

**BASIS DATA**

**NORMALISASI**

# NORMALISASI

- **Normalisasi :**

Teknik/pendekatan yang digunakan dalam membangun disain logik database relasional melalui organisasi himpunan data dengan tingkat ketergantungan fungsional dan keterkaitan yang tinggi sedemikian sehingga menghasilkan struktur tabel yang normal.

- **Tujuan :**

- ✓ Minimalisasi redundansi (pengulangan data)
- ✓ Memudahkan identifikasi entitas
- ✓ Mencegah terjadinya anomali

- **Beberapa bentuk normal (normal forms, NF) :**

- ✓ 1NF, 2NF, 3NF, BCNF  
based on keys and functional dependencies
- ✓ 4NF, 5NF  
based on keys and multi-valued dependencies)

# FIRST NORMAL FORM (1NF)

- Suatu relasi disebut memenuhi bentuk normal pertama (1NF) jika dan hanya jika setiap atribut dari relasi tersebut hanya memiliki nilai tunggal dan tidak ada pengulangan grup atribut dalam baris.
- Bentuk 1NF tidak boleh mengandung grup atribut yang berulang.
- Tujuan membentuk 1NF :
  - ✓ semantik tabel menjadi lebih eksplisit (say anything once).
  - ✓ semua operator aljabar relasional dapat diaplikasikan pada tabel.

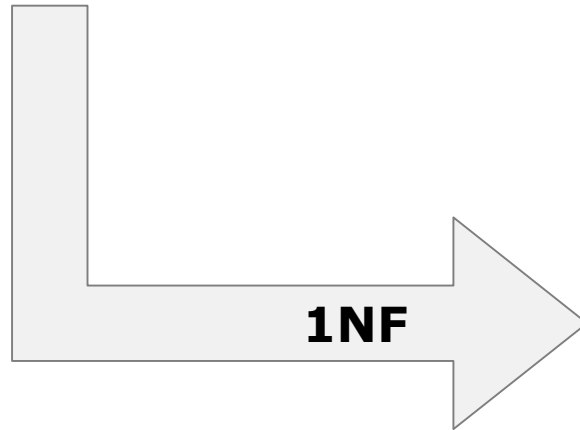
# FIRST NORMAL FORM (1NF)

Tabel: Sales

<b>IDSales</b>	<b>NamaSales</b>	<b>Telepon</b>
ADN006	Yeni, SE	3517261, 3520165
ADN007	Memey	4744621,0812286142
ADN008	Tina	08566241521
ADN009	Ir. Yanto	7265122, 7123910
ADN010	Made	6723192

**non-atomic**

**Unnormalized  
Not 1NF**



<b>IDSales</b>	<b>NamaSales</b>	<b>Telepon</b>
ADN006	Yeni, SE	3517261
ADN006	Yeni, SE	3520165
ADN007	Memey	4744621
ADN007	Memey	08122861427
ADN008	Tina	08566241521
ADN009	Ir. Yanto	7265122
ADN009	Ir. Yanto	7123910
ADN010	Made	6723192

## contoh kasus proses normalisasi

TeraSoft  
Jl. Gajah 29  
Palembang

No. Faktur: 035

30 Juni 2021

Kepada  
Fulan  
Jl. ABC 7

No. HP 0812345

Jumlah Barang	Satuan	Kode Barang	Nama Barang	Harga Satuan	Jumlah Harga
10	keping	DVD17	DVD blank	2000	20000
2	buah	KBD09	Keyboard	35000	70000
				Total	90000

Gambar 1. Faktur belanja

Tgl	Nama	Alamat	NoHP	No Faktur	Jumlah Barang	Satuan	Kode Barang	Nama Barang	Harga Satuan	Jumlah Harga	Total
1/5/21	Fulan	JI.ABC 7	0812345	035	10	keping	DVD17	DVD blank	2000	20000	90000
					2	buah	KBD09	Keyboard	35000	70000	

**Gambar 2.** Tabel UNF (*unnormalized form* 'bentuk tidak normal')

Tgl	Nama	Alamat	NoHP	No Faktur	Jumlah Barang	Satuan	Kode Barang	Nama Barang	Harga Satuan	Jumlah Harga	Total
1/5/21	Fulan	JI.ABC 7	0812345	035	10	keping	DVD17	DVD blank	2000	20000	90000
1/5/21	Fulan	JI.ABC 7	0812345	035	2	buah	KBD09	Keyboard	35000	70000	90000

**Gambar 3.** Relasi 1NF (*first normal form* 'bentuk normal pertama')

### 1NF (First Normal Form 'Bentuk Normal Pertama')

Relasi yang memenuhi 1NF adalah relasi yang setiap perpotongan baris dan kolomnya berisi satu dan hanya satu nilai. Supaya bernilai tunggal, tabel UNF pada Gambar 3 dimodifikasi dengan memisahkan nilai sel yang tidak tunggal ke baris baru sehingga diperoleh relasi seperti pada Gambar 4. Perhatikan bahwa setiap perpotongan baris dan kolomnya berisi satu dan hanya satu nilai. Ini berarti relasi tersebut telah memenuhi syarat 1NF.

Dengan demikian, dari faktur belanja diperoleh sebuah relasi 1NF yang skema relasinya sebagai berikut:

***FakturBelanja*** (Tgl, Nama, Alamat, NoHP, NoFaktur, Jumlah, Satuan, KodeBarang, NamaBarang, HargaSatuan, Jumlah Harga, Total)

Dua atribut terakhir adalah atribut turunan dan boleh diabaikan (boleh tidak disimpan dalam basis data). Skema relasinya menjadi sebagai berikut:

***FakturBelanja*** (Tgl, Nama, Alamat, NoTelepon, NoFaktur, Jumlah, Satuan, KodeBarang, NamaBarang, HargaSatuan)

# FIRST NORMAL FORM (1NF)

Unnormalized  
Not 1NF

Tabel: Buku

repeated

ISBN	Thn_Ter	ID_Pengara	Nama_Pengaran	ID_Pengara	Nama_Pengara
12-1202-19222	1992	K0121	Aris M	K0121	Kosim P
12-1202-19222	2001	K1021	Kosim P		
11-1090-29101	2001	K2091	K Odelia	K0121	Aris M
11-1090-29102	2002	K2092	Renaldi	K2091	K Odelia
12-1201-90871	2001	K2019	Samsuri J		

1NF

ISBN	Thn_Terbit	ID_Pengarang	Nama_Pengarang
12-1202-19222	1992	K0121	Aris M
12-1202-19222	1992	K1021	Kosim P
11-1090-29101	2001	K1021	Kosim P
11-1090-29102	2001	K2091	K Odelia
11-1090-29102	2001	K0121	Aris M
12-1201-90871	2002	K2092	Renaldi
12-1201-90871	2002	K2091	K Odelia
13-2089-12910	2001	K2019	Samsuri J

# SECOND NORMAL FORM (2NF)

- Suatu relasi disebut memenuhi bentuk normal kedua (2NF) jika dan hanya jika :
  1. memenuhi 1NF
  2. setiap atribut yang bukan kunci utama tergantung secara fungsional terhadap semua atribut kunci dan bukan hanya sebagian atribut kunci (fully functionally dependent).
- Untuk normalisasi ke bentuk 2NF, maka tabel 1NF didekomposisi menjadi beberapa tabel yang masing-masing memenuhi 2NF.
- Bila terdapat ketergantungan parsial maka : eliminate.
- Tujuan membentuk 2NF :
  - ✓ semantik tabel 2NF menjadi lebih eksplisit (fully FD)
  - ✓ mengurangi update anomali yang masih mungkin terjadi pada 1NF

## 2NF (Second Normal Form 'Bentuk Normal Kedua')

Relasi 2NF adalah relasi yang memenuhi 1NF dan setiap atribut bukan *primary key* memiliki ketergantungan fungsional penuh pada *primary key*. Jadi, ada dua hal yang berkaitan dengan relasi 2NF, yaitu *primary key* dan ketergantungan fungsional.

### Ketergantungan Fungsional

*Jika A dan B adalah atribut relasi R, maka B dinyatakan memiliki ketergantungan fungsional pada A (ditulis  $A \rightarrow B$ ), jika setiap nilai A berasosiasi tepat dengan satu nilai B. A dan B boleh jadi terdiri atas satu atau beberapa atribut.*

*Determinan ketergantungan fungsional adalah atribut atau sekumpulan atribut yang ada pada sisi kiri anak panah.*

### Ketergantungan fungsional penuh

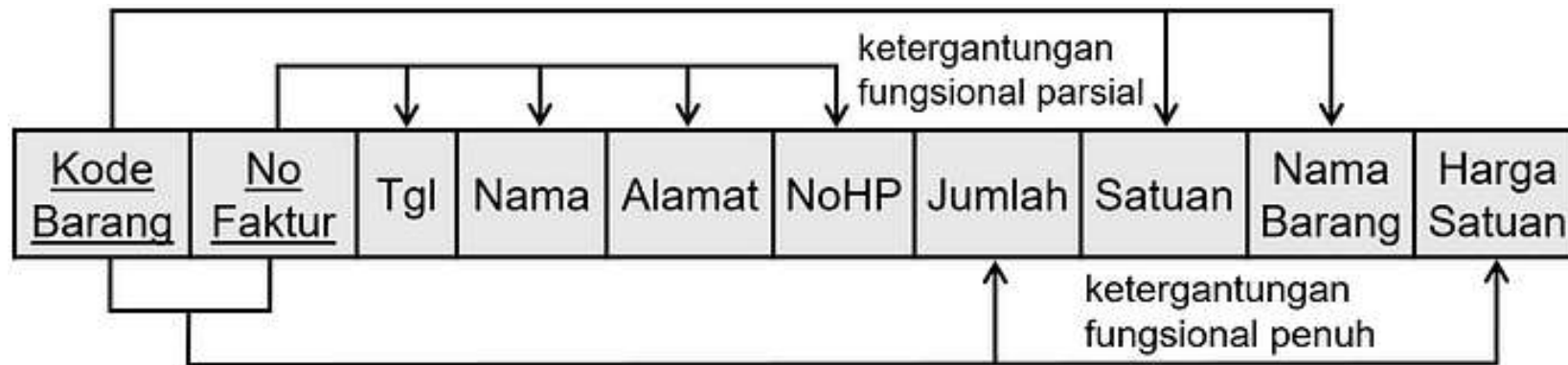
*Bila A dan B adalah atribut sebuah relasi, maka B dinyatakan memiliki ketergantungan fungsional penuh pada A jika B bergantung fungsional pada A secara keseluruhan dan bukan pada sebagian dari A. A adalah primary key atribut tunggal atau primary key komposit yang merupakan gabungan dari beberapa atribut.*

Karena persyaratan relasi 2NF berkaitan dengan *primary key*, maka harus ditentukan terlebih dahulu *primary key* relasi 1NF. Tidak satu pun atribut tunggal memenuhi syarat sebagai *primary key*.

Oleh karena itu, diambil gabungan dua atribut yaitu *NoFaktur + KodeBarang* sebagai *primary key* komposit sehingga skema relasinya menjadi:

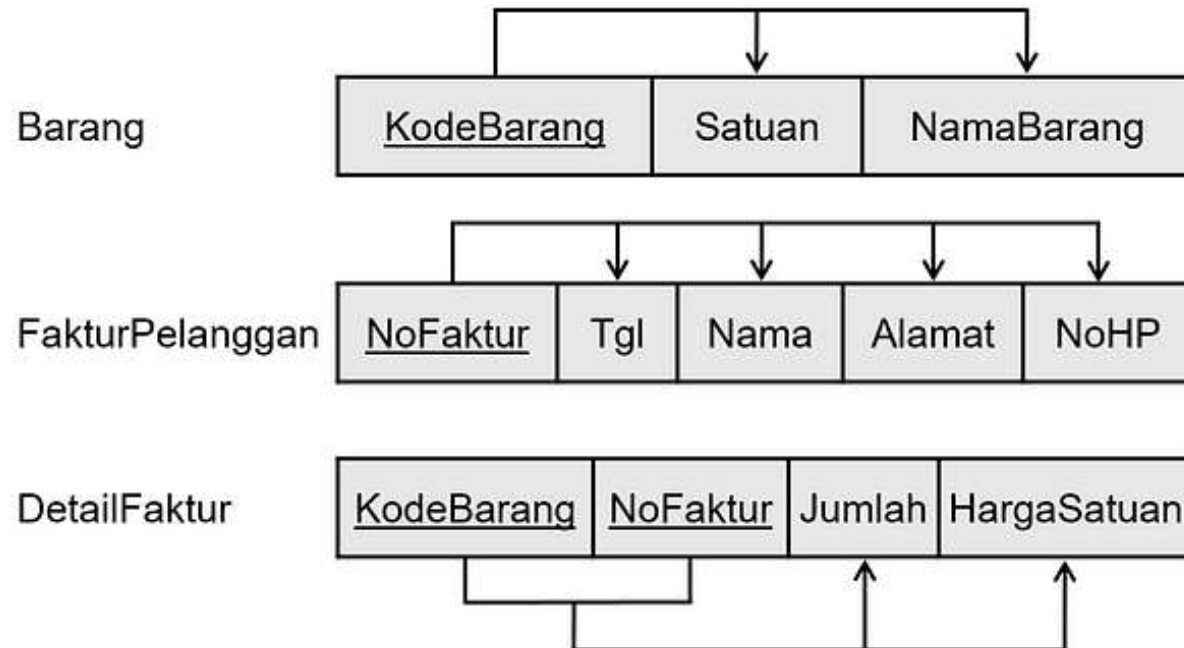
***FakturBelanja*** (*NoFaktur, KodeBarang*, *Tgl, Nama, Alamat, NoTelepon, Jumlah, Satuan, NamaBarang, HargaSatuan*)

Lihat Gambar 4. Ada dua kelompok atribut yang memiliki ketergantungan fungsional pada sebagian *primary key* (ketergantungan parsial). Pertama, atribut *Satuan*, *NamaBarang*, memiliki ketergantungan fungsional hanya pada *KodeBarang*; kedua, atribut *Tgl*, *Nama*, *Alamat*, dan *NoHP* memiliki ketergantungan fungsional hanya pada *NoFaktur*. Ini berarti relasi *FakturBelanja* memang belum memenuhi 2NF.



**Gambar 4.** Diagram ketergantungan fungsional pada relasi 1NF FakturBelanja

Kelompok atribut yang memiliki ketergantungan fungsional parsial dipisahkan ke dalam relasi berbeda sehingga dari relasi 1NF *FakturBelanja* diperoleh tiga buah relasi 2NF yaitu *Barang*, *FakturPelanggan* dan *DetailFaktur*. Lihat Gambar 5. Semua atribut yang bukan *primary key* pada ketiga relasi memiliki ketergantungan fungsional penuh pada *primary key*.



Gambar 5. Relasi 2NF

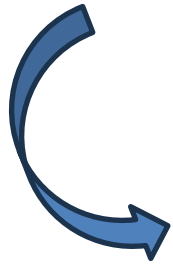
### **3NF (Third Normal Form 'Bentuk Normal Ketiga')**

Relasi 3NF adalah relasi yang memenuhi 1NF, 2NF dan atribut yang bukan *primary key* tidak memiliki ketergantungan transitif pada *primary key*.

#### **Ketergantungan Transitif**

*Misal A, B, C adalah atribut sebuah relasi. Jika  $A \rightarrow B$  dan  $B \rightarrow C$ , maka dikatakan C memiliki ketergantungan transitif pada A melalui B. Dengan kata lain, ketergantungan transitif adalah ketergantungan fungsional di antara atribut bukan primary key.*

Atribut bukan *primary key* setiap relasi 2NF harus dicek apakah memiliki ketergantungan transitif. Lihat Gambar 5. Skema ketiga relasi adalah sebagai berikut.



***Barang*** (*KodeBarang*, *Satuan*, *NamaBarang*)  
***DetailFaktur*** (*KodeBarang*, *NoFaktur*, *Jumlah*, *HargaSatuan*)  
***FakturPelanggan*** (*NoFaktur*, *Tgl*, *Nama*, *Alamat*, *NoHP*)

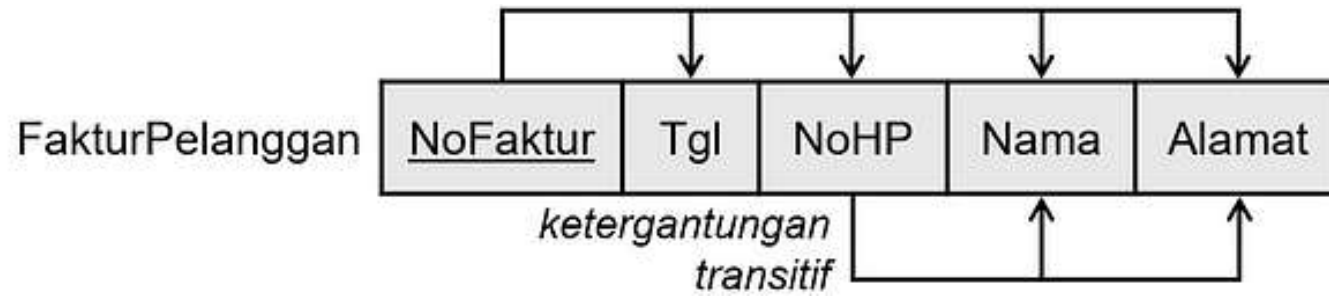
Pada relasi *Barang*, tidak ada ketergantungan antara atribut *Satuan* dengan *NamaBarang*. Begitu pula pada relasi *DetailFaktur*, tidak ada ketergantungan antara atribut *Jumlah* dengan *HargaSatuan*. Ini berarti keduanya sudah memenuhi 3NF karena tidak memiliki ketergantungan transitif.

Pada relasi *FakturPelanggan*, atribut *NoHP* bernilai unik, ada ketergantungan transitif *Nama* dan *Alamat* pada *NoFaktur* melalui *NoHP* sebagai berikut:

*NoFaktur* → *NoHP*

*NoHP* → *Nama*, *Alamat*

Dengan demikian, relasi *FakturPelanggan* belum memenuhi 3NF. Lihat Gambar 6.

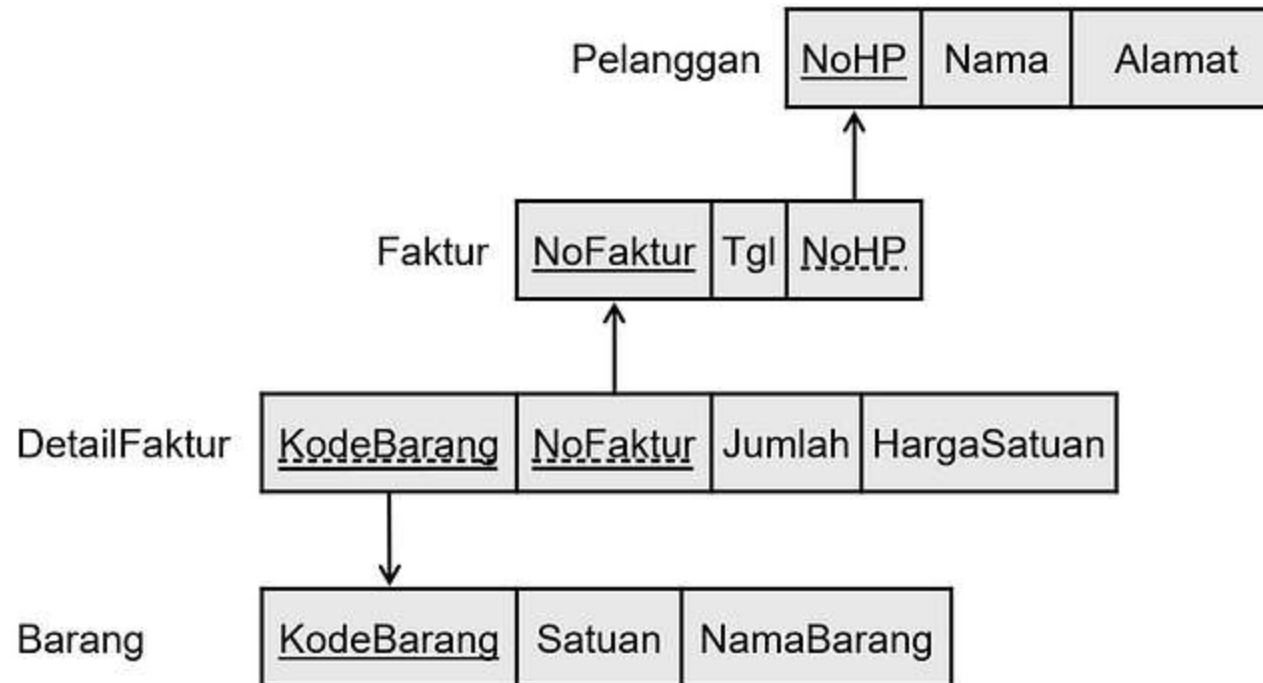


Gambar 6. Ketergantungan transitif



Gambar 7. Relasi 3NF

Selanjutnya, kelompok atribut yang memiliki ketergantungan transitif dipisahkan ke dalam relasi berbeda sehingga dari relasi 2NF *FakturPelanggan* diperoleh dua buah relasi 3NF yaitu relasi *Faktur* dan relasi *Pelanggan*. Lihat Gambar 7.



**Gambar 8.** Hasil normalisasi sampai dengan 3NF

Jadi, hasil akhir proses normalisasi data faktur belanja adalah empat buah relasi 3NF yaitu relasi *Pelanggan*, *Faktur*, *DetailFaktur* dan *Barang*. Lihat Gambar 8. Keterkaitan antar relasi dapat dilihat dari atribut *foreign key* yang digarisbawahi dengan garis terputus-putus merujuk ke atribut *primary key* yang digarisbawahi dengan garis menerus.

# SECOND NORMAL FORM (2NF)

- Contoh :

Diketahui tabel  $R=(A,B,C,D,E)$  ;  $A,B$  kunci utama (primary key)

dengan FD :  $A,B \downarrow C,D,E$  maka tabel  $R$  memenuhi 2NF sebab :

$A,B \downarrow C,D,E$  berarti :

$A,B \downarrow C,$

$A,B \downarrow D$  dan

$A,B \downarrow E$

Jadi semua atribut bukan kunci utama tergantung penuh pada  $(A,B)$ .

# SECOND NORMAL FORM (2NF)

- Contoh:

Bagaimana bila  $R = (A,B,C,D,E)$  tetapi dengan

FD :  $(A,B) \downarrow (C,D)$  dan  $B \downarrow E$ . Apakah memenuhi 2NF ?

Jelas bahwa R bukan 2NF karena ada atribut E yang bergantung hanya pada atribut B saja dan bukan terhadap (A,B).

Dari FD :  $(A,B) \downarrow (C,D)$  juga mencerminkan bahwa hanya C dan D saja yang bergantung secara fungsional terhadap (A,B), tidak untuk E.

Jadi bukan 2NF.

Untuk mengubah menjadi 2NF, lakukan dekomposisi menjadi :

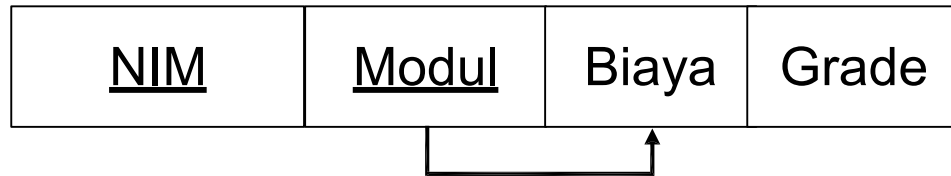
$R1 = (A,B,C,D)$  dan  $R2 = (B,E)$ . Tampak R1 dan R2 memenuhi 2NF.

# SECOND NORMAL FORM (2NF)

- Contoh:

Diketahui Workshop = (NIM, Modul, Biaya, Grade)

Peserta Workshop



Key : NIM+Modul

FD : Modul ↓ Biaya

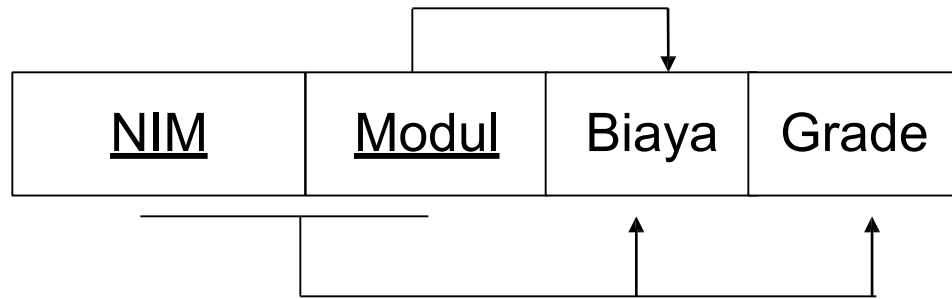
(Biaya ditentukan oleh Modul yang diambil mahasiswa)

Tabel biaya peserta workshop

<u>NIM</u>	<u>Modul</u>	Biaya	Grade
P11.2004.0129	VB.Net	250000	A
P11.2004.0130	Prolog	100000	A
P11.2004.0129	Prolog	100000	B
P11.2004.0201	Delphi 6	150000	A
P11.2004.0250	VB.Net	250000	B

- 1NF
- Not 2NF  
Sebab dalam tabel ini, biaya tidak bergantung penuh pada atribut kunci (NIM, Modul)

# SECOND NORMAL FORM (2NF)

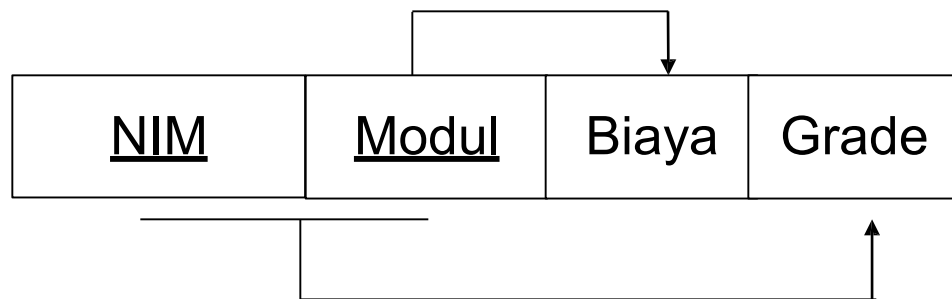


(NIM,Modul) = key

(NIM,Modul) → Biaya (partial)

(NIM,Modul) → Grade (full)

**Eliminate**



**Make Decomposition :**

Works1 = (NIM,Modul,Grade)

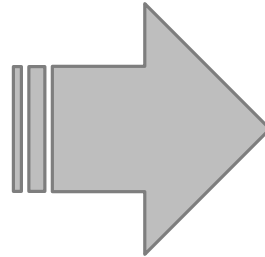
Works2 = (Modul,Biaya)

**Fully Dependency**

# SECOND NORMAL FORM (2NF)

Workshop

<u>NIM</u>	<u>Modul</u>	Biaya	Grade
P11.2004.0129	VB.Net	250000	A
P11.2004.0130	Prolog	100000	A
P11.2004.0129	Prolog	100000	B
P11.2004.0201	Delphi 6	150000	A
P11.2004.0250	VB.Net	250000	B



<u>NIM</u>	<u>Modul</u>	Grade
P11.2004.0129	VB.Net	A
P11.2004.0130	Prolog	A
P11.2004.0129	Prolog	B
P11.2004.0201	Delphi 6	A
P11.2004.0250	VB.Net	B

Works1

<u>Modul</u>	Biaya
VB.Net	250000
Prolog	100000
Delphi 6	150000

Works2



# THIRD NORMAL FORM (3NF)

- Suatu relasi disebut memenuhi bentuk normal ketiga (3NF) jika dan hanya jika :
  1. memenuhi 2NF
  2. setiap atribut yang bukan kunci tidak tergantung secara fungsional terhadap atribut bukan kunci yang lain dalam relasi tsb (tidak terdapat ketergantungan transitif pada atribut bukan kunci).

Atau

- Suatu relasi disebut memenuhi bentuk normal ketiga (3NF) jika dan hanya jika setiap FD nontrivial :  $X \downarrow A$ , di mana  $X$  dan  $A$  atribut (atau kompositnya), memenuhi salah satu kondisi :
  1.  $X$  adalah superkey
  2.  $A$  merupakan anggota candidate key ( $A$  disebut prime attribute)

# THIRD NORMAL FORM (3NF)

- Jika suatu relasi sudah memenuhi 2NF tapi tidak memenuhi 3 NF, maka untuk normalisasi ke bentuk 3NF, tabel 2NF didekomposisi menjadi beberapa tabel hingga masing-masing memenuhi 3NF.
- Tujuan membentuk 3NF :
  - ✓ semantik tabel 3NF menjadi lebih eksplisit (fully FD hanya pada primary key).
  - ✓ menghindari update anomali yang masih mungkin terjadi pada 2NF.
- Catatan:

*Jika suatu relasi memenuhi 2NF dan hanya memiliki tepat satu atribut yang bukan kunci utama maka relasi tsb memenuhi 3NF*

# THIRD NORMAL FORM (3NF)

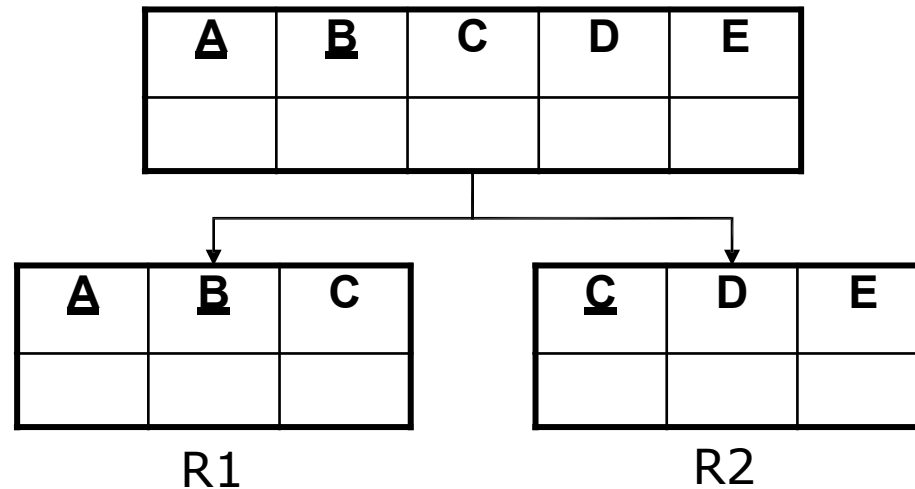
Contoh :

- Diketahui tabel  $R=(A,B,C,D,E)$  ;  $A,B$  kunci utama (primary key) dengan FD :  $A,B \downarrow C,D,E$  dan  $C \downarrow D,E$  maka  $R$  bukan 3NF sebab : Atribut  $D$  dan  $E$  (bukan kunci utama) bergantung secara fungsional pada  $C$  (yang juga bukan kunci utama).
- Melalui FD :
  - ✓ Diketahui  $A,B \downarrow C,D,E$ .  
Karena sifat refleksif maka  $A,B \downarrow A,B$ . Sehingga  $A,B \downarrow A,B,C,D,E$   
 $(A,B)$  : Superkey.
  - ✓ Diketahui  $C \downarrow D,E$ .  
Karena sifat refleksif maka  $C \downarrow C$ . Sehingga  $C \downarrow C,D,E$ .  
Karena  $C \downarrow A,B,C,D,E$  maka  $C$  bukan superkey.
  - ✓ Tidak memenuhi definisi 3NF. Jadi  $R$  bukan 3NF.
- Agar  $R$  memenuhi 3NF maka didekomposisi menjadi :  $R1=(A,B,C)$  dan  $R2=(C,D,E)$  sehingga  $R1$  dan  $R2$  memenuhi 3NF.

# THIRD NORMAL FORM (3NF)

FD :  $A, B \rightarrow C, D, E$  berarti  $A, B \rightarrow C$  ;  $C \rightarrow D, E$  ;  $A, B \rightarrow D, E$   
 $A, B \rightarrow D$  reduce }  
 $A, B \rightarrow E$  reduce }  $\uparrow$

Dekomposisinya :  $R_1 = (\underline{A}, \underline{B}, C)$  ; FD :  $(A, B) \rightarrow C$   
 $R_2 = (\underline{C}, D, E)$  ; FD :  $C \rightarrow D, E$   
R



# THIRD NORMAL FORM (3NF)

Misal diketahui struktur informasi dari suatu dokumen supplier :

S	Status	City	PQ	
			P	Qty
S1	20	LONDON	P1	300
			P2	200
			P3	400
			P4	200
			P5	100
			P6	100
S2	10	PARIS	P1	300
			P2	400
S3	10	PARIS	P2	200
S4	20	LONDON	P2	200
			P4	399
			P5	400

Akan dibentuk suatu tabel dengan skema

$TPS = (\underline{S}, \text{Status}, \text{City}, P, \text{Qty})$

dengan  $(S, P) = \text{primary key}$

dan berlaku FD :

$S \rightarrow \text{Status}$

$S \rightarrow \text{City}$

$\text{City} \rightarrow \text{Status}$

Lakukan normalisasi dari 1NF hingga 3NF.

# THIRD NORMAL FORM (3NF)

TPS

<u>S</u>	Status	City	<u>P</u>	Qty
S1	20	LONDON	P1	300
S1	20	LONDON	P2	200
S1	20	LONDON	P3	400
S1	20	LONDON	P4	200
S1	20	LONDON	P5	100
S1	20	LONDON	P6	100
S2	10	PARIS	P1	300
S2	10	PARIS	P2	400
S3	10	PARIS	P2	200
S4	20	LONDON	P2	200
S4	20	LONDON	P4	399
S4	20	LONDON	P5	400

- 1NF
- Not 2NF

Problem :

- Redundansi → inconsistency (low speed process)
- Anomaly :  
S → (Status, City) tapi kita tidak bisa insert data (S5,30,JAKARTA) tanpa diikuti data P (khususnya) dan Q. Menghapus 1 baris data akan jg merusak keutuhan informasi.
- Solusi :  
Dekomposisi menjadi :  
TPS1 dan TPS2

# THIRD NORMAL FORM (3NF)

TPS1

<u>S</u>	Status	City
S1	20	LONDON
S2	10	PARIS
S3	10	PARIS
S4	20	LONDON

- 1NF
- 2NF
- Not 3NF (trans.)  
S → City  
City → Status

- Sekarang kita dapat menambah data (S5,30,JAKARTA) secara aman
- Tapi masih ada anomaly :  
Karena City → Status maka kita tidak bisa entry data City baru sebelum Status punya nilai. Penghapusan 1 baris sebagian data City juga bisa merusak keutuhan informasi S.
- Selain itu, masih ada redundansi pada Status dan City

TPS2

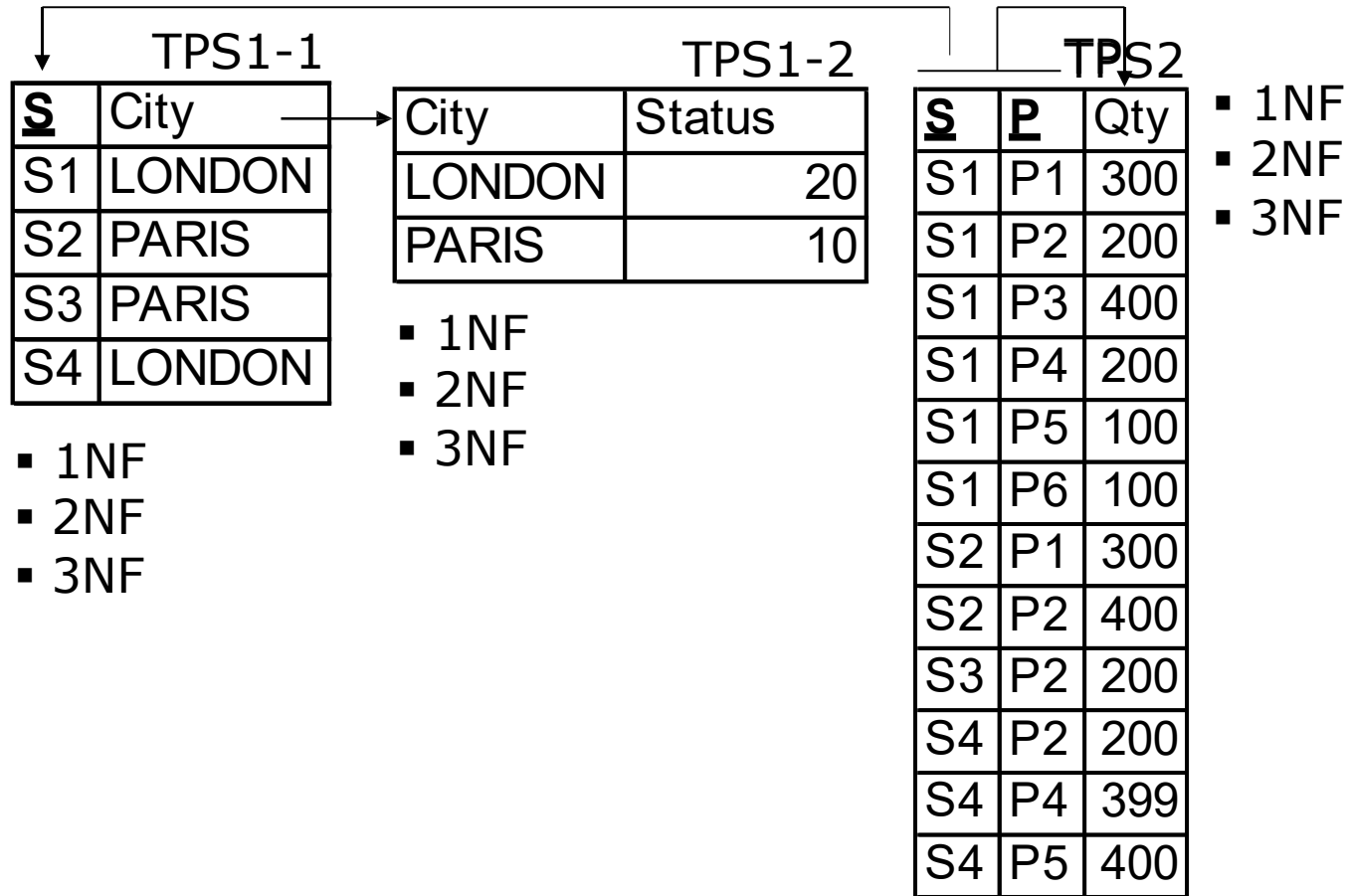
<u>S</u>	<u>P</u>	Qty
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	399
S4	P5	400

- 1NF
- 2NF
- 3NF

Redundansi partial  
→ tidak potensial  
→ lebih baik dari redundan sebelumnya

*Tidak mungkin menghilangkan semua redundan tapi buat yang minimal*

# THIRD NORMAL FORM (3NF)



# THIRD NORMAL FORM (3NF)

1. Diberikan skema relasi  $R = (A, B, C, D, E, F, G, H, I, J, K)$  dengan ketergantungan fungsional:

$$A \rightarrow B, C, D ; C \rightarrow D ; E \rightarrow F ; A, E \rightarrow G, H, I, J, K ; I \rightarrow J, K$$

Apakah  $R$  memenuhi 3NF ? Jika tidak, rancanglah skema relasi  $R$  sedemikian sehingga memenuhi bentuk 3NF. Bila Saudara melakukan dekomposisi tabel, lengkapi dengan uji dekomposisi dan uji lossless.

2. Diketahui  $R = (A, B, C, D, E, F, G, H)$  dimana  $(A, B)$  : primary key

Ketergantungan fungsional yang berlaku (FD) :

$$A \rightarrow C, F ; B \rightarrow G, H ; A, B \rightarrow D, E \text{ dan } D \rightarrow E$$

a. Jika diketahui bahwa  $R$  memenuhi 1NF, apakah  $R$  memenuhi 2NF ? 3NF ?

b. Jika tidak, rancanglah skema relasi  $R$  sedemikian sehingga memenuhi bentuk 2NF dan 3NF.

c. Lengkapi dengan uji dekomposisi dan uji lossless.

# 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.**

# BOYCE CODD NORMAL FORM (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  $B \not\rightarrow A$ . Sehingga  $B \not\rightarrow A,B,C$  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.

# BOYCE CODD NORMAL FORM (BCNF)

## Contoh :

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

- 3NF ?
- BCNF ?

Jawab:

- 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.

# BOYCE CODD NORMAL FORM (BCNF)

## Students

sid	name	age
53666	Jones	18
53668	Smith	18
53669	Melissa	17
53670	Hilden	19

Students=(sid, name, age)

FD : sid  $\rightarrow$  name, age

- BCNF, sebab sid superkey

## Pinjam

idpinjam	sid	bid	date
P-01	53666	B002	10/11/2005
P-02	53668	B001	10/11/2005
P-03	53668	B004	11/12/2005
P-04	53670	B002	14/11/2005

Pinjam=(idpinjam, sid, bid, date)

FD : idpinjam  $\rightarrow$  bid, date

- Bukan BCNF, sebab idpinjam bukan superkey  
idpinjam  $\nrightarrow$  sid

## Books

bid	title	year
B001	MySQL	2002
B002	Algorithm	2003
B003	Visual Foxpro 6.0	2003
B004	Visual basic 6.0	2005

Books=(bid, title, year)

FD : bid  $\rightarrow$  title, year

- BCNF, sebab bid superkey

# BOYCE CODD NORMAL FORM (BCNF)

idpinjam	sid	bid	date
P-01	53666	B002	10/11/2005
P-02	53668	B001	10/11/2005
P-03	53668	B004	11/12/2005
P-04	53670	B002	14/11/2005

Didekomposisi menjadi :

## Pinjam1

idpinjam	sid
P-01	53666
P-02	53668
P-03	53668
P-04	53670

FD trivial  
→ BCNF

## Pinjam2

idpinjam	bid	date
P-01	B002	10/11/2005
P-02	B001	10/11/2005
P-03	B004	11/12/2005
P-04	B002	14/11/2005

idpinjam → bid, date  
idpinjam superkey  
→ BCNF

# PERBANDINGAN ANTARA BCNF DAN 3NF

- Itu selalu untuk mengurai suatu relasi ke dalam relasi di 3NF dan
  - dekomposisinya adalah lossless
  - dependensi dipertahankan
- Itu selalu untuk mengurai suatu relasi ke dalam relasi di BCNF dan
  - dekomposisinya adalah lossless
  - tidak mungkin mempertahankan ketergantungan.

# PERBANDINGAN ANTARA BCNF DAN 3NF

- Contoh kasus redundansi pada 3NF

Jadwal = (Nim, Modul, Dosen)

FD = {Dosen  $\downarrow$  Modul}

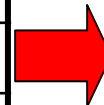
Relasi ini memenuhi 3NF, karena tidak ada ketergantungan transitif.

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	VB.Net	Ajib
P11.2004.0130	Prolog	Aris
P11.2004.0129	VB Net	Ajib
P11.2004.0201	VB Net	Budi
P11.2004.0250	Prolog	Jono
P11.2004.0260	VB.Net	Budi

**NOT BCNF**



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

**BCNF**

Dosen	<u>Modul</u>
Ajib	VB.Net
Aris	Prolog
Jono	Prolog
Budi	VB.Net

# TUJUAN DESAIN

- Tujuan untuk desain basis data relasional adalah:
  - BCNF.
  - Lossless join.
  - Dependency preservation.
- Jika tidak dapat mencapai ini, menerima salah satunya
  - Kurangnya Dependency preservation
  - Redundansi karena penggunaan 3NF

# TAHAP DESAIN

(tidak memenuhi definisi relasi)



Entity Set

Menghapus atribut multivalued & berulang. Memenuhi definisi hubungan

First Normal Form

Hapus partial dependencies

Second Normal Form

Hapus transitive dependencies

Third Normal Form

Pilih yang di mana semua deteminan adalah candidate key

Boyce-Codd Normal Form

Praktisi database kebanyakan menganggap bahwa tingkatan normalisasi hingga BCNF atau 3NF dianggap sudah cukup untuk meminimalisasi masalah dalam desain database (redundansi, lossless, dependency preservation)

# TANTANGAN PERANCANGAN BASIS DATA

Desainer harus membuat kompromi desain yang dipicu oleh tujuan yang saling bertentangan:

- standar desain (desain keanggunan atau kesetiaan),  
untuk mengembangkan desain “baik”: Tanpa Cadangan, Tidak Berlebihan, Pelestarian Ketergantungan
- kecepatan pemrosesan (kinerja), dan
  - kecepatan pemrosesan tinggi adalah prioritas "teratas" (efisiensi)
  - meminimalkan jumlah dan kompleksitas hubungan atau tabel
- persyaratan informasi  
kemampuan untuk mengirimkan semua permintaan dan pelaporan yang ditentukan persyaratan pengguna tepat waktu

**Contoh bentuk kompromi yang populer :**

**→ Denormalisasi (pelanggaran normalisasi)**

# DENORMALISASI

Design Standards Vs (processing speed, information requirements)

- Normalisasi hanyalah merupakan teknik pendekatan yang digunakan untuk mendapatkan desain database (logik) yang baik dan bukan sebagai "aturan baku" DBMS yang harus digunakan.
  - Bersifat "Normatif", memungkinkan untuk dilanggar dengan alasan : Kecepatan Proses (Efisiensi) dan Pelayanan Informasi Tepat Waktu
- Bentuk-bentuk Denormalisasi :
  - Membuat Atribut Turunan pada Tabel, mis :  $Cost = Qty * Price$
  - Atribut yang Berlebihan, mis : NIM mahasiswa sudah mencerminkan program studi mahasiswa, tetapi dalam tabel mahasiswa dibuat atribut Program Studi.
  - Summary Table (mis : summary table for report)
  - Membiarkan relasi transitif dalam satu tabel untuk kemudahan proses
- Konsekuensi Denormalisasi :  
Redundancy, Not Atomic, Worst Space dll

# LATIHAN

Berdasarkan formulir tersebut,

- Rancanglah tabel penyimpanan datanya
- Lakukan normalisasi hingga 3NF atau BCNF



## NOTA PELANGGARAN SOPIR

Nomor : .....

"ARMADA" TAXI

### IDENTITAS SOPIR

Nama \_\_\_\_\_

Alamat \_\_\_\_\_

Kota \_\_\_\_\_ Propinsi \_\_\_\_\_ Kodepos \_\_\_\_\_

No. SIM \_\_\_\_\_ Sex \_\_\_\_\_ Tanggal Lahir \_\_\_\_\_ TB \_\_\_\_\_ BB \_\_\_\_\_

### KENDARAAN

No. Kendaraan \_\_\_\_\_ Warna \_\_\_\_\_ Tahun \_\_\_\_\_ Tipe \_\_\_\_\_ No. Lambung \_\_\_\_\_

Supervisor Area \_\_\_\_\_

### PELANGGARAN

Tanggal Pelanggaran

TGL \_\_\_\_\_ BLN \_\_\_\_\_ THN \_\_\_\_\_ JAM \_\_\_\_\_

Lokasi \_\_\_\_\_

Catatan Pelanggaran \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Tindakan

Peringatan.

Kembali ke pangkalan. Tidak diijinkan mengemudi kendaraan selama 3 hari.

Dipecat dari pekerjaan sopir.

Sopir \_\_\_\_\_

Semarang, \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

Petugas Pemantau \_\_\_\_\_

Nama dan Tanda tangan \_\_\_\_\_

Nama dan Tanda tangan \_\_\_\_\_

# LATIHAN

1. Berikan alasan anda mengapa suatu basis data harus berada pada kondisi normal !
2. Kondisi apa saja yang harus dipenuhi, bahwa suatu basis data di katakan memenuhi normalisasi !
3. Diberikan suatu data sebagai berikut di bawah ini. Dari data yang diberikan belum memenuhi kondisi normal, maka tentukanlah proses normalisasi untuk menghasilkan data yang memenuhi kondisi normal !

No	Class	Time	Day	Teacher	Start	Room	Remark
1	B.1I	17.00-18.30	Tuesday; Thursday	Ms. Avi; Ms. Oki	19/08/2004	A202	Run
2	B.5I	15.30-17.00	Tuesday; Friday	Ms. Beta; Ms. Susi	20/08/2004	A302	Run
3	B.1J	17.00-18.30	Monday; Thursday	Ms. Galuh; Ms. Avi	23/08/2004	A301	Run
4	B.2J	17.00-18.30	Tuesday; Thursday	Mr. Aris; Ms. Beta	24/08/2004	A102	Run
5	B.3J	15.30-17.00	Tuesday; Thursday	Mr. Aris; Ms. Oki	05/08/2004	A103	Run
6	B.2F	15.30-17.00	Monday; Thursday	Ms. Galuh; Mr. Hery	19/08/2004	A203	Run
7	B.1I	18.30-20.00	Wednesday; Friday	Ms. Ria; Ms. Galuh	04/08/2004	A203	Pending