

# **Suplemen Materi**

# **Penerapan Teknik Normalisasi**

# **Dalam Merancang *Database***

Dr. Sutedi, S.Kom., M.T.I, MTA, MCP

# Studi Kasus

Contoh Dokumen I/O

## KARTU RENCANA STUDI

NPM : 1611050120  
Nama : Ihsan Nurdin  
Jurusan : Sistem Informasi

Tahun Akademik : 2019/2020  
Semester : 5



Kode	Nama Mata Kuliah	SKS	Kelas	Waktu	Ruang	Paraf UTS		Paraf UAS	
						Teori	Praktik	Teori	Praktik
SIPI9029	Integrasi Data	3	SI1-02	07:10-08:40 Selasa	F.2.7				
SIPI9025	Sistem Informasi Geografis	3	SI1-02	13:00-14:30 Selasa	Lab. Lanjut C (0.4)				
IEI19400	Praktek Sarja Pengabdian Masyarakat	4	SI1-01	13:00-14:30 Sabtu   14:00-16:10 Sabtu	F.2.3   F.2.3				
IEI19412	Ekripsi	4	SI1-01	10:30-12:00 Sabtu	F.2.3				

Bandar Lampung, 14 April 2020

Kepala BAAK

Neni Purwati, S.Kom, M.TI

NIK: 00500402

# Pembahasan Studi Kasus

## Langkah ke-1:

Membuat *table universal* atau *un-normalized form* (UNF)



UNF

NPM
Nama
Jurusan
ThnAkademik
Semester
KdMatakuliah
NmMatakuliah
Sks
JnsMatakuliah
Kelas
Waktu
Ruang
NIKKaBAAK
NmKaBAAK
TglCetakKRS



## Langkah ke-2:

Menentukan *primary key* pada UNF

### **Primary key** adalah:

Satu atribut atau satu *set* minimum atribut (*composite key*) yang dapat mengidentifikasi satu *set* data secara pasti.

### **Syarat sebuah Primary key:**

- ✓ *Unique* (Tidak ada yang sama)
- ✓ *Not Null* (Selalu memiliki *data value*)
- ✓ *Constant* (Tidak berubah-ubah)
- ✓ *Simple* (Sesederhana mungkin)

## Caranya:

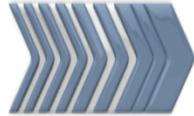
Memasukkan seluruh atribut/*field* yang relevan dan dibutuhkan ke dalam sebuah tabel

## Keterangan:

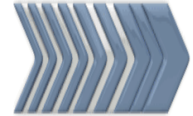
- ✓ Atribut JnsMatakuliah menunjukkan matakuliah tersebut T→Teori, P→Praktek, atau TP →Teori-Praktek.
- ✓ **Paraf petugas** tidak dimasukkan menjadi atribut karena **tidak perlu direkam ke dalam *database***.
- ✓ **Kota** tempat pencetakan KRS **tidak direkam** karena **diasumsikan bernilai standar** “Bandar Lampung”

# Pembahasan Studi Kasus

## Caranya:



- ✓ Cari sebuah atribut yang dapat mengidentifikasi set data secara pasti.
- ✓ Jika tidak dapat dilakukan dengan sebuah atribut maka gabungkan beberapa atribut seminimum mungkin yang dapat mengidentifikasi set data tersebut.
- ✓ Alternatif lain: Menghadirkan atribut baru sebagai *key*.



## Keterangan:

Dalam kasus pada tugas 2, sebuah KRS tidak dapat diidentifikasi oleh atribut manapun yang ada di KRS tersebut, sehingga penyelesaian dalam pembahasan ini dilakukan penggabungan atribut (NPM+ThnAkademik+Semester).

UNF

NPM (PK)
Semester (PK)
Nama
Jurusan
ThnAkademik
KdMatakuliah
NmMatakuliah
Sks
JnsMatakuliah
Kelas
Waktu
Ruang
NIKKaBAAK
NmKaBAAK
TglCetakKRS

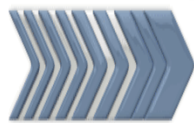
} Composite key

## Pola pikir:

- ✓ Seorang mahasiswa dapat memiliki banyak KRS, sehingga jika **NPM** digunakan sebagai *key* pencarian maka kita akan mendapatkan banyak KRS.
- ✓ Seorang mahasiswa dalam **satu tahun akademik** dapat memiliki dua buah KRS, sehingga jika **NPM + ThnAkademik** digunakan sebagai *key* pencarian maka dapat diperoleh lebih dari satu KRS.
- ✓ Seorang mahasiswa setiap semesternya hanya memiliki sebuah KRS, sehingga jika **NPM + Semester** digunakan sebagai *key* pencarian maka hanya akan didapat sebuah KRS saja yang sesuai, oleh karena itu keduanya layak menjadi *primary key*.

# Pembahasan Studi Kasus UNF

## Langkah ke-3:



Memastikan bahwa *table* yang terbentuk memenuhi kriteria normal pertama (1NF)

NPM (PK)
Semester (PK)
Nama
Jurusan
ThnAkademik
KdMatakuliah
NmMatakuliah
Sks
JnsMatakuliah
Kelas
Waktu
Ruang
NIKKaBAAK
NmKaBAAK
TglCetakKRS

Repeating group

## Kriteria 1NF:

Sebuah *table* dikatakan normal pertama (1NF) apabila *table* tersebut tidak terdapat *repeating group* atau dengan kata lain seluruh atribut *table* tersebut bersifat *atomic* (sebuah *key* hanya mengidentifikasi sebuah *record*).

## Keterangan:

- ✓ Dalam kasus ini, *table* UNF yang terbentuk tidak memenuhi kriteria 1NF karena dalam satu semester mahasiswa dapat mengambil beberapa matakuliah, sehingga NPM+Semester mengidentifikasi beberapa KdMatakuliah, NmMatakuliah, Sks, JnsMatakuliah, Kelas, Waktu, dan Ruang (*repeating group*).
- ✓ Dalam kasus lain mungkin saja *table* UNF yang terbentuk langsung memenuhi kriteria 1NF.

## Tips:

- ✓ Bila dokumen yang dianalisis mengandung *table* dan terdapat atribut lain diluar *table* maka atribut *table* tersebut biasanya akan membentuk *repeating group*.
- ✓ Bila dokumen yang dianalisis tidak mengandung *table* atau mengandung *table* namun tidak ada atribut lain selain atribut *table* maka kemungkinan besar UNF yang terbentuk langsung memenuhi kriteria 1NF.

# Pembahasan Studi Kasus

## Cara menormalkannya:

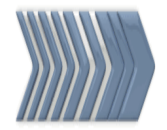
- ✓ Pisahkan seluruh atribut yang membentuk **repeating group** menjadi **table** baru.
- ✓ Tambahkan **primary key table awal** sebagai **composite key** bersama **candidate key table baru** tersebut.

## Keterangan:

**Candidate key** adalah sebuah atribut atau satu set minimum atribut yang berpotensi menjadi **primary key**

UNF

NPM (PK)	
Semester (PK)	
Repeating group	
Nama	
Jurusan	
ThnAkademik	
KdMatakuliah	
NmMatakuliah	
Sks	
JnsMatakuliah	
Kelas	
Waktu	
Ruang	
NIKKaBAAK	
NmKaBAAK	
TglCetakKRS	



1NF-1

NPM (PK)	
Semester (PK)	
Composite key table baru	
Nama	
Jurusan	
ThnAkademik	
NIKKaBAAK	
NmKaBAAK	
TglCetakKRS	

1NF-2

NPM (PK)	
Semester (PK)	
KdMatakuliah (PK)	
Candidate key table baru	
NmMatakuliah	
Sks	
JnsMatakuliah	
Kelas	
Waktu	
Ruang	

Candidate key table baru

Key table awal

# Pembahasan Studi Kasus

## Penjelasan:

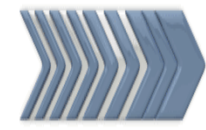
- ✓ Pada tahap ini diperoleh dua buah *table* yaitu 1NF-1 dan 1NF-2
- ✓ Untuk mengecek apakah dekomposisi tersebut *lossless-join decomposition* atau *lossy-join decomposition* maka dapat dilakukan agregasi/*join table*.
- ✓ Hasil *join table*-nya membentuk *table* awal sebelum didekomposisi.

1NF-1

NPM (PK)
Semester (PK)
Nama
Jurusan
ThnAkademik
NIKKaBAAK
NmKaBAAK
TglCetakKRS

1NF-2

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
NmMatakuliah
Sks
JnsMatakuliah
Kelas
Waktu
Ruang



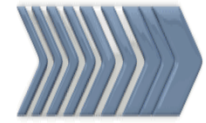
UNF

NPM (PK)
Semester (PK)
Nama
Jurusan
ThnAkademik
KdMatakuliah
NmMatakuliah
Sks
JnsMatakuliah
Kelas
Waktu
Ruang
NIKKaBAAK
NmKaBAAK
TglCetakKRS

# Pembahasan Studi Kasus

## Langkah ke-4:

Memastikan bahwa seluruh *table* 1NF yang terbentuk memenuhi kriteria 2NF



## Kriteria 2NF:

*Table* dikatakan **normal kedua (2NF)** apabila *table* tersebut **memenuhi kriteria 1NF** dan seluruh **atribut non-key-nya** bergantung secara fungsional penuh (**fully functional dependency**) terhadap *primary key-nya*. Dengan kata lain tidak ada atribut *non-key* yang bergantung pada sebagian *key* (**partial dependency**).

1NF-1

NPM (PK)
Semester (PK)
Nama
Jurusan
ThnAkademik
NIKKaBAAK
NmKaBAAK
TglCetakKRS

*Partial dependency*

*fully functional dependency*

1NF-2

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
NmMatakuliah
Sks
JnsMatakuliah
Kelas
Waktu
Ruang

*Partial dependency*

*fully functional dependency*

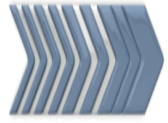
## Penjelasan:

- ✓ Pada kasus ini **kedua table 1NF** yang dihasilkan **tidak memenuhi kriteria 2NF**.
- ✓ Pada kasus lain, mungkin saja *table* 1NF yang dihasilkan langsung memenuhi kriteria 2NF.
- ✓ Pada **table 1NF-1**, untuk mengetahui **Nama** dan **Jurusan** mahasiswa pada KRS cukup didasarkan pada NPM saja, oleh karena itu keduanya **partial dependency** terhadap *key*. Sedangkan untuk mengetahui **ThnAkademik**, **NIKKaBAAK**, **NmKaBAAK**, dan **TglCetakKRS** yang ada pada KRS didasarkan kepada NPM+Semester, oleh karena itu atribut-atribut tersebut **fully functional dependency** terhadap *key*.

# Pembahasan Studi Kasus

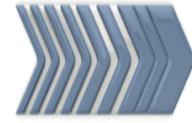
## Lanjutan penjelasan:

✓ Pada **table 1NF-2**, untuk mengetahui **NmMatakuliah, Sks, dan JnsMatakuliah** cukup didasarkan pada **KdMatakuliah** saja, oleh karena itu ketiganya **partial dependency** terhadap **key**. Sedangkan untuk mengetahui **Kelas, Waktu, dan Ruang** yang ada pada **KRS** didasarkan kepada **NPM+Semester+KdMatakuliah**, oleh karena itu atribut-atribut tersebut **fully functional dependency** terhadap **key**.



## Cara menormalkannya:

- ✓ Pisahkan atribut yang **partial dependency** menjadi **table** baru.
- ✓ Tambahkan **key partial** yang mengidentifikasi atribut tersebut sebagai **primary key** pada **table** yang baru.



1NF-1

NPM (PK)
Semester (PK)
Nama
Jurusan
ThnAkademik
NIKKaBAAK
NmKaBAAK
TglCetakKRS

Partial dependency

fully functional dependency



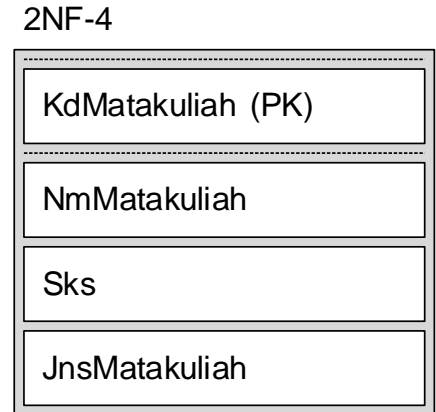
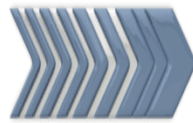
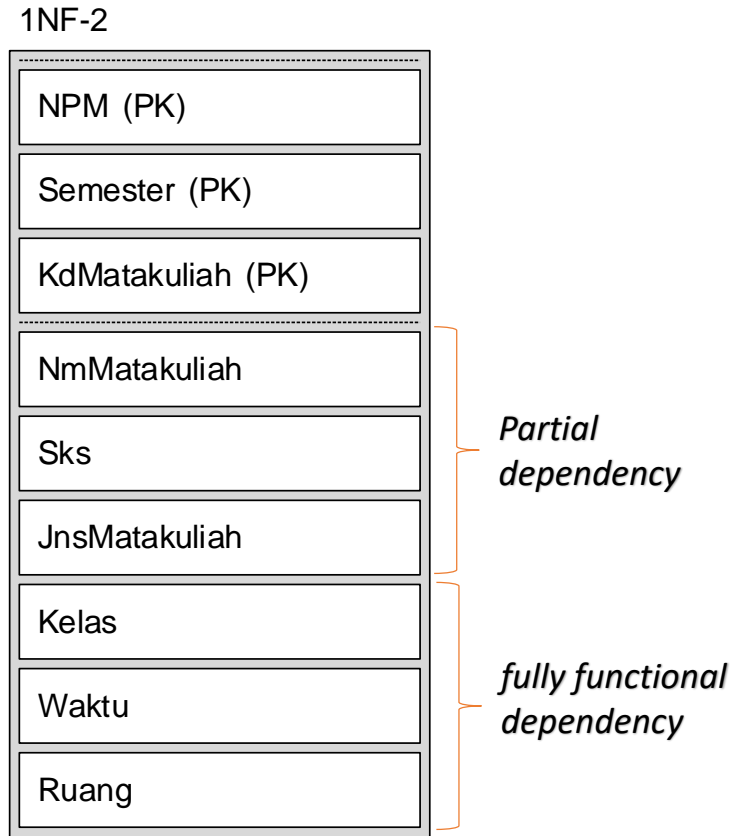
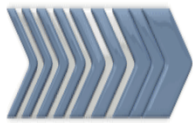
2NF-1

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
NmKaBAAK
TglCetakKRS

2NF-2

NPM (PK)
Nama
Jurusan

# Pembahasan Studi Kasus



**Tips:**

Jika *primary key table* 1NF bukan *composite key* maka tidak ada isu *partial dependency* atau dengan kata lain *table* 1NF tersebut otomatis memenuhi kriteria 2NF.

# Pembahasan Studi Kasus

## Langkah ke-5:

Memastikan seluruh *table* 2NF yang terbentuk memenuhi kriteria 3NF



## Kriteria 3NF:

*Table* dikatakan **normal ketiga (3NF)** apabila *table* tersebut memenuhi kriteria 2NF dan tidak terjadi **transitive dependency** di dalamnya. Dengan kata lain seluruh atribut *non-key* pada *table* tersebut hanya bergantung pada *primary key*-nya.

## Keterangan:

*Transitive dependency* terjadi apabila terdapat **atribut non-key** yang mengidentifikasi atribut *non-key* lainnya.

2NF-1

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
NmKaBAAK
TglCetakKRS

Transitive dependency terhadap key

2NF-3

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Kelas
Waktu
Ruang

Tidak terjadi transitive dependency

2NF-2

NPM (PK)
Nama
Jurusan

Tidak terjadi transitive dependency

2NF-4

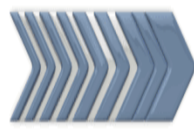
KdMatakuliah (PK)
NmMatakuliah
Sks
JnsMatakuliah

Tidak terjadi transitive dependency

# Pembahasan Studi Kasus

## Penjelasan:

- ✓ Pada *table* 2NF-1 **NmKaBAAK** bergantung secara **transitif terhadap key** karena atribut tersebut **dapat juga diidentifikasi** menggunakan **NIKKaBAAK**, dengan demikian *table* tersebut tidak memenuhi kriteria 3NF.
- ✓ Pada *table* 2NF-2, 2NF-3, dan 2NF-4 tidak terdapat *transitive dependency* sehingga ketiganya langsung memenuhi kriteria 3NF tanpa perlu perubahan.



## Cara menormalkannya:

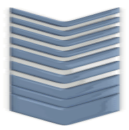
- ✓ Pisahkan atribut yang bergantung transitif terhadap *key* menjadi *table* baru.
- ✓ Tambahkan atribut *non-key* yang mengidentifikasinya sebagai *primary key* pada *table* yang baru tersebut.



2NF-1

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
NmKaBAAK
TglCetakKRS

Transitive dependency terhadap key



3NF-1

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
TglCetakKRS

3NF-2

NIKKaBAAK (PK)
NmKaBAAK

## Tips:

Jika *table* 2NF hanya memiliki sebuah atribut *non-key* atau bahkan tidak memiliki atribut *non-key* sama sekali (membentuk *solitary key*) maka bias dipastikan tidak ada isu *transitive dependency* di dalamnya sehingga otomatis *table* tersebut memenuhi kriteria 3NF.

# Pembahasan Studi Kasus

## Langkah ke-6:

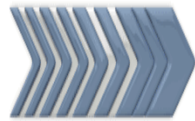
Memastikan seluruh *table* 3NF yang terbentuk memenuhi kriteria *boyce codd normal form* (BCNF)

## Kriteria BCNF:

- ✓ *Table* memenuhi kriteria BCNF apabila *table* tersebut **memenuhi kriteria 3NF** yang tidak terdapat *candidate key* selain *primary key*-nya atau dengan kata lain jika dan hanya jika *primary key* relasi tersebut adalah *candidate key*.
- ✓ Tidak ada *candidate key* yang menyebabkan anomali

## Keterangan:

- ✓ Pada kasus ini tidak ditemukan adanya *candidate key* pada seluruh *table* 3NF sehingga seluruhnya memenuhi kriteria BCNF.
- ✓ BCNF adalah kasus khusus dalam 3NF.
- ✓ **Penjelasan lebih lanjut mengenai BCNF dapat dilihat pada 2 slide setelah ini.**



3NF-1

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
TglCetakKRS

3NF-3

NPM (PK)
Nama
Jurusan

3NF-2

NIKKaBAAK (PK)
NmKaBAAK

3NF-4

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Kelas
Waktu
Ruang

3NF-5

KdMatakuliah (PK)
NmMatakuliah
Sks
JnsMatakuliah

# Penjelasan BCNF

Tidak semua kasus mengandung isu yang diangkat dalam BCNF. Oleh karenanya untuk menjelaskan masalah BCNF digunakan contoh kasus lain.



## Rekapitulasi Presensi Mahasiswa

ID. Dosen : Ds001  
Periode Akademik : Genap 2019-2020  
Kode Matakuliah : Mk0010  
Kelas : 6SI-P2

Sesi	Jumlah Mahasiswa Hadir
1	35
2	30
...	.....
16	35

### Candidate key ke-1:

IdDosen+PeriodeAkademik  
+KdMatakuliah+KdKelas+Sesi

Alternatif-1

IdDosen (PK)
PeriodeAkademik (PK)
KdMatakuliah (PK)
KdKelas (PK)
Sesi (PK)
JmlMHSHadir

### Candidate key ke-2:

PeriodeAkademik+KdMatakuliah  
+KdKelas+Sesi

Alternatif-2

PeriodeAkademik (PK)
KdMatakuliah (PK)
KdKelas (PK)
Sesi (PK)
IdDosen
JmlMHSHadir

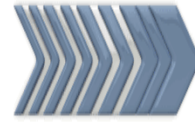
### Keterangan:

- ✓ Kedua *table* di atas memenuhi kriteria 3NF namun tidak BCNF.
- ✓ Keduanya berpotensi *delete anomalies*, dimana penghapusan *record* kehadiran mahasiswa dapat mengakibatkan hilangnya informasi matakuliah tertentu yang diampu dosen terkait.
- ✓ Keduanya juga berpotensi *insert anomalies*, dimana penambahan data kehadiran setiap sesinya mengakibatkan redundansi data IdDosen (munculnya data yang seharusnya tidak diperlukan)
- ✓ Keduanya juga berpotensi *update anomalies*, dimana perubahan IdDosen untuk matakuliah dikelas tertentu harus dilakukan berulang kali sejumlah *record* terkait.

# Penjelasan BCNF

## Cara menormalkannya:

Lakukan dekomposisi terhadap *table* tersebut sehingga tidak *candidate key* lain kecuali *primary key*-nya.



BCNF-1

IdDosen (PK)
PeriodeAkademik (PK)
KdMatakuliah (PK)
Kelas (PK)

Membentuk *Solitary key*

BCNF-2

PeriodeAkademik (PK)
KdMatakuliah (PK)
Kelas (PK)
Sesi (PK)
JmIMHSHadir



## Penjelasan:

- ✓ Kebetulan pada kasus ini salah satu *table* BCNF nya membentuk *solitary key*.
- ✓ *Solitary key* adalah *table* atau relasi yang seluruh atributnya membentuk *primary key* (*Composite key*).
- ✓ BCNF-1 pada kasus ini untuk merekam data matakuliah dan kelas yang diampu oleh masing-masing dosen.
- ✓ BCNF-2 pada kasus ini merekam data jumlah mahasiswa yang hadir pada setiap sesi dari matakuliah tertentu di kelas tertentu pada periode akademik tertentu.



Kembali ke **Pembahasan Studi Kasus**

# Pembahasan Studi Kasus

## Langkah ke-7:

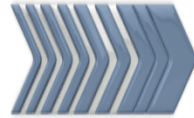
Memastikan seluruh *table* BCNF yang terbentuk memenuhi kriteria normal keempat (4NF).

## Kriteria 4NF:

- ✓ *Table* memenuhi kriteria 4NF apabila *table* tersebut memenuhi kriteria BCNF dan tidak mengandung *multi-valued attributes*.

## Keterangan:

- ✓ Pada BCNF-1, **TglCetakKRS** berpotensi *multi-valued* jika historis pencetakan KRS diperlukan untuk direkam ke dalam *database*, namun jika yang direkam hanya tanggal paling akhir pencetakan KRS nya saja maka *field* tersebut tidak *multi-valued*. (dalam kasus ini diasumsikan yang direkam hanya tanggal terakhir pencetakannya saja).
- ✓ Pada BCNF-2, BCNF-3, dan BCNF-5 tidak terdapat atribut yang *multi-valued*.
- ✓ Pada BCNF-4, *field Waktu* dan **Ruang** merupakan atribut *multi-valued* karena matakuliah yang diikuti mahasiswa dapat **memiliki lebih dari satu waktu** perkuliahan dan **lebih dari satu ruang** pelaksanaan kuliah.



BCNF-1

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
TglCetakKRS

BCNF-3

NPM (PK)
Nama
Jurusan

BCNF-2

NIKKaBAAK (PK)
NmKaBAAK

BCNF-4

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Kelas
Waktu
Ruang

*Multi-valued attributes*

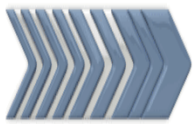
BCNF-5

KdMatakuliah (PK)
NmMatakuliah
Sks
JnsMatakuliah

# Pembahasan Studi Kasus

## Penjelasan:

- ✓ Jika perancang *database* jeli mencermati *field* demi *field* di tahap awal, maka isu ini dapat saja diselesaikan dengan pola penanganan *repeating group* pada kriteria 1NF. Hal tersebut yang menyebabkan kenapa sering kali para perancang *database* dapat langsung mengimplementasikan hasil rancangannya pada kriteria 3NF/BCNF.
- ✓ Kriteria 4NF memang dipersiapkan untuk mengantisipasi ketidakjelian para perancang *database* terkait dengan isu *multi-valued attributes*.
- ✓ Prinsipnya kriteria ini memastikan seluruh atribut yang terbentuk bersifat atomic (*single-valued attributes*).



BCNF-4

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Kelas
Waktu
Ruang

Multi-valued attribute



## Cara menormalkannya:

- ✓ Pisahkan atribut yang *multi-valued* menjadi *table* baru, kemudian tambahkan *primary key* dari *table* awal menjadi *composite key* pada *table* baru bersama *candidate key* yang ada.
- ✓ Jika atribut yang *multi-valued* hanya satu, maka *table* baru yang terbentuk akan menjadi *solitary key*.
- ✓ Jika terdapat lebih dari satu atribut *multi-valued*, maka cek hubungan antar atribut tersebut. Jika terdapat keterkaitan maka dimasukkan dalam satu *table* yang sama, namun jika tidak ada keterkaitan maka dibuat beberapa *tables* baru yang berbeda.

4NF-1

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Kelas

4NF-2

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Waktu (PK)
Ruang

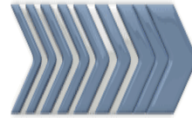
# Pembahasan Studi Kasus

## Langkah ke-8:

Memastikan seluruh *table* 4NF yang terbentuk memenuhi kriteria normal kelima (5NF).

## Kriteria 5NF:

- ✓ *Table* memenuhi kriteria 5NF apabila *table* tersebut **memenuhi kriteria 4NF** dan tidak memiliki sisa *anomalies*.



## Keterangan:

- ✓ Kriteria ini sifatnya lebih menghaluskan desain *database* yang sudah dibuat.
- ✓ Dibutuhkan kejelian dan *instinct* perancang *database* dalam melakukan ini.
- ✓ Jam terbang perancang *database* sangat membantu dalam melakukan hal ini.

4NF-1

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Kelas

4NF-3

NPM (PK)
Semester (PK)
ThnAkademik
NIKKaBAAK
TglCetakKRS

4NF-5

NPM (PK)
Nama
Jurusan

Berpotensi *anomalies*

4NF-2

NPM (PK)
Semester (PK)
KdMatakuliah (PK)
Waktu (PK)
Ruang

Berpotensi *anomalies*

4NF-4

NIKKaBAAK (PK)
NmKaBAAK

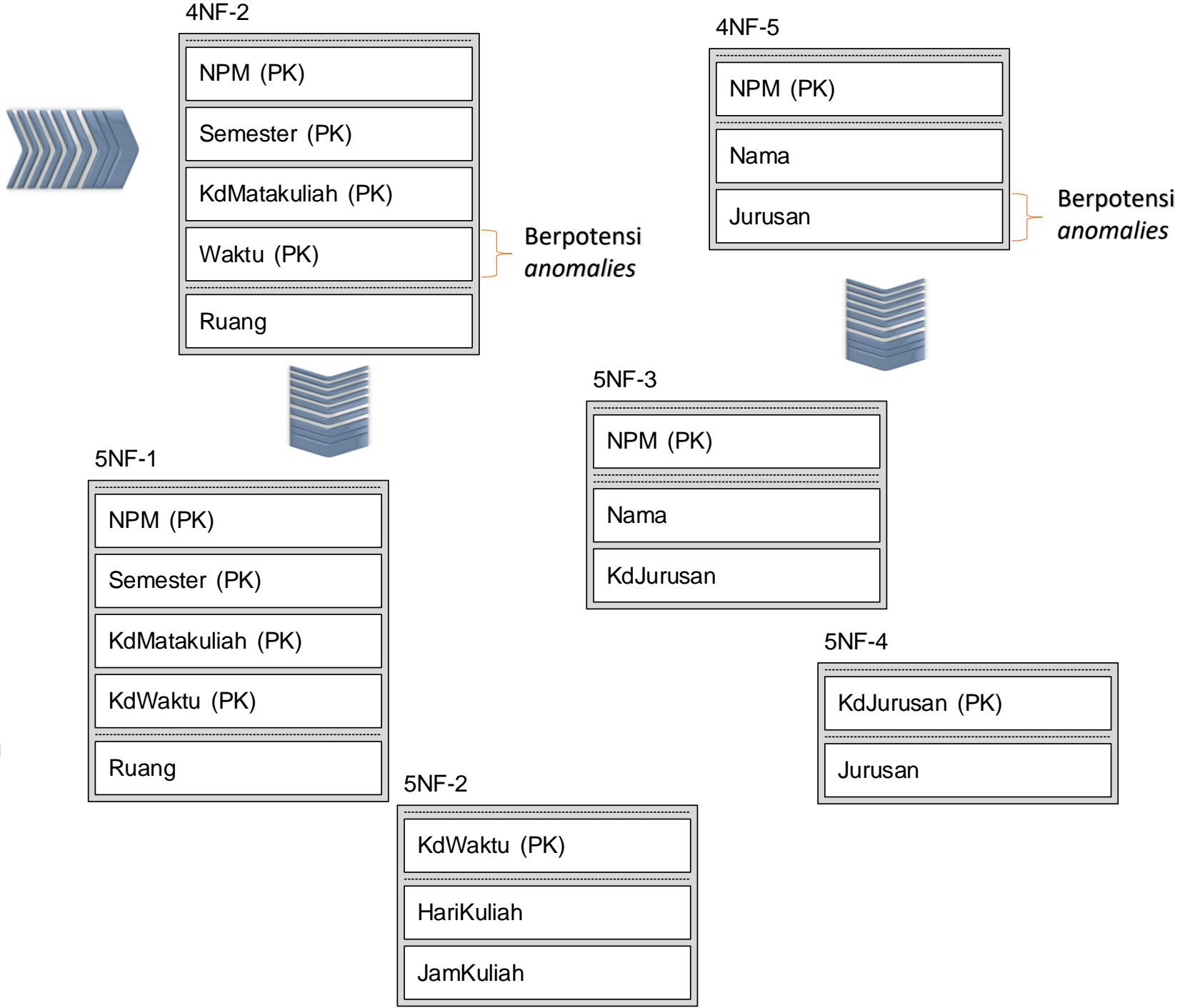
# Pembahasan Studi Kasus

## Cara menormalkannya:

Lakukan dekomposisi table hingga tidak ada lagi potensi *anomalies*.

## Keterangan:

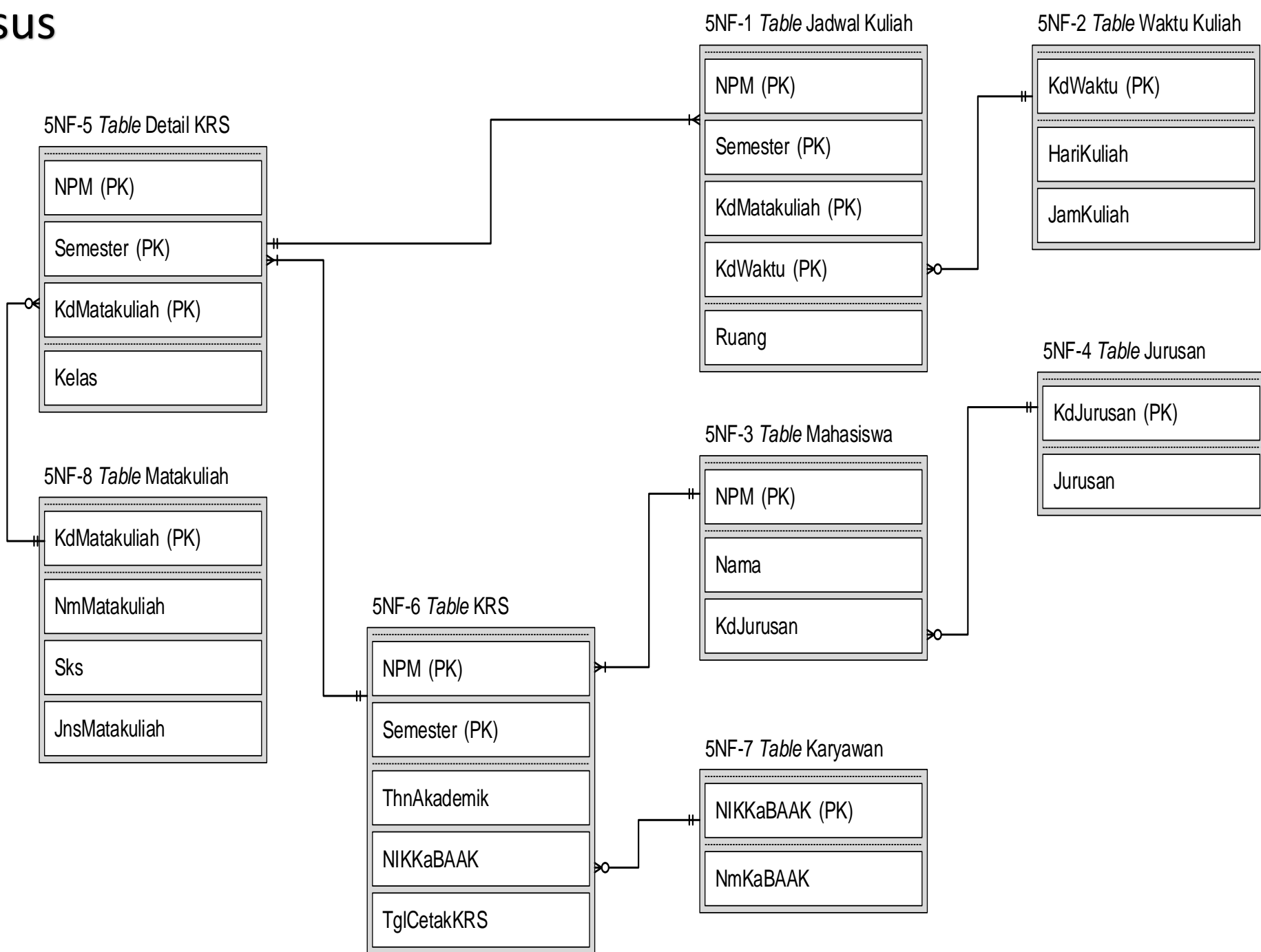
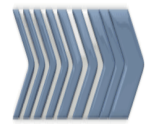
- ✓ **Field Waktu** pada *table* 4NF-2 berpotensi mengakibatkan *insert anomaly*, dimana data waktu yang diinputkan dapat terjadi *inconsistency* dan pemborosan *storage space*.
- ✓ Atribut **Jurusan** pada *table* 4NF-5 berpotensi mengakibatkan *insert anomaly*, *update anomaly*, dan *delete anomaly*.
- ✓ Jika perancang *database* cermat dalam melakukan analisis atribut maka dimungkinkan kriteria 5NF langsung terpenuhi pada tahap 3NF/BCNF/4NF.



# Pembahasan Studi Kasus

## Hasil Akhir:

Setelah semua *table* memenuhi kriteria 5NF, selanjutnya beri nama *table* sesuai fungsi kemudian hubungkan seluruh *table* tersebut sehingga membentuk struktur *database* utuh yang akan diimplementasi ke dalam *engine*.



# **Terima Kasih**

Sampai Jumpa di Sesi Berikutnya.....