

# NORMALISASI

Yogiek Indra Kurniawan

Pengantar Basis Data

[yogiek@unsoed.ac.id](mailto:yogiek@unsoed.ac.id)

Universitas Jenderal Soedirman



# APA YANG AKAN KITA PELAJARI?

## PART 1:

1. Definisi Normalisasi
2. Tujuan Normalisasi
3. Konsep – Konsep yang Mendasarinya
  - The Three Keys: Super Key, Candidate Key & Primary Key
  - Functional Dependencies (FD)

## PART 2:

1. Langkah – Langkah Normalisasi
2. Bentuk – Bentuk Normal
  - 1<sup>st</sup> NF, 2<sup>nd</sup> NF, 3<sup>rd</sup> NF, BCNF
  - Dan bentuk-bentuk normal lainnya
3. Denormalisasi

# DEFINISI NORMALISASI

**Normalisasi** adalah langkah-langkah sistematis untuk menjamin bahwa struktur database memungkinkan untuk *general purpose query* dan bebas dari *insertion, update* dan *deletion anomalies* yang dapat menyebabkan hilangnya integritas data.

E.F Codd, 1970

# TUJUAN NORMALISASI

Normalisasi dilakukan terhadap desain tabel yang sudah ada untuk:

- Meminimalkan redundansi (pengulangan) data sehingga desain tabel yang sudah ada menjadi lebih efisien
- Menjamin integritas data dengan cara menghindari 3 Anomali Data:
  - Update Anomaly
  - Insertion Anomaly
  - Deletion Anomaly

# TUJUAN NORMALISASI



# TUJUAN NORMALISASI (CONT'D)

- Deletion Anomaly

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	TE	Elektro	TE-001	Elektronika	3	A
1-01	Tukimin	TE	Elektro	DU-001	English	2	A
2-01	Jamilah	IF	Informatika	IF-001	Algoritma	3	B
2-01	Jamilah	IF	Informatika	DU-001	English	2	C
2-02	Maemunah	IF	Informatika	IF-002	Database	2	A

Jika kita menghapus data mahasiswa bernama Maemunah

Maka kita juga akan **kehilangan data mata kuliah Database**



# TUJUAN NORMALISASI (CONT'D)

- **Update Anomaly**

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	TE	Elektro	TE-001	Elektronika	3	A
1-01	Tukimin	TE	Elektro	DU-001	English	2	A
2-01	Jamilah	IF	Informatika	IF-001	Algoritma	3	B
2-01	Jamilah	IF	Informatika	DU-001	English	2	C
2-02	Maemunah	IF	Informatika	IF-002	Database	2	A

Jika ingin mengupdate jumlah sks mata kuliah English dari 2 menjadi 3 sks, maka harus mengupdate lebih dari 1 record, yaitu baris 2 dan 4

jika hanya salah satu yang diupdate, maka data menjadi tidak konsisten

# TUJUAN NORMALISASI (CONT'D)

- Insertion Anomaly

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	TE	Elektro	TE-001	Elektronika	3	A
1-01	Tukimin	TE	Elektro	DU-001	English	2	A
2-01	Jamilah	IF	Informatika	IF-001	Algoritma	3	B
2-01	Jamilah	IF	Informatika	DU-001	English	2	C
2-02	Maemunah	IF	Informatika	IF-002	Database	2	A

Kita tidak bisa menambah data mahasiswa bernama Zubaedah yang belum mengambil kuliah apapun / belum registrasi

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-02	Zubaedah	TE	Elektro	???	???	???	???





# THE THREE KEYS

Ada 3 macam key dalam sebuah tabel:

- Super Key
- Candidate Key (minimal Super Key)
- Primary Key (chosen Candidate Key)

# THE THREE KEYS (CONT'D)

- **Super Key:** satu atribut atau gabungan atribut (kolom) pada tabel yang dapat membedakan semua baris secara unik.

- Contoh:

Kode_MK	Nama_MK	Semester	SKS
DU-001	English	2	2
DU-002	Kalkulus	1	3
IF-001	Algoritma	1	3
IF-002	Database	2	3
IF-003	Artificial Intelligence	5	2
TE-001	Elektronika	4	3

- **Super key** dari tabel di atas: (kode\_mk) ; (kode\_mk,nama\_mk, semester) ; (kode\_mk,nama\_mk, sks) ; (kode\_mk,nama\_mk, semester, jml\_temu) ; dan lain-lain.
- (sks), (semester) & (semester, sks) **bukan super key (non key)**

# THE THREE KEYS (CONT'D)

- **Candidate Key:** minimal super key, yaitu super key yang tidak mengandung super key yang lain.

- Contoh:

Kode_MK	Nama_MK	Semester	SKS
DU-001	English	2	2
DU-002	Kalkulus	1	3
IF-001	Algoritma	1	3
IF-002	Database	2	3
IF-003	Artificial Intelligence	5	2
TE-001	Elektronika	4	3

Super key: **(kode\_mk)** ; (kode\_mk,nama\_mk, semester) ; (kode\_mk,nama\_mk, sks) ; (kode\_mk,nama\_mk, sks, semester) ; dan lain-lain.

**Candidate key:**(kode\_mk)

# THE THREE KEYS (CONT'D)

- **Primary Key:** salah satu candidate key yang dipilih (dengan berbagai pertimbangan). Tiap tabel hanya memiliki 1 primary key, namun primary key tersebut bisa saja dibentuk dari beberapa atribut (kolom)
- Contoh:

Kode_MK	Nama_MK	Semester	SKS
DU-001	English	2	2
DU-002	Kalkulus	1	3
IF-001	Algoritma	1	3
IF-002	Database	2	3
IF-003	Artificial Intelligence	5	2
TE-001	Elektronika	4	3

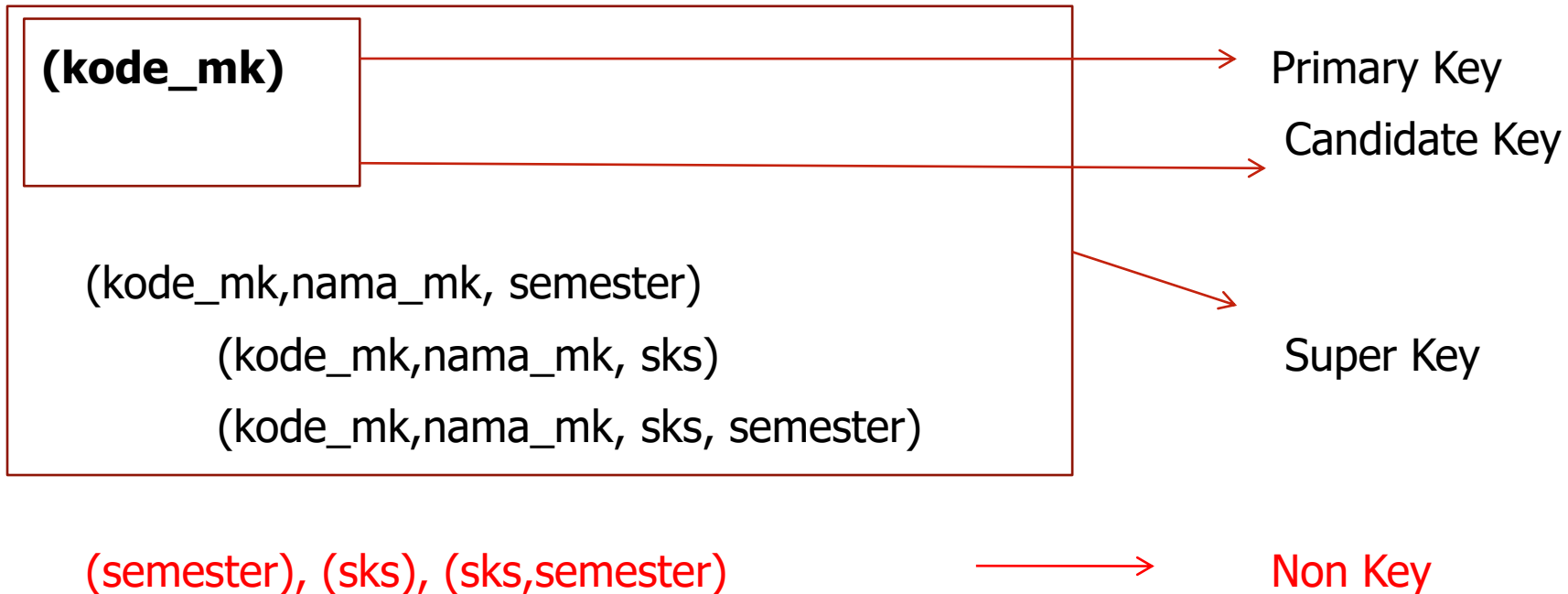
Super key: (kode\_mk) ; (kode\_mk,nama\_mk, semester) ; (kode\_mk,nama\_mk, sks) ; (kode\_mk,nama\_mk, sks, semester) ; dan lain-lain.

Candidate key:(kode\_mk)

**primary key:** (kode\_mk)

# THE THREE KEYS (CONT'D)

Kode_MK	Nama_MK	Semester	SKS
DU-001	English	2	2
DU-002	Kalkulus	1	3
IF-001	Algoritma	1	3
IF-002	Database	2	3



# FUNCTIONAL DEPENDENCIES

**Functional dependency** atau kebergantungan fungsional adalah *constraint* atau batasan/ ketentuan antara 2 buah himpunan atribut pada sebuah tabel

$A \rightarrow B$  berarti

- A menentukan B, atau
- B secara fungsional bergantung kepada A,  
dimana A dan B adalah satu atau sekumpulan atribut dari tabel T

Syarat  $A \rightarrow B$  :

Pada sebuah tabel T, jika ada dua baris data atau lebih dengan nilai atribut A yang sama maka baris-baris data tersebut pasti akan memiliki nilai atribut B yang sama.

Hal ini **tidak** berlaku sebaliknya.



# FUNCTIONAL DEPENDENCIES (CONT'D)

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	TE	Elektro	TE-001	Elektronika	3	A
1-01	Tukimin	TE	Elektro	DU-001	English	2	A
2-01	Jamilah	IF	Informatika	IF-001	Algoritma	3	B
2-01	Jamilah	IF	Informatika	DU-001	English	2	C
2-02	Maemunah	IF	Informatika	IF-002	Database	2	A

Beberapa FD dari tabel di atas:

- FD1: (nim)  $\rightarrow$  (nama\_mhs, kd\_jur, nama\_jur)
- FD2: (kd\_jur)  $\rightarrow$  (nama\_jur)
- FD3: (kode\_mk)  $\rightarrow$  (nama\_mk, sks)
- FD4: (nim, kode\_mk)  $\rightarrow$  (nilai)

# PARTIAL FUNCTIONAL DEPENDENCIES

**Partial Functional Dependency** atau kebergantungan fungsional parsial terjadi bila:

- $B \rightarrow A$
- B adalah bagian dari candidate key

Dengan kata lain

Jika (B,C) adalah *candidate key* dan  $B \rightarrow A$

Maka A bergantung secara parsial terhadap (B,C)  
atau (B,C) menentukan A secara parsial

# FUNCTIONAL DEPENDENCIES (CONT'D)

Partial FD (Contoh):

NIM	Nama_Mhs	Kode_MK	Nilai
1-01	Tukimin	TE-001	A
1-01	Tukimin	DU-001	A
2-01	Jamilah	IF-001	B
2-01	Jamilah	DU-001	C
2-02	Maemunah	IF-002	A

Super key:

(nim,kode\_mk)

(nim,nama\_mhs,kode\_mk)

(nim,nama\_mhs,kode\_mk,nilai)

Candidate key: (nim,kode\_mk)

FD1: (nim)  $\rightarrow$  (nama\_mhs)

FD2: (nim, Kode\_MK)  $\rightarrow$  (Nilai)

**Kesimpulan:**

- (nama\_mhs) bergantung kepada (nim,kode\_mk) secara parsial
- (nim,kode\_mk) menentukan (nama\_mhs) secara parsial

# TRANSITIVE FUNCTIONAL DEPENDENCIES (CONT'D)

## Transitive Functional Dependency

Jika  $A \rightarrow B$  dan  $B \rightarrow C$

Maka  $A \rightarrow C$

Dengan kata lain:

- C bergantung secara transitif terhadap A melalui B
- A menentukan C secara transitif melalui B

# FUNCTIONAL DEPENDENCIES (CONT'D)

- Transitive FD (Contoh):

NIM	Nama_Mhs	Kd_Jur	Nama_Jur
1-01	Tukimin	TE	Elektro
1-01	Tukimin	TE	Elektro
2-01	Jamilah	IF	Informatika
2-01	Jamilah	IF	Informatika
2-02	Maemunah	IF	Informatika

**FD1:** (nim)  $\rightarrow$  (nama\_mhs, **kd\_jur**, nama\_jur)

**FD2:** (**kd\_jur**)  $\rightarrow$  (nama\_jur)

**Kesimpulan:**

- (nama\_jur) bergantung secara transitif terhadap (nim) melalui (kd\_jur)
- (nim)  $\rightarrow$  (nama\_jur) secara transitif melalui (kd\_jur)

# BENTUK NORMAL

**Bentuk Normal** adalah sekumpulan kriteria yang harus dipenuhi oleh sebuah desain tabel untuk mencapai tingkat/level bentuk normal tertentu.

**Parameter** yang biasanya digunakan dalam menentukan kriteria bentuk normal: Functional Dependency & The Three Keys

Makin tinggi level bentuk normal yang dicapai maka:

- kualitas desain tabel tersebut dinyatakan makin baik
- semakin kecil peluang terjadinya anomali dan redundansi data



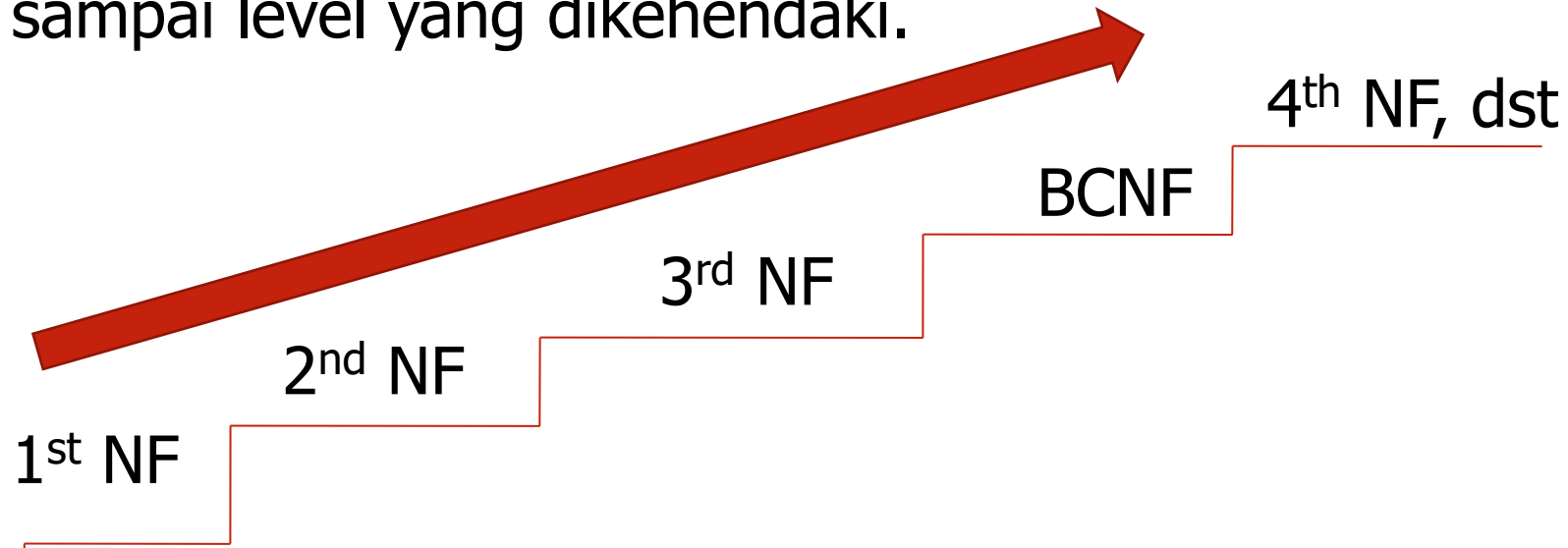
# BENTUK NORMAL (CONT'D)

Beberapa Bentuk Normal yang penting:

- Bentuk Normal Pertama (1<sup>st</sup> Normal Form)  
diperkenalkan oleh Edgar F. Codd pada tahun 1970
- Bentuk Normal Ke-2 (2<sup>nd</sup> Normal Form)  
diperkenalkan oleh Edgar F. Codd pada tahun 1971
- Bentuk Normal Ke-3 (3<sup>rd</sup> Normal Form)  
diperkenalkan oleh Edgar F. Codd, juga pada tahun 1971
- Bentuk Normal Boyce-Codd (BC Normal Form)  
diperkenalkan oleh Raymond F. Boyce & Edgar F. Codd pada tahun 1974

# LANGKAH – LANGKAH NORMALISASI

Menerapkan Bentuk-Bentuk Normal secara bertahap dari level terendah sampai level yang dikehendaki.



Jika mencapai 3<sup>rd</sup> NF atau BCNF maka desain tabel itu biasanya dianggap sudah 'cukup normal'

# 1<sup>ST</sup> NORMAL FORM (1<sup>ST</sup> NF)

## Kriteria 1<sup>st</sup> NF:

- Tidak ada atribut (kolom) pada tabel yang bersifat *multi-value*  
Contoh: kolom telepon yang berisi '0813xx, 022xxx'
- Tidak memiliki lebih dari satu atribut dengan domain yang sama  
Contoh: kolom *telepon1*, *telepon2*, *telepon3* pada tabel mahasiswa

# 1<sup>ST</sup> NF (CONT'D)

## Tabel T Tidak Memenuhi 1<sup>st</sup> NF:

(a) lebih dari 1 atribut dengan domain yang sama:

NIM	Nama_Mhs	Telp_1	Telp_2	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	0813xx	022xxx	TE	Elektro	TE-001	Elektronika	3	A
2-01	Jamilah	0812xx	021xxx	IF	Informatika	IF-001	Algoritma	3	B
2-02	Maemunah	0852xx	031xxx	IF	Informatika	IF-002	Database	2	A

(b) Atribut bersifat multi-value:

NIM	Nama_Mhs	Telepon	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	0813xx, 022xxx	TE	Elektro	TE-001	Elektronika	3	A
2-01	Jamilah	0812xx, 021xxx	IF	Informatika	IF-001	Algoritma	3	B
2-02	Maemunah	0852xx, 031xxx	IF	Informatika	IF-002	Database	2	A

# 1<sup>ST</sup> NF (CONT'D)

Solusi agar memenuhi 1<sup>st</sup> NF: **Dekomposisi Tabel**

Dengan memperhatikan FD **(nim)→(telepon)**

**Tabel T** (*memiliki lebih dari 1 atribut dengan domain yang sama*):

NIM	Nama_Mhs	Telp_1	Telp_2	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
-----	----------	--------	--------	--------	----------	---------	---------	-----	-------

**Atau Tabel T** (*memiliki atribut bersifat multi-value*):

NIM	Nama_Mhs	Telepon	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
-----	----------	---------	--------	----------	---------	---------	-----	-------

Dipecah menjadi 2 tabel berikut:

Tabel T-1:	NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
------------	-----	----------	--------	----------	---------	---------	-----	-------

Tabel T-2:	NIM	Telepon
------------	-----	---------

# 1<sup>ST</sup> NF (CONT'D)

Hasil dekomposisi tabel yang sudah memenuhi 1<sup>st</sup> NF:

**Tabel T-1:**

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	TE	Elektro	TE-001	Elektronika	3	A
2-01	Jamilah	IF	Informatika	IF-001	Algoritma	3	B
2-02	Maemunah	IF	Informatika	IF-002	Database	2	A

**Tabel T-2:**

NIM	Telepon
1-01	0813xx
1-01	022xxx
2-01	0812xx
2-01	021xxx
2-02	0852xx
2-02	031xxx



# 2<sup>ND</sup> NORMAL FORM (2<sup>ND</sup> NF)

Kriteria 2<sup>nd</sup> NF:

- Memenuhi 1<sup>st</sup> NF
- Tidak ada Partial Functional Dependency  
(B,C) adalah *candidate key*  
 $B \rightarrow A$   
A bergantung secara parsial terhadap (B,C)

## 2<sup>ND</sup> NF (CONT'D)

Tabel T-1 sudah memenuhi 1<sup>st</sup> NF tapi tidak memenuhi 2<sup>nd</sup> NF:

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
1-01	Tukimin	TE	Elektro	TE-001	Elektronika	3	A
1-01	Tukimin	TE	Elektro	DU-001	English	2	A
2-01	Jamilah	IF	Informatika	IF-001	Algoritma	3	B
2-01	Jamilah	IF	Informatika	DU-001	English	2	C
2-02	Maemunah	IF	Informatika	IF-002	Database	2	A

(nim, kode\_mk) adalah candidate key

**FD1:** (nim) → (nama\_mhs, kd\_jur, nama\_jur)

**FD2:** (kode\_mk) → (nama\_mk, sks)

**FD3:** (nim, kode\_mk) → nilai

Berarti Terjadi **Partial FD**:

- FD 1: (nim, kode\_mk) → (nama\_mhs, kd\_jur, nama\_jur) secara parsial
- FD 2: (nim, kode\_mk) → (nama\_mk, sks) secara parsial

## 2<sup>ND</sup> NF (CONT'D)

Solusi agar memenuhi 2<sup>nd</sup> NF: **Dekomposisi Tabel**

**Tabel T-1** (terjadi Partial FD pada FD 1 dan FD 2):

NIM	Nama_Mhs	Kd_Jur	Nama_Jur	Kode_MK	Nama_MK	SKS	Nilai
-----	----------	--------	----------	---------	---------	-----	-------

Dipecah (sesuai dengan FD1, FD2 dan FD 3) menjadi 3 tabel berikut:

**Tabel T-1-1:**

NIM	Nama_Mhs	Kd_Jur	Nama_Jur
-----	----------	--------	----------

**Tabel T-1-2:**

Kode_MK	Nama_MK	SKS
---------	---------	-----

**Tabel T-1-3:**

NIM	Kode_MK	Nilai
-----	---------	-------

## 2<sup>ND</sup> NF (CONT'D)

Hasil dekomposisi tabel T-1 yang sudah memenuhi 2<sup>nd</sup> NF:

**Tabel T-1-1:**

NIM	Nama_Mhs	Kd_Jur	Nama_Jur
1-01	Tukimin	TE	Elektro
2-01	Jamilah	IF	Informatika
2-02	Maemunah	IF	Informatika

**Tabel T-1-2:**

Kode_MK	Nama_MK	SKS
TE-001	Elektronika	3
DU-001	English	2
IF-001	Algoritma	3
IF-002	Database	2

**Tabel T-1-3:**

NIM	Kode_MK	Nilai
1-01	TE-001	A
1-01	DU-001	A
2-01	IF-001	B
2-01	DU-001	C
2-02	IF-002	A

# 3<sup>RD</sup> NORMAL FORM (3<sup>RD</sup> NF)

Kriteria 3<sup>rd</sup> NF:

- Memenuhi 2<sup>nd</sup> NF
- Tidak ada Transitive Functional Dependency  
 $A \rightarrow B$   
 $B \rightarrow C$   
C bergantung secara transitif terhadap A melalui B

## 3<sup>RD</sup> NF (CONT'D)

Tabel T-1-1 sudah memenuhi 2<sup>nd</sup> NF tapi tidak memenuhi 3<sup>rd</sup> NF:

NIM	Nama_Mhs	Kd_Jur	Nama_Jur
1-01	Tukimin	TE	Elektro
2-01	Jamilah	IF	Informatika
2-02	Maemunah	IF	Informatika

**FD1:** (nim) → (nama\_mhs, kd\_jur, nama\_jur)

**FD2:** (kd\_jur) → (nama\_jur)

Berarti Terjadi **Transitive FD**:

(nim) → (nama\_jur) secara transitif melalui (kd\_jur)



## 3<sup>RD</sup> NF (CONT'D)

Solusi agar memenuhi 2<sup>nd</sup> NF: **Dekomposisi Tabel**

**Tabel T-1-1** (terjadi Transitive FD):

NIM	Nama_Mhs	Kd_Jur	Nama_Jur
-----	----------	--------	----------

Dipecah (sesuai dengan FD1 dan FD2) menjadi 2 tabel berikut:

**Tabel T-1-1-1:**

NIM	Nama_Mhs	Kd_Jur
-----	----------	--------

**Tabel T-1-1-2:**

Kd_Jur	Nama_Jur
--------	----------

## 3<sup>RD</sup> NF (CONT'D)

Hasil dekomposisi tabel T-1-1 yang sudah memenuhi 3<sup>rd</sup> NF:

**Tabel T-1-1-1:**

NIM	Nama_Mhs	Kd_Jur
1-01	Tukimin	TE
2-01	Jamilah	IF
2-02	Maemunah	IF

**Tabel T-1-1-2:**

Kd_Jur	Nama_Jur
TE	Elektro
IF	Informatika

# BOYCE-CODD NORMAL FORM (BCNF)

Kriteria BCNF:

- Memenuhi 3<sup>rd</sup> NF
- Untuk semua FD yang terdapat di tabel, ruas kiri dari FD tersebut adalah superkey
- Jarang ada kasus dimana tabel yang memenuhi 3<sup>rd</sup> NF tapi tidak memenuhi BCNF

Umumnya sebuah tabel dikategorikan sudah 'cukup normal' jika sudah memenuhi kriteria BCNF

Jika tidak memungkinkan untuk memenuhi kriteria BCNF, maka mencapai 3<sup>rd</sup> NF juga sudah dianggap cukup memadai

## BCNF (CONT'D)

Sejauh ini ada 5 tabel yang dihasilkan dari mulai 1<sup>st</sup> NF – 3<sup>rd</sup> NF:

(Ingat bahwa tabel T-1 & T-1-1 dipecah menjadi tabel-tabel yang lebih kecil)

1. Tabel T-2:

NIM	Telepon
-----	---------

2. Tabel T-1-2:

Kode_MK	Nama_MK	SKS
---------	---------	-----

3. Tabel T-1-3:

NIM	Kode_MK	Nilai
-----	---------	-------

4. Tabel T-1-1-1:

NIM	Nama_Mhs	Kd_Jur
-----	----------	--------

5. Tabel T-1-1-2:

Kd_Jur	Nama_Jur
--------	----------

Semuanya telah memenuhi kriteria 1<sup>st</sup> NF sampai BCNF.

## BENTUK – BENTUK NORMAL LAINNYA

- Bentuk Normal ke-4 (4<sup>th</sup> NF)  
diperkenalkan oleh Ronald Fagin pada tahun 1977
- Bentuk Normal ke-5 (5<sup>th</sup> NF)  
diperkenalkan oleh Ronald Fagin pada tahun 1979
- Domain/Key Normal Form (DKNF)  
diperkenalkan oleh Ronald Fagin pada tahun 1981
- Bentuk Normal ke-6 (6<sup>th</sup> NF)  
diperkenalkan oleh Date, Darwen dan Lorentzos pada tahun 2002


# DENORMALISASI

- **Denormalisasi** : proses menggandakan data secara sengaja (sehingga menyebabkan redundansi data) untuk meningkatkan performa database
- Denormalisasi  $\neq$  Tidak Melakukan Normalisasi
- Denormalisasi dilakukan **setelah** tabel dalam kondisi 'normal' (mencapai level bentuk normal yang diinginkan)
- Salah satu contoh teknik Denormalisasi adalah **Materialized View** pada DBMS Oracle

# DENORMALISASI (CONT'D)

- Alasan melakukan Denormalisasi:
  - Mempercepat proses query dengan cara meminimalkan cost yang disebabkan oleh operasi join antar tabel
  - Untuk keperluan Online Analytical Process (OLAP)
  - Dan lain-lain
- Konsekuensi Denormalisasi:
  - Perlu ruang ekstra untuk penyimpanan data
  - Memperlambat pada saat proses insert, update dan delete sebab proses-proses tersebut harus dilakukan terhadap data yang redundant (ganda)





— THANK YOU —