

NORMALISASI

Mochammad Agri Triansyah, S.Kom., M.Kom

Pengantar Basis Data

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
 - 1st NF, 2nd NF, 3rd 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

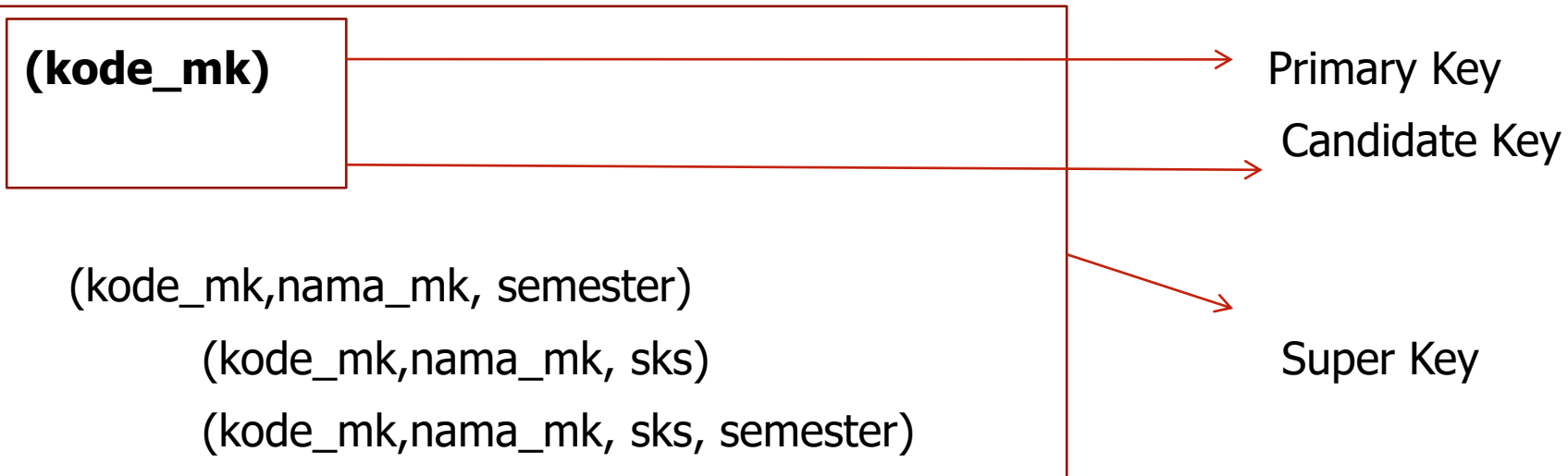
Super key: (kode_mk) ; (kode_mk,nama_mk, semester) ; (kode_mk,nama_mk, sks) ; (kode_mk,nama_mk, sks, semester) ; dan lain-lain.

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



(semester), (sks), (sks, semester)



Non Key

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

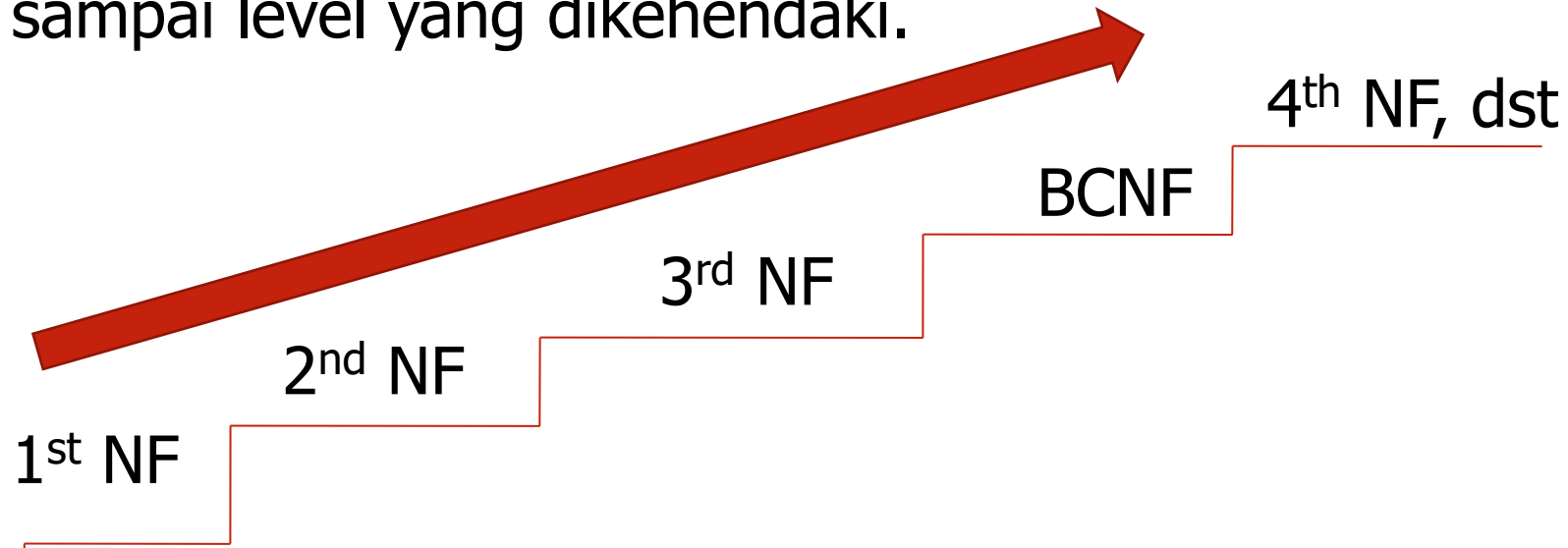
BENTUK NORMAL (CONT'D)

Beberapa Bentuk Normal yang penting:

- Bentuk Normal Pertama (1st Normal Form)
diperkenalkan oleh Edgar F. Codd pada tahun 1970
- Bentuk Normal Ke-2 (2nd Normal Form)
diperkenalkan oleh Edgar F. Codd pada tahun 1971
- Bentuk Normal Ke-3 (3rd 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 3rd NF atau BCNF maka desain tabel itu biasanya dianggap sudah 'cukup normal'

1ST NORMAL FORM (1ST NF)

Kriteria 1st 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

1ST NF (CONT'D)

Tabel T Tidak Memenuhi 1st 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

1ST NF (CONT'D)

Solusi agar memenuhi 1st 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
------------	-----	---------

1ST NF (CONT'D)

Hasil dekomposisi tabel yang sudah memenuhi 1st 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

2ND NORMAL FORM (2ND NF)

Kriteria 2nd NF:

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

2ND NF (CONT'D)

Tabel T-1 sudah memenuhi 1st NF tapi tidak memenuhi 2nd 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

2ND NF (CONT'D)

Solusi agar memenuhi 2nd 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
-----	---------	-------

2ND NF (CONT'D)

Hasil dekomposisi tabel T-1 yang sudah memenuhi 2nd 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

3RD NORMAL FORM (3RD NF)

Kriteria 3rd NF:

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

3RD NF (CONT'D)

Tabel T-1-1 sudah memenuhi 2nd NF tapi tidak memenuhi 3rd 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)

3RD NF (CONT'D)

Solusi agar memenuhi 2nd 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
--------	----------

3RD NF (CONT'D)

Hasil dekomposisi tabel T-1-1 yang sudah memenuhi 3rd 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 3rd NF
- Untuk semua FD yang terdapat di tabel, ruas kiri dari FD tersebut adalah superkey
- Jarang ada kasus dimana tabel yang memenuhi 3rd 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 3rd NF juga sudah dianggap cukup memadai

BCNF (CONT'D)

Sejauh ini ada 5 tabel yang dihasilkan dari mulai 1st NF – 3rd 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 1st NF sampai BCNF.

BENTUK – BENTUK NORMAL LAINNYA


- Bentuk Normal ke-4 (4th NF)
diperkenalkan oleh Ronald Fagin pada tahun 1977
- Bentuk Normal ke-5 (5th NF)
diperkenalkan oleh Ronald Fagin pada tahun 1979
- Domain/Key Normal Form (DKNF)
diperkenalkan oleh Ronald Fagin pada tahun 1981
- Bentuk Normal ke-6 (6th 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 —

