

# **SQL DML**

Boldson H. Situmorang, S.Kom., MMSI

# MANIPULASI DATA

- ◆ Data Manipulation Language (DML) merupakan bahasa basis data yang berguna untuk melakukan modifikasi dan pengambilan data pada suatu basis data
- ◆ Modifikasi data terdiri dari: penambahan (insert), pembaruan (update) dan penghapusan (delete).

# Penambahan Data

- ◆ Instruksi SQL untuk melakukan penambahan data adalah menggunakan syntax:

```
INSERT INTO <namaTabel> [(field1, field2, ...)]  
VALUES (field1 [,field2, ...]) | SQL-SELECT
```

## Keterangan

- ❖ <namaTabel> → nama tabel yang akan ditambahkan datanya
- ❖ [(field1, field2, ...)] → field-field di dalam tabel yang akan diisikan nilainya
- ❖ **VALUES** (nilai1 [,nilai2, ...]) | **SQL-SELECT** → nilai yang diisikan  
Jika mengisikan sebuah data tunggal saja yang tidak diambil dari tabel lain, gunakan:

```
VALUES (nilai1 [,nilai2, ...])
```

# SQL-Select

- ◆ Adapun jika nilai-nilai yang diisikan merupakan hasil query dari tabel lain, maka gunakan: **SQL-SELECT**
- ◆ **SQL-SELECT** memiliki ketentuan sebagai berikut:
  - ❖ jumlah kolom yang didefinisikan,
  - ❖ jenis tipe data kolom
  - ❖ dan jumlahnya

harus sama dengan field-field yang didefinisikan pada tabel yang akan ditambahkan isinya

# Penambahan Data

## Contoh

Untuk mengisi data pada tabel penerbit:

```
INSERT INTO penerbit ↵  
VALUES (90, 'CV Cempaka', 'Jl Gebang Wetan 99', ↵  
'59911111', 'http://www.cempaka.co.id')
```

Contoh di atas tidak menyertakan klausa [(field1, field2, ...)], sehingga pengisiannya harus seluruh field dan urutannya harus benar sesuai dengan urutan field pada struktur tabel.

# Penambahan Data

Contoh

Untuk mengisikan data pada tabel penerbit:

```
INSERT INTO penerbit (PN_ID, PN_Nama) ↵  
VALUES (91, 'CV Angkasa')
```

Contoh di atas menyebutkan field-field yang diisikan pada tabel penerbit, sehingga nilai-nilai yang ditulis setelah klausa VALUES juga harus mengikuti field-field tersebut.

# Mengubah Data

- ◆ Instruksi SQL untuk melakukan perubahan data adalah menggunakan syntax:

```
UPDATE <namaTabel>  
SET <field1>=<nilai1> [ , <field2> = <nilai2>, ...]  
[WHERE <kondisi>]
```

## Keterangan

- ❖ <namaTabel> → nama tabel yang akan ditambahkan datanya
- ❖ SET <field1>=<nilai1> [,<field2>=<nilai2>,... ] → nilai baru yang akan diisikan pada field tertentu
- ❖ [WHERE <kondisi>] → filter yang berlaku untuk menentukan data mana saja yang diupdate

# Mengubah Data

## Contoh

- ◆ Untuk melakukan update massal (berlaku untuk seluruh field), yakni menaikkan seluruh harga sebesar 110% pada koleksi:

```
UPDATE koleksi SET KL_Harga=KL_Harga*1.1
```

- Untuk melakukan update tertentu, yakni memberikan keterangan dg isian 'Buku TA' untuk semua koleksi yang berjenis buku TA (KL\_TK\_ID=4):

```
UPDATE koleksi SET KL_Keterangan = 'Buku TA'  
WHERE KL_TK_ID=4
```



# Menghapus Data

- ◆ Instruksi SQL untuk menghapus data adalah menggunakan syntax:

```
DELETE FROM <namaTabel>  
[WHERE <kondisi>]
```

## Keterangan

- ❖ <namaTabel> → nama tabel yang akan ditambahkan datanya
- ❖ [WHERE <kondisi>] → filter yang berlaku untuk menentukan data mana saja yang dihapus

# Menghapus Data

## Contoh

- ◆ Untuk menghapus seluruh data peminjaman:

```
DELETE FROM Peminjaman
```

- Untuk menghapus seluruh koleksi yang berjenis buku TA (idJenisKoleksi=4)

```
DELETE FROM koleksi WHERE KL_TK_ID=4
```

# Contoh Tabel

## Contoh Data:

Tabel Mahasiswa

NRP	Nama	NIPWali
51001	Joko	131001
51002	Dini	131002
51003	Dono	131002
51004	Paijo	<NULL>

Tabel Dosen:

NIP	NamaDosen
131001	Pak Adi
131002	Pak Budi
131003	Bu Cici
131004	Pak Dono
131005	Bu Eni