



# DESAIN DATABASE DAN NORMALISASI



# Relasi (review)

---

- # Kumpulan atribut yang nilainya :
    - Setiap atribut harus berharga tunggal.
    - Semua harga pada suatu atribut harus mempunyai tipe yang sama
    - Setiap atribut harus mempunyai nama yang unik
    - Urutan atribut tidak penting
    - Tidak ada dua baris (tuple) dalam relasi yang isinya identik
    - Urutan baris (tuple) tidak penting
-

# Proses Rancangan Database (review)

---

- # Mengumpulkan kebutuhan user/business
  - # Mengembangkan E-R Model berdasarkan kebutuhan user/business
  - # Konversikan E-R Model ke himpunan relasi (tabel)
  - # Normalisasikan relasi untuk menghilangkan anomali
  - # Implementasikan ke database dengan membuat table untuk setiap relasi yang sudah ternormalisasi
-

# Normalisasi Basis Data (1)

---

- # *Normalisasi* adalah proses pembentukan struktur basis data sehingga sebagian besar *ambiguity* bisa dihilangkan.
  - # 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.
-

# Normalisasi Basis Data (2)

---

- # Mengapa dilakukan normalisasi ?
    - Optimalisasi struktur-struktur tabel
    - Meningkatkan kecepatan
    - Menghilangkan pemasukan data yang sama
    - Lebih efisien dalam penggunaan media penyimpanan
    - Mengurangi redundansi
    - Menghindari anomali (*insertion anomalies, deletion anomalies, update anomalies*).
    - Integritas data yang ditingkatkan
-

# Normalisasi Basis Data (3)

---

# 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)

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

---

# Functional Dependency (1)

---

- # *Functional Dependency* menggambarkan hubungan *attributes* dalam sebuah relasi
- # Suatu attribute dikatakan *functionally dependant* pada yang lain jika kita menggunakan harga atribut tersebut untuk menentukan harga atribut yang lain.
- # Simbol yang digunakan adalah  $\rightarrow$  untuk mewakili functional dependency.  
 $\rightarrow$  dibaca secara fungsional menentukan

# Functional Dependency (2)

---

## # 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: $A \not\rightarrow B$ atau $A \nrightarrow B$

Adalah kebalikan dari notasi sebelumnya.

---

# Functional Dependency (3)

## # Contoh tabel nilai

<b>Matakuliah</b>	<b>NIM</b>	<b>NamaMhs</b>	<b>NilaiHuruf</b>
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
Bahasa Indonesia	980003	Ali Akbar	B
Matematika 1	980002	Budi Haryanto	C
Matematika 1	980003	Ali Akbar	A

# Functional Dependency (4)

---

## # Functional Dependency dari tabel nilai

- **NIM  $\rightarrow$  NamaMhs**

Karena untuk setiap nilai nim yang sama, maka nilai NamaMhs juga sama

- **{Matakuliah, NIM}  $\rightarrow$  NilaiHuruf**

Karena atribut NilaiHuruf tergantung pada Matakuliah dan NIM secara bersama-sama. Dalam arti lain untuk Matakuliah dan NIM yang sama, maka NilaiHuruf juga sama, karena Matakuliah dan NIM merupakan key (bersifat unik).

- **Matakuliah  $\not\rightarrow$  NIM**

- **NIM  $\not\rightarrow$  NilaiHuruf**

---

# BENTUK NORMAL PERTAMA (First Normal Form - 1NF)

---

Tidak diperbolehkan adanya :

- # Atribut yang bernilai banyak (*Multivalued attribut*).
- # Attribut komposit atau kombinasi dari keduanya.

Jadi :

- # Harga domain atribut harus merupakan harga atomik
-

# Contoh (1)

# Misal Data Mahasiswa sbb :

<b>NIM</b>	<b>Nama</b>	<b>Hobi</b>
12020001	Heri Susanto	Sepak bola, membaca komik, berenang
12020013	Siti Zulaiha	Memasak, menyanyi
12020015	Dini Susanti	Menjahit, membuat roti

Atau

<b>NIM</b>	<b>Nama</b>	<b>Hobi1</b>	<b>Hobi2</b>	<b>Hobi3</b>
12020001	Heri Susanto	Sepak bola	membaca komik	berenang
12020013	Siti Zulaiha	Memasak	menyanyi	
12020015	Dini Susanti	Menjahit	membuat roti	

# Tabel-tabel di atas tidak memenuhi syarat 1NF

# Contoh (2)

## # Dekomposisi menjadi :

### ■ Tabel Mahasiswa :

<b>NIM</b>	<b>Nama</b>
12020001	Heri Susanto
12020013	Siti Zulaiha
12020015	Dini Susanti

### ■ Tabel Hobi :

<b>NIM</b>	<b>Hobi</b>
12020001	Sepak bola
12020001	membaca komik
12020001	berenang
12020013	Memasak
12020013	menyanyi
12020015	Menjahit
12020015	membuat roti

# BENTUK NORMAL KEDUA

## (Second Normal Form - 2NF) (1)

---

- # Bentuk normal 2NF terpenuhi dalam sebuah tabel jika telah memenuhi bentuk 1NF, dan 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
-

# BENTUK NORMAL KEDUA

## (Second Normal Form - 2NF) (2)

---

- # Ketergantungan fungsional  $X \rightarrow Y$  dikatakan **penuh** jika menghapus suatu atribut A dari X berarti Y tidak lagi bergantung fungsional.
  - # Ketergantungan fungsional  $X \rightarrow Y$  dikatakan **partial** jika menghapus suatu atribut A dari X berarti Y masih bergantung fungsional.
  - # Skema relasi R dalam bentuk 2NF jika setiap atribut non primary key  $A \in R$  bergantung penuh secara fungsional pada primary key R.
-

# Contoh (1)

- # Tabel berikut ini memenuhi 1NF, tetapi tidak termasuk 2NF

<b>NIM</b>	<b>NamaMhs</b>	<b>Alamat</b>	<b>KodeMk</b>	<b>Matakuliah</b>	<b>sks</b>	<b>NilaiHuruf</b>
980001	Ali Akbar	Jl Ahmad Yani	SD	Struktur Data	2	A
980004	Indah Susanti	Jl Tanjungpura	SD	Struktur Data	2	B
980001	Ali Akbar	Jl Ahmad Yani	BD	Basis Data	3	
980002	Budi Haryanto	Jl Purnama	BD	Basis Data	3	
980004	Indah Susanti	Jl Tanjungpura	BD	Basis Data	3	
980001	Ali Akbar	Jl Ahmad Yani	BI	Bahasa Indonesia	2	B
980003	Ali Akbar	Jl Imam Bonjol	BI	Bahasa Indonesia	2	B
980002	Budi Haryanto	Jl Ahmad Yani	M1	Matematika 1	2	C
980003	Ali Akbar	Jl Imam Bonjol	M1	Matematika 1	2	A

# Contoh (2)

---

- # Tidak memenuhi 2NF, karena {NIM, KodeMk} yang dianggap sebagai primary key sedangkan:
  - {NIM, KodeMk} → NamaMhs [T]
  - {NIM, KodeMk} → Alamat [T]
  - {NIM, KodeMk} → Matakuliah [Y]
  - {NIM, KodeMk} → Sks [Y]
  - {NIM, KodeMk} → NilaiHuruf [Y]
- # Tabel tersebut perlu didekomposisi menjadi beberapa tabel yang memenuhi syarat 2NF

# Contoh (3)

---

# Functional dependencynya sbb:

- $\{NIM, KodeMk\} \rightarrow NilaiHuruf$  (fd1)
- $NIM \rightarrow \{NamaMhs, Alamat\}$  (fd2)
- $KodeMk \rightarrow \{Matakuliah, Sks\}$  (fd3)

# Sehingga :

- fd1 (NIM, KodeMk, NilaiHuruf)  $\rightarrow$  Tabel Nilai
  - fd2 (NIM, NamaMhs, Alamat)  $\rightarrow$  Tabel Mahasiswa
  - fd3 (KodeMk, Matakuliah, Sks)  $\rightarrow$  Tabel MataKuliah
-

# BENTUK NORMAL KETIGA (Third Normal Form - 3NF) (1)

---

- # Bentuk normal 3NF terpenuhi jika telah memenuhi bentuk 2NF, dan jika **tidak ada** atribut *non primary key* yang memiliki ketergantungan terhadap atribut *non primary key* yang lainnya (**ketergantungan transitif**).
-

# BENTUK NORMAL KETIGA (Third Normal Form - 3NF) (2)

---

- # 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.
  - # Ketergantungan fungsional  $X \rightarrow Y$  dalam relasi R dikatakan bergantung transitif jika terdapat kumpulan atribut  $Z \notin$  himpunan primary key R sehingga  $X \rightarrow Z$  dan  $Z \rightarrow Y$ .
-

# Contoh (1)

- # Tabel mahasiswa berikut ini memenuhi syarat 2NF, tetapi tidak memenuhi 3NF

<b>NIM</b>	<b>NamaMhs</b>	<b>Jalan</b>	<b>Kota</b>	<b>Provinsi</b>	<b>KodePos</b>
980001	Ali Akbar	Jl Ahmad Yani	Pontianak	Kal-Bar	768854
980002	Budi Haryanto	Jl Purnama	Singkawang	Kal-Bar	776582
980003	Ali Akbar	Jl Lintas Melawi	Sintang	Kal-Bar	765862
980004	Indah Susanti	Jl Tanjungpura	Pontianak	Kal-Bar	768854

- # Karena masih terdapat atribut *non primary key* (yakni Kota dan Provinsi) yang memiliki ketergantungan terhadap atribut *non primary key* yang lain (yakni KodePos) :

KodePos  $\rightarrow$  {Kota, Provinsi}

# Contoh (2)

---

- # Sehingga tabel tersebut perlu didekomposisi menjadi :
    - Mahasiswa (NIM, NamaMhs, Jalan, KodePos)
    - KodePos (KodePos, Provinsi, Kota)
-

# Latihan

# Berikut ini contoh basis data yang belum ternormalisasi :

NoFaktur	TglFaktur	Suplier	Alamat	KdBrg	NamaBrg	Harga	JmlBrg
F001	12/12/2007	Sampurna	Jl. A Yani	001, 002	Pensil, Penghapus	500, 1000	100, 50
F002	12/12/2007	Permata	Jl. Gajah Mada	004, 001	Karton, Pensile	1200, 500	150, 70

Masalah :

- Tidak perlu berulang-ulang kali menyimpan data alamat, nama barang, dan harga.
- *Redundancy* dapat menimbulkan masalah, yaitu data yang tidak konsisten (pensile)