

# Pertemuan 5

# Normalisasi

# Normalisasi

**Dalam merancang basis data dapat dilakukan dengan:**

1. Menerapkan Normalisasi terhadap struktur tabel yang telah diketahui, atau
2. Langsung membuat model *Entity-Relationship*.

Normalisasi merupakan cara pendekatan lain dalam membangun desain logik basis data dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal.

# Normalisasi Lanjutan

## **BEBERAPA PENGERTIAN NORMALISASI :**

Normalisasi merupakan proses pengelompokan elemen data menjadi tabel–tabel yang menunjuk-kan entity dan relasinya.

Normalisasi adalah proses pengelompokan attribute-attribute dari suatu relasi sehingga membentuk WELL STRUCTURE RELATION.

# Well Structure Relation

Adalah sebuah relasi yang jumlah kerangkapan datanya sedikit (*minimum Amount Of Redundancy*), serta memberikan kemungkinan bagi user untuk melakukan INSERT, DELETE, dan MODIFY terhadap baris-baris data pada relation tersebut, yang tidak berakibat terjadinya ERROR atau INKONSESTENSI DATA, yang disebabkan oleh operasi-operasi tersebut

# Keuntungan Normalisasi

## **Keuntungan dari normalisasi, yaitu :**

1. Meminimalkan ukuran penyimpanan yang diperlukan untuk menyimpan data.
2. Meminimalkan resiko inkonsistensi data pada basis data
3. Meminimalkan kemungkinan anomali pembaruan
4. Memaksimalkan stabilitas struktur data

# ANOMALY

ANOMALY merupakan penyimpangan-penyimpangan atau Error atau inkonsistensi data yang terjadi pada saat dilakukan proses insert, delete maupun update.

Terdapat 3 jenis Anomaly :

## 1. Insertion Anomaly

Error yang terjadi sebagai akibat operasi insert record/tuple pada sebuah relation

## 2. Deletion Anomaly

Error yang terjadi sebagai akibat operasi delete record/tuple pada sebuah relation

# Anomaly Lanjutan

## 3. Update Anomaly

Error yang terjadi sebagai akibat inkonsistensi data yang terjadi sebagai akibat dari operasi update record/tuple dari sebuah relation

# Problem-Problem Pada Relation yang Sudah Dinormalisasi

- Performance problem  
Masalah terhadap performa database
- Referential Integrity Problem  
Masalah yang timbul terhadap referensi antar data-data diantara dua tabel atau lebih

## **BEBERAPA KONSEP YANG HARUS DIKETAHUI:**

- a. Field/ Atribut Kunci (sudah di bahas di pertemuan 2)
- b. Kebergantungan Fungsi



# Atribut Kunci (Field)

(Sudah Di bahas di Pertemuan 2)

a. Key Field / attribute kunci dalam database:

1. *Super key*
2. *Candidate key*
3. *Primary key*
4. *Alternate key*
5. *Foreign key*

# Kebergantungan Kunci

1. Ketergantungan Fungsional (Fungsional Dependent)  
Keterkaitan antar hubungan antara 2 atribut pada sebuah relasi. Dituliskan dengan cara :  $A \rightarrow B$ , yang berarti :  
Atribut B fungsionalitas Dependent terhadap atribut A atau  
Isi (*value*) atribut A menentukan isi atribut B  
Definisi dari functional dependent :  
Diketahui sebuah relasi R, atribut Y dari R adalah FD pada atribut X dari R ditulis  $R.X \rightarrow R.Y$  jika dan hanya jika tiap harga X dalam R bersesuaian dengan tepat satu harga Y dalam R

# Kebergantungan Kunci lanjutan

## 2. Fully Functionaly Dependent (FFD)

Suatu rinci data dikatakan fully functional dependent pada suatu kombinasi rinci data jika functional dependent pada kombinasi rinci data dan tidak functional dependent pada bagian lain dari kombinasi rinci data.

Definisi dari FDD:

Attribute Y pada relasi R adalah FFD pada attribute X pada relasi R jika Y FD pada X tida FD pada himpunan bagian dari X

Contoh:

PersonID, Project, Project\_budget  $\rightarrow$  time\_spent\_byperson\_onProject (bukan FFD)

PersonID, Project  $\rightarrow$  time\_spent\_byperson\_onProject (FDD)

# Kebergantungan Kunci lanjutan

## 3. Ketergantungan Partial

Sebagian dari kunci dapat digunakan sebagai kunci utama

## 4. Ketergantungan Transitif

Menjadi attribute biasa pada suatu relasi tetapi menjadi kunci pada relasi lain

## 5. Determinan

Suatu attribute (field) atau gabungan attribute dimana beberapa attribute lain bergantung sepenuhnya pada attribute tersebut

# Bentuk Normal

Aturan-aturan normalisasi dinyatakan dengan istilah bentuk normal. **Bentuk normal** adalah suatu aturan yang dikenakan pada relasi-relasi dalam basis data dan harus dipenuhi oleh relasi-relasi tersebut pada level-level normalisasi.

Beberapa level yang biasa digunakan pada normalisasi adalah:

- Bentuk normal pertama (1NF)
- Bentuk normal kedua (2NF)
- Bentuk normal ketiga (3NF)
- Bentuk normal Boyce-Codd (BCNF)
- Bentuk normal keempat (4NF)
- Bentuk Normal kelima (5NF)

# Latihan Soal

1. Urutan level yang biasa digunakan pada normalisasi adalah :
  - a. 1NF, BCNF, 2NF, 3NF, 4NF, 5NF
  - b. 1NF, 2NF, BCNF, 3NF, 4NF, 5NF
  - c. 1NF, 2NF, 3NF, BCNF, 4NF, 5NF
  - d. 1NF, 2NF, 3NF, 4NF, BCNF, 5NF
  - e. 1NF, 2NF, 3NF, 4NF, 5NF, BCNF

2. Sebuah relasi yang jumlah kerangkapan datanya sedikit.  
Merupakan pengertian dari :
- a. Best Structure Relation
  - b. Good Structure Relation
  - c. Well Structure Relation
  - d. Bad Structure Relation
  - e. Worst Structure Relation

3. Sebagian dari kunci dapat digunakan sebagai kunci utama. Merupakan pengertian dari:

- a. Ketergantungan Fungsional
- b. Fully Functionaly Dependent
- c. Ketergantungan Partial
- d. Ketergantungan Transitif
- e. Determinan



4. Proses pengelompokan elemen data menjadi tabel–tabel yang menunjuk-kan entity dan relasinya
  - a. Well Structure Relation
  - b. Normalisasi
  - c. ERD
  - d. Struktur file
  - e. Inkonsistensi Data

## 5. Anomaly merupakan:

- a. Error yang terjadi sebagai akibat operasi insert record/tuple pada sebuah relation
- b. Error yang terjadi sebagai akibat inkonsistensi data yang terjadi sebagai akibat dari operasi update record/tuple dari sebuah relation
- c. Penyimpangan-penyimpangan atau Error atau inkonsistensi data yang terjadi pada saat dilakukan proses insert, delete maupun update.
- d. Penyimpangan-penyimpangan atau Error atau inkonsistensi data yang terjadi pada saat dilakukan proses create, drop maupun alter.
- e. Masalah yang timbul terhadap referensi antar data-data diantara dua tabel atau lebih