



NORMALISASI

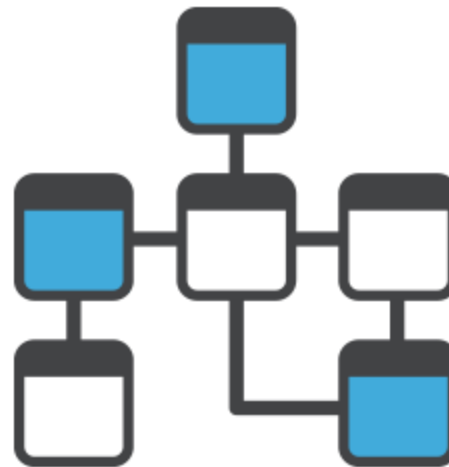
SUBTITLE COMES HERE

TUJUAN PERKULIAHAN:

1. Memahami **konsep normalisasi** dan **ketergantungan fungsional**.
2. Memahami **tahapan normalisasi**.
3. Memahami aturan **normalisasi pertama (1NF)**.
4. Memahami aturan **normalisasi kedua (2NF)**.
5. Memahami aturan **normalisasi ketiga (3NF)**.
6. Memahami aturan **normalisasi Boyce Codd Normal Form (BCNF)**.



1. Konsep Normalisasi Dan Ketergantungan Fungsional



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 tergantung pada NIM)

$nama_kul, nim \rightarrow indeks_nilai$
(nilai tergantung pada NIM, nama_kul)

Non FD

$nama_kul \rightarrow nim$

$nim \rightarrow indeks_nilai$

- 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).

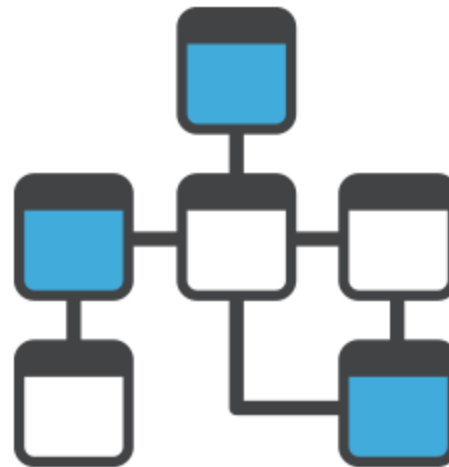
➤ **$\text{nama_kul} \rightarrow \text{nim}$**

➤ **$\text{nim} \rightarrow \text{indeks_nilai}$**

CONTOH FD 1

- Andaikan ada tabel:
NILAI (NIM, Nm-mk, Semester, Nilai)
- Atribut kunci: NIM, Nm-mk, Semester
- Maka Functional Dependency:
NIM, Nm-mk, Semester \rightarrow Nilai

2. Tahapan Normalisasi



TAHAPAN NORMALISASI

- Dalam normalisasi terdapat beberapa tahapan yang saling terkait.
- Tahap Normalisasi dimulai dari tahap **paling ringan (1NF)** hingga **paling ketat (5NF)**
- Normalisasi yang sering dilakukan hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.

KRITERIA TABEL NORMAL

Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria sbb:

1. Jika ada dekomposisi (penguraian) tabel, maka dekomposisinya harus dijamin aman (***Lossless-Join Decomposition***). Artinya, setelah tabel tersebut diuraikan / didekomposisi menjadi tabel-tabel baru, tabel-tabel baru tersebut bisa menghasilkan tabel semula dengan sama persis.
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (**Dependency Preservation**).
3. Tidak melanggar **Boyce-Codd Normal Form (BCNF)** (-akan dijelaskan kemudian-) Jika BCNF tidak terpenuhi, maka paling tidak suatu tabel tidak melanggar bentuk normal tahap ketiga

CONTOH RELASI NORMAL DAN TIDAK NORMAL

relasi tidak normal (Tabel Personil)

ID_Personil	Tanggal Lahir	Karakteristik
I102	17 Januari 1970	Tinggi 162
		Berat 50
		Rambut hitam
A212	12 Desember 1966	Tinggi 170
		Berat 64
		Rambut hitam

Relasi Normal (Tabel Personil)

ID Personil	Tanggal lahir	Tinggi	Berat	Warna Rambut
I102	17 Januari 1970	162	50	Hitam
A212	12 Desember 1966	170	64	Hitam

TABEL *UNIVERSAL*

Tabel Universal (*Universal / Star Table*) → sebuah tabel yang merangkum semua kelompok data yang saling berhubungan, bukan merupakan tabel yang baik.

nim	nama_mhs	alamat_mhs	tgl_lahir	kode_kul	nama_kul	sks
163010015	Betha Susanti	Jl. Mawar III/no, 1a	02-Jun-97	BD-111	Basisdata	3
163010015	Betha Susanti	Jl. Mawar III/no, 1a	02-Jun-97	MT-111	Matematika	2
163010025	Kyla Nuri M.	Jl. Melati Timur I/no. 15f	06-Jan-97	BG-111	Bahasa inggris	2
163010033	Mega Rinasa	Jl. Gajayana IV/no. 9	17-Okt-97	IM-111	IMK	3
163010033	Mega Rinasa	Jl. Gajayana IV/no. 9	17-Okt-97	MT-111	Matematika	2
163010035	Tera Akbar	Jl. Majapahit VII/no. 11	25-Mar-97	BD-111	Basisdata	3

Lanjutan
tabel

semester	indeks_nilai	waktu	tempat	nama_dos	alamat_dos
2	A	Rabu, 12.30-16.00 dan Kamis, 08.00-12.30	LBD.1	Dr. Manaf Putra	Jl. Sulfat Timur I/no. 7A
1		Senin, 12.30-15.00 dan Kamis, 08.00-11.30	KR.01	Dr. Maryam Harlina	Jl. Ciliwung III/no.6
1	B	Selasa 08.00-11.30 dan Jumat, 08.00-11.30	KR.04	Dr. Ristianto Arif	Jl. Merdeka I/ no.1
2		Kamis, 08.00-12.30 dan Jumat, 08.00-12.30	KR.06	Dr. Maya Rista	Jl. Kalimantan IV/no 5a
1	C	Senin, 12.30-15.00 dan Kamis, 08.00-11.30	KR.01	Dr. Maryam Harlina	Jl. Ciliwung III/no. 6
2	A	Rabu, 12.30-16.00 dan Kamis, 08.00-12.30	LBD.1	Dr. Manaf Putra	Jl. Sulfat Timur I/no. 7A

LOSSLESS-JOIN DECOMPOSITION

- Dalam bahasa Indonesia Lossless-Join decomposition dalam bahasa Indonesia dapat diartikan **Dekomposisi Aman**
- Dekomposisi aman adalah kondisi apabila tabel-tabel hasil dekomposisi digabungkan kembali dapat menghasilkan tabel awal sebelum di dekomposisi.
- Dekomposisi yang tidak aman dapat berasal dari FD yang diperoleh dari asumsi yang kurang tepat

DEKOMPOSISI AMAN DAN TIDAK AMAN

Dekomposisi Tidak Aman (Lossy-Join Decomposition)

Contoh :

- Terdapat tabel ABC dengan FD :
 $A \rightarrow B$ dan $B \rightarrow C$

A	B	C
a1	100	c1
a2	200	c2
a3	300	c3
a4	200	c4

A	B
a1	100
a2	200
a3	300
a4	200

B	C
100	c1
200	c2
300	c3
200	c4

Hasil Penggabungan Kembali		
A	B	C
a1	100	c1
a2	200	c2
a2	200	c4
a3	300	c3
a4	200	c2
a4	200	c4

Dekomposisi Aman (Lossless-Join Decomposition)

Contoh :

- Terdapat tabel ABC dengan FD :
 $A \rightarrow B$ dan $B \rightarrow C$

A	B	C
a1	100	c1
a2	200	c2
a3	300	c3
a4	200	c2

A	B
a1	100
a2	200
a3	300
a4	200

B	C
100	c1
200	c2
300	c3

Hasil Penggabungan Kembali		
A	B	C
a1	100	c1
a2	200	c2
a3	300	c3
a4	200	c2

DEPENDENCY PRESERVATION

- **Dependency preservation** dapat diartikan sebagai pemeliharaan ketergantungan.
- Ketika terjadi perubahan terhadap suatu basisdata, maka seharusnya dapat dijamin perubahan tersebut tidak menghasilkan inkonsistensi data yang mengakibatkan FD sudah benar menjadi tidak terpenuhi.
- **Contoh :**
 - Pada tabel universal slide sebelumnya jika terjadi perubahan **tidak efisien** pada data mahasiswa dengan nim '163010015' maka perubahan harus dilakukan pada alamat_mhs di semua baris nim tersebut.
 - Perubahan yang **efisien** terdapat pada tabel hasil dekomposisi **tabel mahasiswa{nim, nama_mhs, alamat_mhs, tgl_lahir}** dan **nilai{nama_kul, nim, nama_mhs, indeks_nilai}**.
 - Jika pada tabel mahasiswa terjadi perubahan data pada atribut alamat_mhs, maka perubahan tidak perlu dijalankan ke tabel 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)

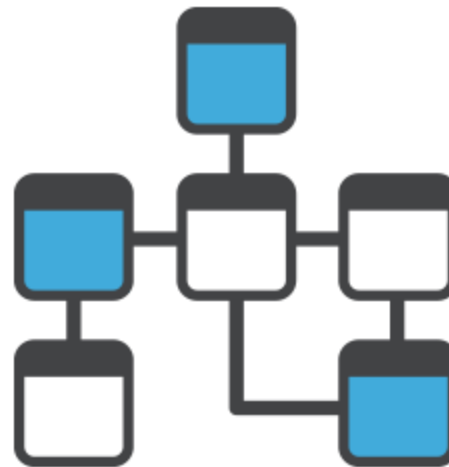
↓ Menghilangkan ketergantungan Multivalue

Bentuk Normal Keempat (4NF)

↓ Menghilangkan anomali-anomali yang tersisa

Bentuk Normal Kelima

3. NORMALISASI PERTAMA (1 NF)



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-VALUE)

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)

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...)

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

CONTOH 2 (COMPOSITE)

JadwalKuliah

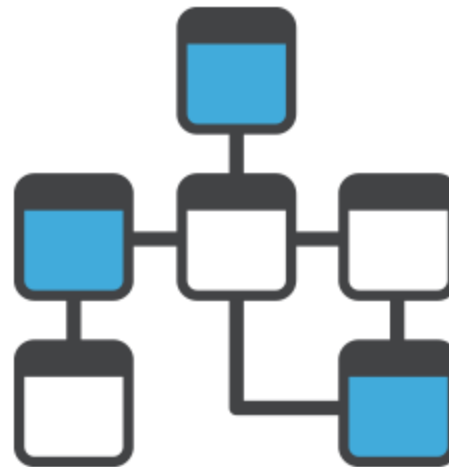
Kodekul	NamaKul	Dosen	Kelas	Jadwal
---------	---------	-------	-------	--------

- 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:

JadwalKuliah

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

4. NORMALISASI KEDUA (2 NF)



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

CONTOH 2NF (2)

Part	Warehouse	Quantity	WarehouseAddress
42	Boston	2000	24 Main St
333	Boston	1000	24 Main St
390	New York	3000	99 Broad St

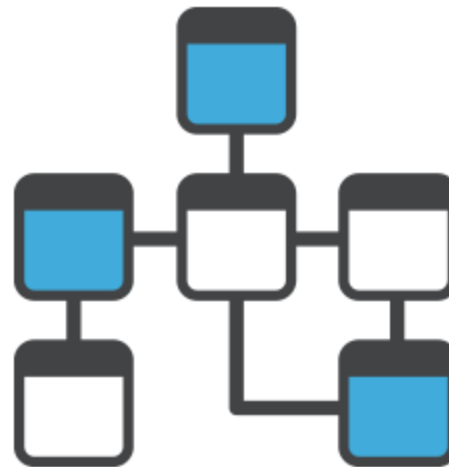
- Semua atribut bukan key primer hanya boleh memiliki ketergantungan (functional dependency) pada atribut key primer
 - **Contoh yang melanggar bentuk normal kedua:**
 - Key : part + warehouse
 - WarehouseAddress adalah penjelas tentang Warehouse, bukan tentang Part
 - **Problems:**
 - WarehouseAddress diulang di setiap baris mengacu pada part yang disimpan
 - Jika warehouseAddress berubah, setiap baris yang merujuk ke bagian yang disimpan di gudang tersebut harus diperbarui.
 - Jika pada suatu saat tidak ada part yang disimpan di gudang, mungkin tidak ada catatan untuk menyimpan WarehouseAddress.
 - Data mungkin menjadi tidak konsisten, dengan baris berbeda menunjukkan alamat berbeda untuk gudang yang sama.

CONTOH 2NF (2)

- Solution
- Buat Dua jenis tabel: Inventory dan Warehouse
- Keuntungan: memecahkan masalah dari slide sebelumnya

Part	Warehouse	Quantity	Warehouse	WarehouseAddress
42	Boston	2000	Boston	24 Main St
333	Boston	1000	New York	99 Broad St
390	New York	3000		

5. NORMALISASI KETIGA (3 NF)



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** **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.**

LATIHAN 1 : NORMALISASI DATA

NORMALISASIKAN TABLE BERIKUT

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

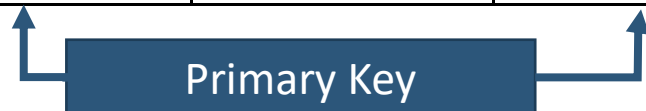
Untuk mendapatkan hasil yang paling normal, maka proses normalisasi dimulai dari normal pertama.

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



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>Noprojek</u>	NamaProyek
NP001	BRR
NP002	PEMDA

TABEL PEGAWAI

<u>Nopegawai</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

LATIHAN 1 : NORMALISASI DATA

NORMALISASIKAN TABLE BERIKUT

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 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

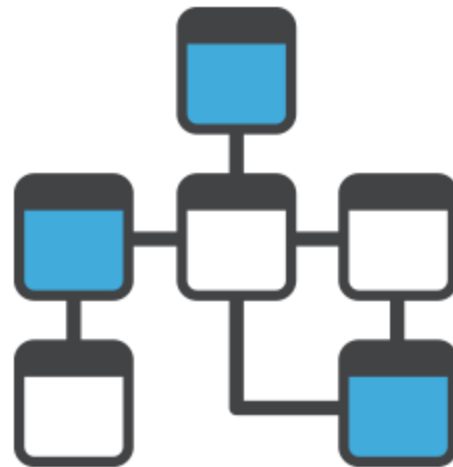
Tabel Dosen

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

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

6. 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 :

$X \twoheadrightarrow A$ 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 : $A \twoheadrightarrow B$ dan $B \twoheadrightarrow C$ maka R bukan BCNF, sebab :

- A superkey ?
 $A \twoheadrightarrow B$ (diketahui)
 $A \twoheadrightarrow B$ dan $B \twoheadrightarrow C$ maka $A \twoheadrightarrow C$ (transitif)
 $A \twoheadrightarrow A$ (refleksif)
Sehingga $A \twoheadrightarrow (A,B,C)$ atau $A \twoheadrightarrow R$. Jadi A superkey.
- B superkey ?
 $B \twoheadrightarrow C$ (diketahui)
 $B \twoheadrightarrow B$ (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 : $A \twoheadrightarrow B$ dan

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

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

Contoh :

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

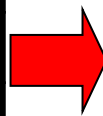
dengan FD : $AB \twoheadrightarrow C$ dan $C \twoheadrightarrow B$. Apakah :

- 3NF ?
- BCNF ?
- R memenuhi 3NF karena :
 $AB \twoheadrightarrow C$; maka $AB \twoheadrightarrow ABC$, atau $A \twoheadrightarrow R$. Jadi AB superkey dari R
 $C \twoheadrightarrow B$; maka $AC \twoheadrightarrow AB$, atau $AC \twoheadrightarrow ABC$ dan $AC \twoheadrightarrow R$.
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 \square 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



<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

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

NOT BCNF

BCNF

SOAL

- Normalisasikan table berikut

Location code	Location name	Plant code	Plant name	Soil category	Soil description
11	Kirstenbosch Gardens	431	Leaucadendron	A	Sandstone
		446	Protea	B	Sandstone/limestone
		482	Erica	C	Limestone
12	Karbonkelberg Mountains	431	Leucadendron	A	Sandstone
		449	Restio	B	Sandstone/limestone



JURUSAN
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG



THANK YOU!



+62 (0341) 404424 – 404425



[HTTPS://JTI.POLINEMA.AC.ID/](https://jti.polinema.ac.id/)

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.