



PERANCANGAN BASIS DATA

PERTEMUAN KE -3

Rauf Fauzan, S.Kom.,M.Kom

Outline



- ✓ Normalisasi
 - ❑ Studi Kasus





- Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data yang mengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).
- *Normalisasi* adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan.



Tujuan Normalisasi



- Untuk menghilangkan kerangkapan data
- Untuk mengurangi kompleksitas
- Untuk mempermudah pemodifikasian data



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.



Tahapan Normalisasi



- Tahap Normalisasi dimulai dari tahap paling ringan (1NF) hingga paling ketat (5NF)
- Biasanya hanya sampai pada tingkat 3NF atau BCNF karena sudah cukup memadai untuk menghasilkan tabel-tabel yang berkualitas baik.
- Urutan: 1NF, 2NF, 3NF, BCNF, 4NF, 5NF

NF : Normal Form



Tahapan 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 Multivalued

Bentuk Normal Keempat (4NF)



Menghilangkan anomali-anomali yang tersisa

Bentuk Normal Kelima



Tahapan 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



Contoh :



Contoh 2

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



Fungsional Dependency



Notasi: $A \rightarrow B$

A dan B adalah atribut dari sebuah tabel. Berarti 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

➤ **Notasi: $A \rightarrow B$ atau $A \not\rightarrow B$**

Adalah kebalikan dari notasi sebelumnya.



Fungsional Dependency



CONTOH TABEL NILAI

<u>Namakul</u>	<u>Nrp</u>	<u>namaMhs</u>	<u>NiHuruf</u>
Struktur Data	980001	Ali Akbar	A
Struktur Data	980004	Indah Susanti	B
Basis Data	980001	Ali Akbar	
Basis Data	980002	Budi Haryanto	
Basis Data	980004	Indah Susanti	
Bahasa Indonesia	980001	Ali Akbar	B
Matematika I	980002	Budi Haryanto	C



Fungsional Dependency



Functional Dependency dari tabel nilai

➤ **Nrp** → **namaMhs**

Karena untuk setiap nilai **nrp** yang sama, maka nilai **namaMhs** juga sama.

➤ **{NamaKul, nrp}** → **NiHuruf**

Karena atribut **Nihuruf** tergantung pada **Namakul** dan **nrp** secara bersama-sama. Dalam arti lain untuk **Namakul** dan **nrp** yang sama, maka **NiHuruf** juga sama, karena **Namakul** dan **nrp** merupakan key (bersifat unik).

➤ **NamaKul** ↗ **nrp**

➤ **Nrp** ↗ **NiHuruf**



Fungsional Dependency



Andaikan ada tabel:

NILAI (NIM, Nm-mk, Semester, Nilai)

Atribut kunci: NIM, Nm-mk, Semester

Maka Functional Dependency:

NIM, Nm-mk, Semester \rightarrow Nilai





Bentuk Tidak Normal

Bentuk tidak normal atau *Unnormalized Form*, merupakan sekumpulan data yang akan direkam, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tersebut tidak lengkap maupun terduplikasi. Data dikumpulkan dengan apa adanya sesuai dengan kedatangannya.





1. Bentuk Normal Pertama

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)



Normalisasi



Contoh 1 (atribut multi-value)

Misal data mahasiswa sbb:

<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

Atau:

<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 di atas tidak memenuhi syarat 1NF



Normalisasi



Contoh 1 (samb..)

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



Normalisasi



Contoh 1 (samb..)

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



2. Bentuk Normal Kedua

Aturan :

- ✓ Sudah memenuhi dalam bentuk normal kesatu (1NF)
- ✓ Semua atribut bukan kunci hanya boleh tergantung (functional dependency) pada atribut kunci
- ✓ Jika ada **ketergantungan parsial** maka atribut tersebut harus dipisah pada tabel yang lain
- ✓ Perlu ada tabel penghubung ataupun kehadiran foreign key bagi atribut-atribut yang telah dipisah tadi

Normalisasi

Contoh



Tabel berikut memenuhi 1NF tapi tidak termasuk 2NF:

Mhs_nrp	mhs_nama	mhs_alamat	mk_kode	mk_nama	mk_sks	nihuruf
---------	----------	------------	---------	---------	--------	---------

- Tidak memenuhi 2NF, karena {Mhs_nrp, mk_kode} yang dianggap sebagai primary key sedangkan:
- Tabel di atas perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF





Normalisasi

Contoh(Samb..)

Functional dependencynya sbb:

$\{Mhs_nrp, mk_kode\} \rightarrow nihuruf$ (fd1)
 $Mhs_nrp \rightarrow \{mhs_nama, mhs_alamat\}$ (fd2)
 $Mk_kode \rightarrow \{mk_nama, mk_sks\}$ (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



3. Bentuk Normal Ketiga

Aturan :

- ✓ Sudah berada dalam bentuk normal kedua (2NF)
- ✓ Tidak ada ketergantungan transitif (dimana atribut bukan kunci tergantung pada atribut bukan kunci lainnya).

Normalisasi



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**):

$\text{alm_kodepos} \rightarrow \{\text{alm_Provinsi}, \text{alm_kota}\}$

- Sehingga tabel tersebut perlu didekomposisi menjadi:

Mahasiswa (Nrp, nama, alm_jalan, alm_kodepos)

Kodepos (alm_kodepos, alm_provinsi, alm_kota)





CATATAN :

Tabel-tabel yang memenuhi kriteria normalisasi ketiga, sudah siap diimplementasikan. Sebenarnya masih ada lagi bentuk normalisasi yang lain; Normalisasi Boyce-Codd, 4NF, 5NF, hanya saja jarang dipakai. Pada kebanyakan kasus, normalisasi hanya sampai ketiga.



~ TERIMA KASIH ~