-field mana yang tergantung pada *NoProyek* dan mana yang tergantung pada *NoPegawai*.

<u>NoProyek</u>	NamaProyek	<u>NoPegawai</u>	NamaPegawai	Golongan	BesarGaji
NP001	BRR	Peg01	Anton	A	1.000.000
NP001	BRR	Peg02	Paula	B	900.000
NP001	BRR	Peg06	Koko	C	750.000
NP002	PEMDA	Peg01	Anton	A	1.000.000
NP002	PEMDA	Peg12	Sita	B	900.000
NP002	PEMDA	Peg14	Yusni	B	900.000

Primary Key

# NORMALISASI KEDUA



- Field-field yang tergantung pada satu field haruslah dipisah dengan tepat, misalnya *NoProyek* menjelaskan *NamaProyek* dan *NoPegawai* menjelaskan *NamaPegawai*, *Golongan* dan *BesarGaji*.

# NORMALISASI KEDUA



## TABEL PROYEK

<u><i>Noprojek</i></u>	NamaProyek
NP001	BRR
NP002	PEMDA

## TABEL PEGAWAI

<u><i>Nopegawai</i></u>	NamaPegawai	Golongan	BesarGaji
Peg01	Anton	A	1.000.000
Peg02	Paula	B	900.000
Peg06	<u>Koko</u>	C	750.000
Peg12	<u>Sita</u>	B	900.000 )
Peg14	<u>Yusni</u>	B	900.000



Untuk membuat hubungan antara dua tabel, dibuat suatu tabel yang berisi key-key dari tabel yang lain.

## TABEL PROYEKPEGAWAI

<u><i>No proyek</i></u>	<u><i>No Pegawai</i></u>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

Kolom Noprojek dan NoPegawai merupakan foreign key ke table proyek dan pegawai

# NORMALISASI KETIGA



Pada tabel diatas masih terdapat masalah, bahwa *BesarGaji* tergantung kepada *Golongan* nya. Padahal disini *Golongan* bukan merupakan field kunci.

Artinya kita harus memisahkan field non-kunci *Golongan* dan *BesarGaji* yang tadinya tergantung secara parsial kepada field kunci *NoPegawai*, untuk menghilangkan ketergantungan transitif.

## TABEL PROYEKPEGAWAI



### TABEL PROYEK

<u>Noprojek</u>	NamaProyek
NP001	BRR
NP002	PEMDA

<u>Noprojek</u>	<u>NoPegawai</u>
NP001	Peg01
NP001	Peg02
NP001	Peg06
NP002	Peg01
NP002	Peg12
NP002	Peg14

### TABEL PEGAWAI

<u>Nopegawai</u>	NamaPegawai	Golongan
Peg01	Anton	A
Peg02	Paula	B
Peg06	Koko	C
Peg12	Sita	B
Peg14	Yusni	B

### TABEL GOLONGAN

<u>Golongan</u>	BesarGaji
A	1.000.000
B	900.000
C	750.000



# PEMBAHASAN LATIHAN 2 : NORMALISASI DATA



No-Mhs	Nm-Mhs	Jurusan	Kd-MK	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350 MI465	Manajemen Basis Data Analisis Prc. Sistem	B104 B317	Ati Dita	A B
5432	Bakri	AK	MI350 AKN201 MKT300	Manajemen Basis Data Akuntansi Keuangan Dasar Pemasaran	B104 D310 B212	Ati Lia Lola	C B A

# 1NF



<u>No-Mhs</u>	Nm-Mhs	Jurusan	<u>Kd-MK</u>	Nama-MK	Kd-Dosen	Nm_Dosen	Nilai
2683	Welli	MI	MI350	Manajemen Basis Data	B104	Ati	A
2683	Welli	MI	MI465	Analisis Prc. Sistem	B317	Dita	B
5432	Bakri	AK	MI350	Manajemen Basis Data	B104	Ati	C
5432	Bakri	AK	AKN201	Akuntansi Keuangan	D310	Lia	B
5432	Bakri	AK	MKT300	Dasar Pemasaran	B212	Lola	A



2NF

<u>No-Mhs</u>	Nama-Mhs	Jurusan
---------------	----------	---------

2683	Welli	MI
5432	Bakri	AK

**Tabel Mahasiswa**

<u>Kode-MK</u>	Nama-MK	Kode-Dosen	Nama-Dosen
----------------	---------	------------	------------

MI350	Manajemen Basis Data	B104	Ati
MI465	Analisis Prc. Sistem	B317	Dita
AKN201	Akuntansi Keuangan	D310	Lia
MKT300	Dasar Pemasaran	B212	Lola

**Tabel Mata Kuliah**

<u>No-Mhs</u>	<u>Kode MK</u>	Nilai
---------------	----------------	-------

2683	MI350	A
2683	MI465	B
5432	MI350	C
5432	AKN201	B
5432	MKT300	A

**Tabel Nilai**



2NF

**Tabel Mahasiswa**

<u>No-Mhs</u>	Nama-Mhs	Jurusan
2683	Welli	MI
5432	Bakri	AK

**Tabel Dosen**

<u>Kode-Dosen</u>	Nama-Dosen
B104	Ati
B317	Dita
D310	Lia
B212	Lola

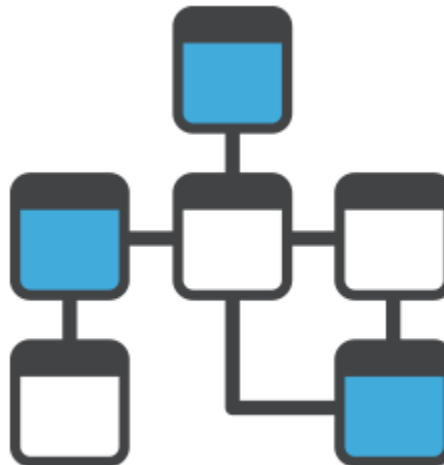
**Tabel Mata Kuliah**

<u>Kode-MK</u>	Nama-MK	Kode-Dosen
MI350	Manajemen Basis Data	B104
MI465	Analisis Prc. Sistem	B317
AKN201	Akuntansi Keuangan	D310
MKT300	Dasar Pemasaran	B212

**Tabel Nilai**

<u>No-Mhs</u>	<u>Kode MK</u>	Nilai
2683	MI350	A
2683	MI465	B
5432	MI350	C
5432	AKN201	B
5432	MKT300	A

## 2. NORMALISASI BOYCE CODD NORMAL FORM (BCNF)



# BOYCE CODD NORMAL FORM (BCNF)



- Suatu relasi disebut memenuhi BCNF jika dan hanya jika setiap determinan yang ada pada relasi tersebut adalah candidate key.  
**Definisi yang lain :**  
Suatu relasi disebut memenuhi BCNF jika untuk setiap FD nontrivial :  $X \rightarrow A$  atribut X adalah superkey.
- Untuk normalisasi ke bentuk BCNF, maka tabel 3NF didekomposisi menjadi beberapa tabel yang masing-masing memenuhi BCNF.
- Tujuan membentuk BCNF :
  - :: semantik multiple candidate key menjadi lebih eksplisit (FD hanya pada candidate key).
  - :: menghindari update anomali yang masih mungkin terjadi pada 3NF.

**Dari definisi 3NF dan BCNF, maka apabila suatu relasi memenuhi BCNF pasti memenuhi 3NF, tetapi belum tentu sebaliknya.**

# CONTOH BCNF

## Contoh :

Diketahui tabel  $R=(A,B,C)$

dengan FD :  $A \rightarrow B$  dan  $B \rightarrow C$  maka  $R$  bukan BCNF, sebab :

- $A$  superkey ?

$A \rightarrow B$  (diketahui)

$A \rightarrow B$  dan  $B \rightarrow C$  maka  $A \rightarrow C$  (transitif)

$A \rightarrow A$  (refleksif)

Sehingga  $A \rightarrow (A,B,C)$  atau  $A \rightarrow R$ . Jadi  $A$  superkey.

- $B$  superkey ?

$B \rightarrow C$  (diketahui)

$B \rightarrow B$  (refleksif)

Tapi  $B \nrightarrow A$ . Sehingga  $B \nrightarrow A,B,C$  atau  $B$  bukan superkey.

Agar  $R$  memenuhi BCNF maka didekomposisi menjadi :

$R_1=(A,B)$  ; FD :  $A \rightarrow B$  dan

$R_2=(B,C)$  ; FD :  $B \rightarrow C$ .

sehingga  $R_1$  dan  $R_2$  masing-masing memenuhi BCNF. Sebab  $A$  dan  $B$  dua-duanya sekarang menjadi superkey.

### Contoh :

Diketahui tabel  $R=(A,B,C)$   
dengan FD :  $AB \rightarrow C$  dan  $C \rightarrow B$ . Apakah :

- 3NF ?
- BCNF ?
  
- R memenuhi 3NF karena :  
 $AB \rightarrow C$  ; maka  $AB \rightarrow ABC$ , atau  $A \rightarrow R$ . Jadi AB superkey dari R  
 $C \rightarrow B$  ; maka  $AC \rightarrow AB$ , atau  $AC \rightarrow ABC$  dan  $AC \rightarrow R$ .  
Jadi AC juga superkey (sekaligus juga candidate key) dari R  
Karena AB superkey dan C subset candidate key maka  
R memenuhi 3NF
- R bukan BCNF karena :  
AB superkey tetapi C bukan superkey.

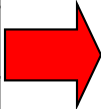


# CONTOH KASUS BCNF

- Contoh kasus redundansi pada 3NF  
Jadwal = (Nim, Modul, Dosen)  
FD = {Dosen  $\rightarrow$  Modul}  
Relasi ini memenuhi 3NF, karena tidak ada ketergantungan transitif.  
Tetapi tidak memenuhi BCNF karena dari Dosen  $\rightarrow$  Modul maka Dosen bukan candidate key.  
Alternatif yang dilakukan adalah dekomposisi tabel menjadi :

<u>NIM</u>	<u>Modul</u>	Dosen
P11.2004.0129	C#	Amin
P11.2004.0130	Basdat	Aris
P11.2004.0129	C#	Amin
P11.2004.0201	C#	Budi
P11.2004.0250	Basdat	Jono
P11.2004.0260	C#	Budi

**NOT BCNF**

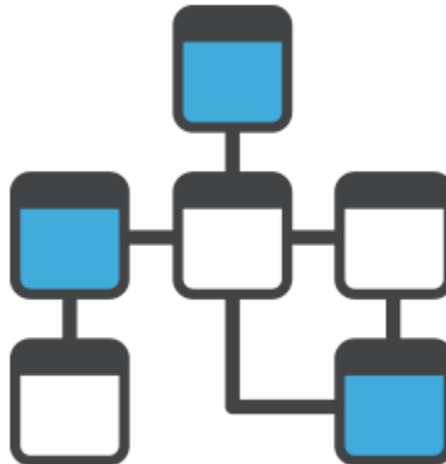


<u>NIM</u>	Dosen
P11.2004.0129	Amin
P11.2004.0130	Aris
P11.2004.0129	Amin
P11.2004.0201	Budi
P11.2004.0250	Jono
P11.2004.0260	Budi

**BCNF**

Dosen	<u>Modul</u>
Amin	C#
Aris	Basdat
Jono	Basdat
Budi	C#

### 3. NORMALISASI KEEMPAT (4NF)



# BENTUK NORMAL TAHAP KEEMPAT (4NF)



- Bentuk normal 4NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk BCNF, dan tabel tersebut tidak boleh memiliki lebih dari sebuah *multivalued attribute*
- Untuk setiap *multivalued dependencies* (MVD) juga harus merupakan *functional dependencies*
- Dependensi nilai banyak (multivalued dependency) diperkenalkan oleh R. Fagin pada tahun 1977, dipakai pada bentuk normal keempat (4NF). Dependensi ini menyatakan hubungan satu ke banyak. Misal : Ahmad adalah teknisi, perenang juga penembak.
- Secara umum dependensi nilai banyak muncul pada relasi yang sedikitnya memiliki tiga atribut dan dua diantaranya bernilai banyak. Dua atribut bernilai banyak ini tergantung pada atribut bernilai tunggal. Misal:



MATAKULIAH	DOSEN	ISI
Pengetahuan Komputer	Amir Fitria	Dasar Komputer Pengenaln Pengolah kata Sejarah komputer
Matematika I	Fitria	Diferensial Integral

Akan salah jika dinormalisasikan sebagai berikut

MATAKULIAH	DOSEN	ISI
Pengetahuan Komputer	Amir	Dasar Komputer
Pengetahuan Komputer	Amir	Pengenaln Pengolah kata
Pengetahuan Komputer	Amir	Sejarah komputer
Pengetahuan Komputer	Fitria	Dasar Komputer
Pengetahuan Komputer	Fitria	Pengenaln Pengolah kata
Pengetahuan Komputer	Fitria	Sejarah komputer
Matematika I	Fitria	Diferensial
Matematika I	Fitria	Integral

Karena Atribut Dosen dan Isi tidak ada ketergantungan



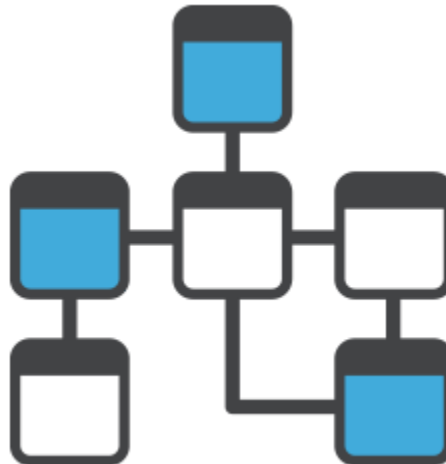
- Dependensi dalam atribut bernilai banyak ini disebut multidependen Misal :  
Bila suatu relasi R dengan atribut A, B, C, maka atribut B dikatakan multidependen terhadap A jika masing-masing nilai B hanya bergantung pada A saja tak tergantung pada P.
- Atau dinyatakan dengan :  
 $A \twoheadrightarrow B$   
(dibaca A menentukan banyak nilai B atau B multidependen terhadap A)
- Teorema Fagin yang berkaitan dengan dependensi nilai banyak:  
Bila  $R(A,B,C)$  merupakan suatu relasi, dengan A,B,C adalah atribut-atribut relasi tersebut, maka proyeksi dari R berupa (A,B) dan (A,C) jika R memenuhi MVD  
 $A \twoheadrightarrow B \mid C$
- Perlu diketahui jika  
 $A \twoheadrightarrow B$   
 $A \twoheadrightarrow C$   
Maka keduanya dapat ditulis  
 $A \twoheadrightarrow B \mid C$

- Berdasarkan terorema Fagin, relasi di atas dapat didekomposisikan menjadi dua relasi

MATAKULIAH	DOSEN
Pengetahuan Komputer	Amir
Pengetahuan Komputer	Fitria
Matematika I	Fitria

MATAKULIAH	ISI
Pengetahuan Komputer	Dasar Komputer
Pengetahuan Komputer	Pengenalan Pengolahan Kata
Pengetahuan Komputer	Pengenalan Lembar Kerja
Matematika I	Differensial
Matematika I	Integral

## 4. NORMALISASI KELIMA (5NF)



# BENTUK NORMAL TAHAP KELIMA (5NF)



- Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yg lebih kecil.
- Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency*, 5NF dibentuk berdasarkan konsep *join dependence*. Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil, harus bisa digabungkan lagi (join) untuk membentuk tabel semula



# DEPENDENSI GABUNGAN DAN BENTUK NORMAL KELIMA (5NF)



- Dependensi gabungan mendasari bentuk normal kelima
- Suatu relasi  $R(X,Y,Z)$  memenuhi dependensi gabungan bila dimungkinkan dibuat gabungan dari proyeksi  $A,B,C$  dengan  $A, B, C$  merupakan sub himpunan dari atribut-atribut  $R$ .

Ditulis :

$*(A, B, C)$

dengan  $A = (X,Y), B = (Y,Z), C = (Z,X)$

# CONTOH STUDI KASUS

CONTOH TERDAPAT HUBUNGAN DEALER YANG MELAYANI SUATU PERUSAHAAN DISTRIBUTOR KENDARAAN

DEALER	DISTRIBUTOR	KENDARAAN
PT Sumber Jaya	Nissan	Truk Nissan
PT Sumber Jaya	Toyota	Toyota Kijang
PT Sumber Jaya	Toyota	Truk Dyna
Pt Asterindo	Nissan	Sedan Nissan

Relasi di atas memenuhi dependensi gabungan \*(DEALER DISTRIBUTOR, DISTRIBUTOR KENDARAAN, DEALER KENDARAAN). Oleh karena itu relasi ini dapat didekomposisikan menjadi tiga relasi :

- DEAL\_DIST(DEALER, DISTRIBUTOR)
- DIST\_KEND(DISTRIBUTOR, KENDARAAN)
- DEAL\_KEND(DEALER, KENDARAAN)

- Gabungan dari dua proyeksi di atas bisa jadi menghasilkan relasi yang mengandung baris yang salah. Namun gabungan ketiga proyeksi akan menghasilkan relasi yang sesuai dengan aslinya

DEALER	DISTRIBUTOR
PT Sumber Jaya	Nissan
PT Sumber Jaya	Toyota
PT Sumber Jaya	Toyota
Pt Asterindo	Nissan

DISTRIBU	KENDARAAN
Nissan	Truk Nissan
Toyota	Toyota Kijang
Toyota	Truk Dyna
Nissan	Sedan Nissan

DEALER	DISTRIBUTOR	KENDARAAN
PT Sumber Jaya	Nissan	Truk Nissan
PT Sumber Jaya	Toyota	Toyota Kijang
PT Sumber Jaya	Toyota	Truk Dyna
Pt Asterindo	Nissan	Truk Nissan
Pt Asterindo	Nissan	Sedan Nissan



- Bentuk Normal kelima (5NF) disebut juga PJ/NF (Projection Join/Normal Forma) menggunakan acuan dependensi gabungan. Suatu relasi berada dalam 5NF jika dan hanya jika setiap dependensi gabungan dalam R tersirat oleh kunci kandidat relasi R.
- Secara sederhana suatu relasi berada pada 5NF jika tidak dapat lagi didekomposisikan menjadi relasi-relasi yang lebih kecil yang memiliki kunci kandidat yang tidak sama dengan kunci kandidat relasi besarnya



# LATIHAN SOAL 1

Diketahui tabel berikut, buatlah langkah-langkah normalisasi, sehingga tabel menjadi normal !

NIP	Nama	Jabatan	Keahlian	Lama Kerja
156107	Rangga	Analisis Senior	Java	6
			Oracle	1
156109	Maura	Analisis Junior	Java	2
			C++	2
156120	Cinta	Programmer	Dbase	3
			Sybase	1
			Java	1



## LATIHAN SOAL 2

- Diketahui tabel berikut yang menyimpan data pada suatu rental film. Buatlah langkah-langkah normalisasi, sehingga tabel menjadi normal !

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3 <sup>rd</sup> Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5 <sup>th</sup> Avenue	Clash of the Titans	Mr.



**TERIMAKASIH**



# REFERENSI

- Dwi Puspitasari, S.Kom, “**Buku Ajar Dasar Basis Data**”, *Program Studi Manajemen Informatika Politeknik Negeri Malang*, 2012.
- Fathansyah, “**Basisdata Revisi Kedua**”, Bandung: Informatika, 2015.