

Normalisasi Data

Rizka Hadiwiyanti, S.Kom, M.Kom



Key

- Key adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam tabel secara unik
- Tiga macam key:
 - Superkey
 - Candidate key
 - Primary key

Superkey

- Merupakan satu atau lebih atribut (kumpulan atribut) yang dapat membedakan setiap baris data dalam tabel secara unik
- Contoh, di tabel Mahasiswa, atribut yang dapat menjadi superkey adalah
 - Nim, nama_mhs, alamat_mhs, tgl_lahir
 - Nim, nama_mhs, alamat_mhs
 - Nim, nama_mhs
 - nama_mhs (jika bisa dijamin tidak ada nilai yang sama)
 - Nim

Candidate key

- Merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik
- tidak boleh berisi atribut atau kumpulan atribut yang telah menjadi superkey yang lain
- Contoh yang bisa menjadi candidate key
 - Nim
 - nama_mhs (jika bisa dijamin tidak ada nilai yang sama)

Candidate key dan Primary key

- Candidate key dapat dijadikan sebagai primary key dengan dasar sebagai berikut:
 - Key tersebut lebih sering (lebih natural) untuk dijadikan sebagai acuan
 - Key tersebut lebih ringkas
 - Jaminan keunikan key tersebut lebih baik
- Dengan dasar tersebut, dari kedua candidate key pada tabel mahasiswa (nim dan nama_mhs), yang lebih cocok dipilih sebagai primary key adalah nim

Atribut sederhana dan atribut komposit

- Atribut sederhana: atribut yang tidak dapat dipilah lagi
- Atribut komposit: atribut yang masih dapat diuraikan lagi menjadi sub atribut

nim	nama_mhs	alamat_mhs	tgl_lahir
...	Ali Akbar	Jl. Dago Pojok 91, Bandung 40135	...
...	Budi Haryanto	Jl. Pasantren 25D, Cimahi 40533	...
...

dapat diuraikan menjadi 3 atribut

alamat	kota	kode_pos
Jl. Dago Pojok 91	Bandung	40135
Jl. Pasantren 25D	Cimahi	40533
...

Gambar 4.1 - Atribut Sederhana dan Atribut Komposit

Atribut bernilai tunggal dan atribut bernilai banyak

- Atribut bernilai tunggal: ditujukan pada atribut yang memiliki paling banyak satu nilai untuk setiap baris data
- Atribut bernilai banyak: ditujukan pada atribut yang dapat kita isi dengan lebih dari satu nilai, tetapi jenis datanya sama.

Nim	nama_mhs	alamat_mhs	tgl_lahir	hobby
100001	sepak bola melukis berenang
100002	membaca
100003	
100004	musik, menari

Mandatory attribute dan Nilai Null

- Mandatory attribute: atribut yang HARUS mempunyai nilai atau berisi data
- Atribut yang nilai atau datanya boleh dikosongkan dinyatakan dengan nilai NULL

The diagram shows a table with five columns: **nim**, **nama_mhs**, **alamat_mhs**, **tgl_lahir**, and **hobby**. The first three columns are labeled as *Mandatory Attribute*, while the last two are labeled as *Non Mandatory Attribute*. The table contains five rows of data. The first row is a new data row. The second and third rows have null values in the **alamat_mhs** and **hobby** columns. The fourth row has null values in the **alamat_mhs** and **hobby** columns. The fifth row has null values in the **alamat_mhs** and **hobby** columns.

nim	nama_mhs	alamat_mhs	tgl_lahir	hobby
...
100002	membaca
100003	
100004	musik, menari
100005	Abdullah			naik gunung

Mandatory Attribute (points to nim, nama_mhs, alamat_mhs)

Non Mandatory Attribute (points to tgl_lahir, hobby)

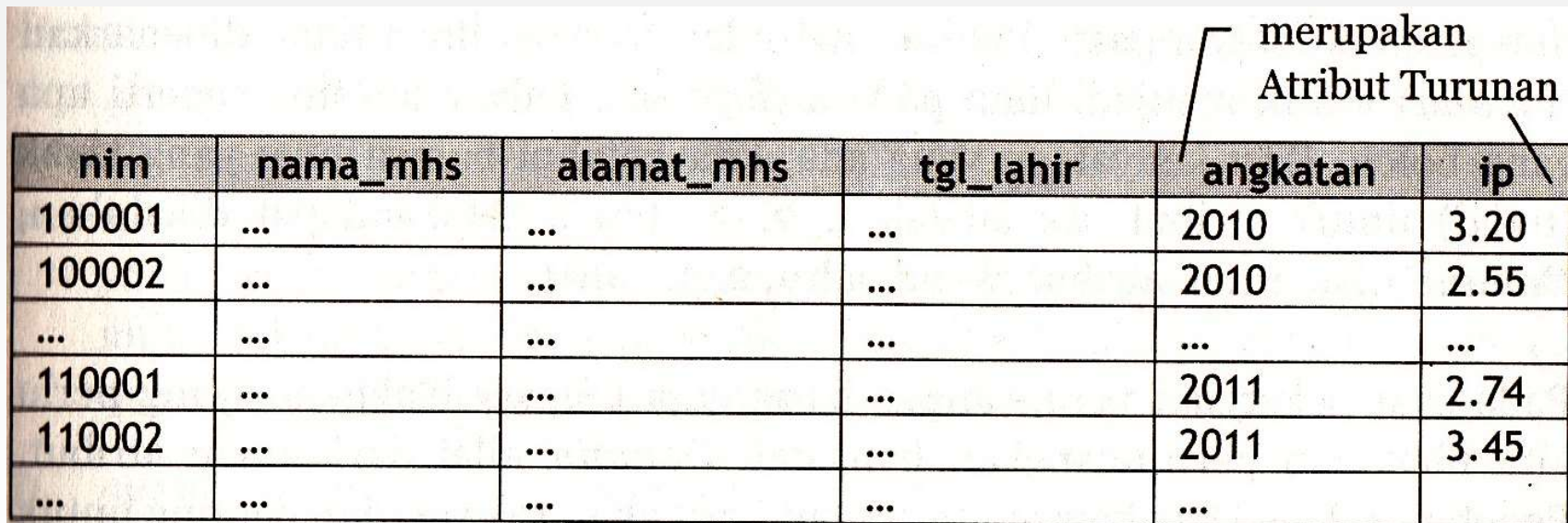
baris data baru (points to the first row)

berisi Null, karena datanya belum siap/masih meragukan (points to the null values in the second and third rows)

berisi Null, karena memang tidak punya hobby (points to the null value in the fourth row)

Atribut turunan

- Atribut yang nilai-nilainya diperoleh dari pengolahan data atau diturunkan dari atribut atau tabel lain yang berhubungan



nim	nama_mhs	alamat_mhs	tgl_lahir	angkatan	ip
100001	2010	3.20
100002	2010	2.55
...
110001	2011	2.74
110002	2011	3.45
...

Gambar 4.4 - Atribut Turunan

Anomali

- Anomali basis data merupakan penyimpangan yang diakibatkan dari suatu proses pembentukan tidak normal suatu struktur tabel, sehingga mengakibatkan inkonsistensi data
- Tujuan konsep basis data relasional → menghilangkan anomali yang terjadi
- Anomali dapat terjadi selama adanya perubahan dalam suatu basis data, yang dapat merusak data secara logika
- Jenis anomali: anomali *insert*, anomali *delete*, anomali *update*

Anomali *insert*

- Terjadi pada *record* yang memiliki value kosong pada suatu *field*, dimana value kosong tersebut berperan sebagai primary key
- Contoh: terdapat tabel yang berisi data pasien rumah sakit yang melakukan terapi pengobatan

<u>Kodepasien</u>	Terapi	Dokter	Biaya
kp3	cikungunya	Pram	Rp 400.000
kp1	migrain	Rizal	Rp 250.000
kp2	migrain	Rizal	Rp 250.000
kp4	asam urat	Yudi	Rp 500.000

Anomali *insert* (cont'd)

- Anomali terjadi pada saat rumah sakit akan mengadakan terapi untuk penyakit lain, misalnya terapi obesitas.

<u>Kodepasien</u>	Terapi	Dokter	Biaya
kp3	cikungunya	Pram	Rp 400.000
kp1	migrain	Rizal	Rp 250.000
kp2	migrain	Rizal	Rp 250.000
kp4	asam urat	Yudi	Rp 500.000
	obesitas	Yudo	Rp 450.000

- Anomali insert terjadi karena terdapat value yang kosong pada field Kodepasien, sehingga value “obesitas”, “Yudo”, dan “Rp 450.000” tidak dapat dimasukkan pada tabel tersebut

Anomali *delete*

- Terjadi pada satu record yang memiliki value untuk dihapus, namun mengakibatkan data lain dalam record tersebut ikut terhapus
- Contoh: kodepasien kp3 akan dihapus, karena pasien tersebut dinyatakan sehat oleh dokter yang memeriksanya.

<u>Kodepasien</u>	Terapi	Dokter	Biaya
kp3	cikungunya	Pram	Rp 400.000
kp1	migrain	Rizal	Rp 250.000
kp2	migrain	Rizal	Rp 250.000
kp4	asam urat	Yudi	Rp 500.000

- Efek dari penghapusan tersebut juga menghilangkan data lainnya (cikungunya, Pram, dan Rp 400.000), sehingga data rekam medis untuk pasien tersebut terhapus semua

Anomali *update*

- Terjadi akibat operasi pengubahan value pada satu record yang memiliki kesamaan dengan record lainnya, sehingga terdapat data yang tidak konsisten untuk beberapa record yang sama
- Contoh: biaya untuk jenis terapi migrain di-*update* dari Rp 250.000 menjadi Rp 225.000. Efek update hanya berakibat pada kodepasien kp2 saja.

<u>Kodepasien</u>	Terapi	Dokter	Biaya
kp3	cikungunya	Pram	Rp 400.000
kp1	migrain	Rizal	Rp 250.000
kp2	migrain	Rizal	Rp 225.000
kp4	asam urat	Yudi	Rp 500.000
kp5	migrain	Rizal	Rp 250.000

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.

Proses Normalisasi

- Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat.
- Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

Functional Dependency (Ketergantungan Fungsional)

- Dinyatakan dengan notasi $A \rightarrow B$ dimana A dan B adalah atribut atau field dari sebuah tabel
- Artinya, secara fungsional A menentukan B atau B tergantung pada A, jika dan hanya jika ada 2 baris data dengan nilai A yang sama, maka nilai B juga sama

Functional Dependency

	nama_kul	nim	nama_mhs	indeks_nilai
row 1	Struktur Data	100001	Ali Akbar	A
row 2	Struktur Data	100004	Indah Susanti	B
row 3	Basis Data	100001	Ali Akbar	
row 4	Basis Data	110002	Budi Haryanto	
row 5	Basis Data	110004	Indah Susanti	
row 6	Bahasa Indonesia	100001	Ali Akbar	B
row 7	Matematika I	100002	Budi Haryanto	C

- FD yang dapat ditentukan adalah:

- Nim \rightarrow nama_mhs

nama_mhs hanya tergantung pada atribut nim

- Nama_kul, nim \rightarrow indeks_nilai

indeks_nilai tergantung pada atribut nama_kul dan nim secara bersamaan

Ketidaktergantungan fungsional

	nama_kul	nim	nama_mhs	indeks_nilai
row 1	Struktur Data	100001	Ali Akbar	A
row 2	Struktur Data	100004	Indah Susanti	B
row 3	Basis Data	100001	Ali Akbar	
row 4	Basis Data	110002	Budi Haryanto	
row 5	Basis Data	110004	Indah Susanti	
row 6	Bahasa Indonesia	100001	Ali Akbar	B
row 7	Matematika I	100002	Budi Haryanto	C

- Nama_kul \nrightarrow nim
 - nim tidak tergantung pada nama_kul
- Nim \nrightarrow indeks_nilai
 - Indeks_nilai tidak tergantung pada nim

Penerapan Normalisasi

nim	nama_mhs	alamat_mhs	tgl_lahir	kode_kul	nama_kul
100001	Ali Akbar	Jl. Dago Poj ...	2 Jan 1992	IF-110	Struktur Data
100001	Ali Akbar	Jl. Dago Poj ...	2 Jan 1992	IF-310	Basis Data
100001	Ali Akbar	Jl. Dago Poj ...	2 Jan 1992	KU-234	Bahasa Indonesia
100002	Budi Haryanto	Jl. Pasantre ...	6 Okt 1991	IF-310	Basis Data
100002	Budi Haryanto	Jl. Pasantre ...	6 Okt 1991	MA-115	Matematika I
100004	Hasanudin	Jl. Titiran ...	21 Jun 1992	IF-310	Basis Data
100004	Hasanudin	Jl. Titiran ...	21 Jun 1992	IF-110	Struktur Data

disambung dengan kolom

sks	semester	indeks_nilai	waktu	tempat
3	1	A	Senin, 08:00-09:40 dan Kamis, 11:00-11:50	Ruang A
4	3		Selasa, 10:00-11:40 dan Jum'at, 08:00-09:40	Ruang B
2	2	B	Rabu, 09:00-10:40	Ruang A
4	3		Selasa, 10:00-11:40 dan Jum'at, 08:00-09:40	Ruang B
3	1	C	Rabu, 13:00-14:40 dan Jum'at, 14:00-14:50	Ruang C
4	3		Selasa, 10:00-11:40 dan Jum'at, 08:00-09:40	Ruang B
3	1	B	Senin, 08:00-09:40 dan Kamis, 11:00-11:50	Ruang A

disambung dengan kolom

nama_dos	alamat_dos
Dr. Umar Hakim	Cibaduyut Kidul No.57C, Bandung 40239
Dr. Umar Hakim	Cibaduyut Kidul No.57C, Bandung 40239
Dewi Sundari, MA	Jl. Antapani No.14, Bandung 40291
Dr. Umar Hakim	Cibaduyut Kidul No.57C, Bandung 40239
Ir. Taufik Ismail	Kanayakan Baru No. 135, Bandung 40135
Dr. Umar Hakim	Cibaduyut Kidul No.57C, Bandung 40239
Dr. Umar Hakim	Cibaduyut Kidul No.57C, Bandung 40239

- Tabel berikut adalah tabel universal yang merangkum semua data yang saling berhubungan
- Tentukan functional dependency yang bisa dibentuk!

Dari tabel universal tersebut, functional dependency yang terbentuk adalah

- Nim → nama_mhs, alamat_mhs, tgl_lahir
- Kode_kul → nama_kul, sks, semester, waktu, tempat, nama_dos
- Nama_dos → alamat_dos
- Nim, kode_kul → indeks_nilai

Kelemahan tabel universal

- Pengulangan informasi (redundansi atau duplikasi data)
- Potensi inkonsistensi data jika ada data yang diubah
- Tersembunyinya informasi tertentu

- Kelemahan tersebut bisa dihilangkan dengan melakukan Dekomposisi, yaitu pemilahan tabel menjadi beberapa tabel

Normalisasi dengan Functional Dependency

- Sebuah tabel dikatakan baik (efisien) atau normal jika memenuhi 3 kriteria minimal 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)
- Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF).



Lossless-Join Decomposition (Dekomposisi aman)

- Dekomposisi yang benar terjadi jika tabel-tabel hasil dekomposisi digabungkan kembali, dapat menghasilkan tabel awal sebelum dekomposisi
- Ex: tabel ABC didefinisikan dengan 2 FD: $A \rightarrow B$ dan $B \rightarrow C$

A	B	C
A1	100	C1
A2	200	C2
A3	300	C3
A4	200	C4

Didekomposisi menjadi



A	B
A1	100
A2	200
A3	300
A4	200

B	C
100	C1
200	C2
300	C3
200	C4

Digabungkan kembali



A	B	C
A1	100	C1
A2	200	C2
A2	200	C4
A3	300	C3
A4	200	C2
A4	200	C4

Lossy-Decomposition
(dekomposisi tidak aman)



PresentationPoint



Dependency Preservation (pemeliharaan ketergantungan)

- Ketika melakukan perubahan data, pastikan agar perubahan tersebut tidak menghasilkan inkonsistensi data yang mengakibatkan FD yang sudah benar menjadi tidak terpenuhi

nama_kul	nim	nama_mhs	indeks_nilai
Struktur Data	100001	Ali Akbar	A
Struktur Data	100004	Indah Susanti	B
Basis Data	100001	Ali Akbar	
Basis Data	110002	Budi Haryanto	
Basis Data	110004	Indah Susanti	
Bahasa Indonesia	100001	Ali Akbar	B
Matematika I	100002	Budi Haryanto	C

- Apa pengaruhnya terhadap tabel Nilai jika atribut alamat_mhs diubah?
- Apa pengaruhnya terhadap tabel Nilai jika atribut nama_mhs diubah?
- Solusi: Menghapus atribut nama_mhs dari tabel Nilai

Boyce-Code Normal Form (BCNF)

- Tabel dikatakan dalam Boyce-Code Normal Form (BCNF) jika untuk semua functional dependency dengan notasi $X \rightarrow Y$, maka X harus merupakan superkey pada tabel tersebut
- Contoh: tabel universal didekomposisi menjadi 4 tabel sesuai dengan ke-4 FD sebelumnya
 - Mahasiswa = {nim, nama_mhs, alamat_mhs, tgl_lahir}
 - Kuliah = {kode_kul, nama_kul, sks, semester, waktu, tempat, nama_dos}
 - Dosen = {nama_dos, alamat_dos}
 - Nilai = {nim, kode_kul, indeks_nilai}

Tabel Mahasiswa

nim	nama_mhs	alamat_mhs	tgl_lahir
100001	Ali Akbar	Jl. Dago Pojok 91, Bandung 40135	2 Jan 1992
100002	Budi Haryanto	Jl. Pasantren 25D, Cimahi 40533	6 Okt 1991
100004	Hasanudin	Jl. Titiran No. 2, Bandung 40133	21 Jun 1992

- Functional Dependency: $nim \rightarrow nama_mhs, alamat_mhs, tgl_lahir$
- Tabel tersebut sudah memenuhi BCNF karena sisi X (nim) merupakan superkey

Tabel Kuliah

kode_kul	nama_kul	sks	semester	waktu	tempat	nama_dos
IF-110	Struktur Data	3	1	Senin, 08:00-09:40 dan Kamis, 11:00-11:50	Ruang A	Dr. Umar Hakim
IF-310	Basis Data	4	3	Selasa, 10:00-11:40 dan Jum'at, 08:00-09:40	Ruang B	Dr. Umar Hakim
KU-234	Bahasa Indonesia	2	2	Rabu, 09:00-10:40	Ruang A	Dewi Sundari, MA
MA-115	Matematika I	3	1	Rabu, 13:00-14:40 dan Jum'at, 14:00-14:50	Ruang C	Drs. Taufik Ismail

- Functional Dependency: kode_kul \rightarrow nama_kul, sks, semester, waktu, tempat, nama_dos
- Tabel tersebut sudah memenuhi BCNF karena sisi kode_kul merupakan superkey

Tabel Dosen dan Tabel Nilai

■ Tabel dosen

nama_dos	alamat_dos
Drs. Taufik Ismail	Perum. Dosen Griya Indah C-5, Bogor 43111
Dewi Sundari, MA	Jl. Selamat No.14, Jakarta 40121
Dr. Umar Hakim	Jl. Garuda No.1, Depok 43212

■ Tabel Nilai

kode_kul	nim	indeks_nilai
IF-110	100001	A
IF-110	100004	B
IF-310	100001	
IF-310	100002	

- Apakah kedua tabel tersebut sudah memenuhi BCNF?

Jika kriteria ketiga (BCNF) tidak dapat terpenuhi, maka paling tidak tabel tersebut tidak melanggar Bentuk Normal tahap ketiga (3rd Normal Form / 3NF).

Bentuk-bentuk Normal

- Bentuk Normal Tahap Pertama (1st Normal Form / 1NF)
- Bentuk Normal Tahap Kedua (2nd Normal Form / 2NF)
- Bentuk Normal Tahap (3rd Normal Form / 3NF)
- Boyce-Code Normal Form (BCNF)
- Bentuk Normal Tahap (4th Normal Form / 4NF)
- Bentuk Normal Tahap (5th Normal Form / 5NF)

Normal Pertama (1st Normal Form)

- Aturan:

- Tidak adanya **atribut multi-value**, **atribut komposit** atau kombinasinya
- Mendefinisikan atribut kunci.
- Setiap atribut dalam tabel tersebut harus bernilai *atomic* (tidak dapat dibagi-bagi lagi)

- Contoh: terdapat tabel Mahasiswa sebagai berikut

<u>Nrp</u>	<u>nama</u>	<u>Hobi</u>
12020001	Heri Susanto	Sepakbola, membaca komik, berenang
12020013	Siti Zulaiha	Memasak, mrogram komputer
12020015	Dini Susanti	Menjahit, membuat roti

- Tabel-tabel tersebut

tidak memenuhi syarat 1NF

<u>Nrp</u>	<u>nama</u>	<u>hobi1</u>	<u>hobi2</u>	<u>Hobi3</u>
12020001	Heri Susanto	Sepak Bola	Membaca komik	berenang
12020013	Siti Zulaiha	Memasak	mrogram komputer	
12020015	Dini Susanti	Menjahit	membuat kue	

- Tabel-tabel tersebut didekomposisi menjadi

- 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)

Jadwal Kuliah

Kodekul	NamaKul	Dosen	Kelas	Jadwal
A001	Database	Bu Risqa	305	Senin 07.00
A002	IMK	Pak Pandu	205	Selasa 09.30

- 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:

Kodekul	NamaKul	Dosen	Kelas	JadwalHari	JadwalJam
A001	Database	Bu Risqa	305	Senin	07.00
A002	IMK	Pak Pandu	205	Selasa	09.30

Normalisasi Kedua (2nd Normal Form)

Aturan :

- Sudah memenuhi dalam bentuk normal kesatu (1NF)
- Semua atribut selain primary key, secara utuh memiliki Functional Dependency pada primary key.
- Sebuah tabel tidak memenuhi 2NF, jika ada atribut yang ketergantungannya (Functional Dependency) hanya bersifat parsial saja (hanya tergantung pada sebagian dari primary key).
- Jika terdapat atribut yang tidak memiliki ketergantungan terhadap primary key, maka atribut tersebut harus dipindah atau dihilangkan.

Contoh

■ Nilai Mahasiswa

Mhs_nrp	mhs_nama	mhs_alamat	mk_kode	mk_nama	mk_sks	nihuruf
133501006	Ahmad	Malang	A001	Basisdata	3	A
133501007	Suci	Blitar	A001	Basisdata	3	AB

- Tidak memenuhi 2NF, karena {Mhs_nrp, mk_kode} yang dianggap sebagai primary key, sedangkan:

{Mhs_nrp, mk_kode}	→	mhs_nama
{Mhs_nrp, mk_kode}	→	mhs_alamat
{Mhs_nrp, mk_kode}	→	mk_nama
{Mhs_nrp, mk_kode}	→	mk_sks
{Mhs_nrp, mk_kode}	→	nihuruf

tabel tersebut perlu didekomposisi

{Mhs_nrp, mk_kode} → nihuruf (fd1)
Mhs_nrp → {mhs_nama, mhs_alamat} (fd2)
Mk_kode → {mk_nama, mk_sks} (fd3)

fd1 (mhs_nrp, mk_kode, nihuruf) → Tabel Nilai

fd2 (Mhs_nrp, mhs_nama, mhs_alamat) → Tabel Mahasiswa

fd3 (mk_kode, mk_nama, mk_sks) → Tabel MataKuliah

Mhs_nrp	mk_kode	nihuruf
133501006	A001	A
133501007	A001	AB

Mhs_nrp	mhs_nama	mhs_alamat
133501006	Ahmad	Malang
133501007	Suci	Blitar

mk_kode	mk_nama	mk_sks
A001	Basisdata	3
A001	Basisdata	3

Normalisasi Ketiga (3rd Normal Form)

Aturan:

- Sudah berada dalam bentuk normal kedua (2NF)
- **Tidak ada** atribut *non primary key* yang memiliki ketergantungan terhadap atribut *non primary key* yang lainnya.
- Untuk setiap Functional Dependency dengan notasi $X \rightarrow A$, maka:
 - X harus menjadi superkey pada tabel tersebut
 - Atau A merupakan bagian dari primary key pada tabel tersebut

Contoh: Tabel Mahasiswa

Nrp	Nama	Alm_Jalan	Alm_Kota	Alm_Provinsi	Alm_Kodepos
133501006	ahmad	Kalasan	Malang	Jatim	60111
133501007	Suci	Candi	Blitar	Jatim	60312

- 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)

Nrp	Nama	Alm_Jalan	Alm_Kodepos
133501006	ahmad	Kalasan	60111
133501007	Suci	Candi	60312

Alm_Kota	Alm_Provinsi	Alm_Kodepos
Malang	Jatim	60111
Blitar	Jatim	60312

- Tabel-tabel yang memenuhi kriteria normalisasi ketiga, sudah siap diimplementasikan.
- Masih ada lagi bentuk normalisasi yang lain, seperti 4NF, 5NF, hanya saja jarang dipakai. Pada kebanyakan kasus, normalisasi hanya sampai ketiga.

- Fathansyah, *Basis Data*, Informatika, cetakan pertama, 1999
- Kroenke, David. *Database Processing: Fundamental, Design & Implementation 9th Edition*. New Jersey: Pearson Prentice Hall. 2005.
- Ramakrishnan, Raghu and Gehrke, Johannes. *Database Management Systems 3rd Edition*. McGraw-Hill, Inc. 2003.
- Priyadi, Yudi. *Kolaborasi SQL dan ERD dalam Implementasi Database*. Andi Yogyakarta. 2014