

MODUL II MANIPULASI DATA

Tujuan :

1. Mahasiswa dapat melakukan manipulasi data yang tersimpan dalam basis data.
2. Mahasiswa dapat memahami perintah INSERT.
3. Mahasiswa dapat memahami perintah UPDATE.
4. Mahasiswa dapat memahami perintah DELETE.

Tugas Pendahuluan :

1. Apa yang anda ketahui tentang DML?
2. Sebutkan perintah-perintah SQL yang tergolong dalam DML dan jelaskan perbedaan antar DML dan DDL!
3. Apa yang anda ketahui tentang perintah INSERT, UPDATE, dan DELETE?
4. Sebutkan macam-macam klausa dan operator pada perintah SELECT!

DASAR TEORI

DML (Data Manipulation Language) adalah bahasa yang memungkinkan pengguna mengakses atau memanipulasi data yang berbentuk suatu model data tertentu (Silberschatz,2011). Tipe akses data adalah (Silberschatz,2011):

1. Pengambilan informasi yang disimpan dalam basis data.
2. Penempatan informasi baru dalam basis data.
3. Penghapusan informasi dari basis data.
4. Modifikasi informasi yang disimpan dalam basis data.

Adapun perintah-perintah umum dalam DML yang disediakan dalam MySQL adalah sebagai berikut:

1. **INSERT**, menyisipkan atau menambahkan data (*tuple*) baru ke dalam tabel.
2. **SELECT**, mengambil atau menampilkan data dari tabel. Perintah ini adalah perintah dasar yang digunakan untuk mengambil informasi dari basis data
3. **UPDATE**, memperbaharui data yang lama ke dalam data yang baru.
4. **DELETE**, menghapus data dalam tabel.

Untuk lebih memahami perintah-perintah DML (Data Manipulation Language), berikut ini adalah penjelasan dalam praktikum untuk masing-masing dari perintah-perintah di atas.

KEGIATAN PRAKTIKUM

INSERT

Sebelum memulai praktikum, pastikan bahwa struktur basis data yang digunakan sudah ada sebelumnya. Struktur basis data yang digunakan disini adalah struktur basis data yang dibuat dalam MODUL II sebelum dikenakan perintah ALTER. Setelah tabel dibuat beserta constraint-nya jika diperlukan, maka tabel siap digunakan untuk menampung data. Perintah pada MySQL untuk memasukkan data ke dalam tabel adalah perintah INSERT. Masukkan data pada tabel barang dan tabel pembelian. Bentuk perintah INSERT adalah sebagai berikut :

```
INSERT INTO [tabel]
([field1],[field2],[field3],...)
VALUES ([value1],[value2],[value3],...);
```

Memasukkan data pada tabel Barang

```
INSERT INTO barang
(id_barang,nama_barang,tanggal_terima)
VALUES ('BRG01','Sony','2015-04-03');
```

Untuk memasukkan beberapa data sekaligus, dapat menggunakan perintah sebagai berikut:

```
INSERT INTO barang
(id_barang,nama_barang,tanggal_terima)
VALUES ('BRG01','Sony','2015-04-03'),
('BRG02','Samsung','2015-05-03');
```

Dan data seterusnya silahkan masukkan sendiri dengan petunjuk perintah diatas berdasarkan data pada Gambar 2.1.

<input type="checkbox"/>	id_barang	nama_barang	tanggal_terima	stok_barang
<input type="checkbox"/>	BRG01	Sony	2015-04-03 00:00:00	0
<input type="checkbox"/>	BRG02	Samsung	2015-05-03 00:00:00	0
<input type="checkbox"/>	BRG03	HTC	2015-03-08 00:00:00	0
<input type="checkbox"/>	BRG04	Microsoft	2015-03-10 00:00:00	0
<input type="checkbox"/>	BRG05	Motorola	2015-03-15 00:00:00	0
<input type="checkbox"/>	BRG06	Xiaomi	2015-03-17 00:00:00	0
*	(NULL)	(NULL)	(NULL)	0

Gambar 2.1 Hasil Penambahan data pada tabel barang

Kolom yang tidak disebutkan pada pernyataan INSERT secara otomatis akan diisi dengan NULL. Pada Stok_Barang nilainya akan terisi 0 karena pada saat membuat tabel Barang memberikan tipe data int, default value untuk int adalah 0.

Memasukkan data pada tabel pembelian.

```
INSERT INTO pembelian
(id_pembeli, id_barang, tanggal_beli, nama_pembeli,
jumlah_pembelian)
VALUES ('P01', 'BRG03', '2015-04-07', 'Mahadewi Istirani', 2);
```

Dan data seterusnya silahkan masukkan sendiri dengan petunjuk perintah diatas berdasarkan data pada Gambar 2.2.

<input type="checkbox"/>	id_pembeli	id_barang	tanggal_beli	nama_pembeli	jumlah_pembelian
<input type="checkbox"/>	P01	BRG03	2015-03-04 00:00:00	Putu	2
<input type="checkbox"/>	P02	BRG02	2015-04-06 00:00:00	Made	2
<input type="checkbox"/>	P03	BRG06	2015-05-08 00:00:00	Komang	2
<input type="checkbox"/>	P04	BRG06	2015-06-10 00:00:00	Ketut	1
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 2.2 Hasil penambahan data pada tabel pembelian

Memasukkan data dari tabel lain juga dimungkinkan dalam perintah insert. Untuk mempraktekannya buatlah satu tabel lagi pada basis data “toko” dengan nama tabel “pelanggan” yang memiliki *field* ID_Pelanggan dan Nama_Pelanggan. Untuk membuat tabel pelanggan gunakan perintah dibawah ini:

```
CREATE TABLE pelanggan
(id_pelanggan VARCHAR(10),
nama_pelanggan VARCHAR(80));
```

Setelah tabel pelanggan dibuat, kemudian jalankan perintah dibawah ini untuk mengisi tabel “Pelanggan” dengan data dari tabel pembelian.

```
INSERT INTO pelanggan
(id_pelanggan, nama_pelanggan)
SELECT id_pembeli, nama_pembeli FROM pembelian;
```

Untuk memeriksa kebenarannya, gunakan perintah dibawah ini dan lihat hasilnya!

```
SELECT * FROM pelanggan;
```

<input type="checkbox"/>	id_pelanggan	nama_pelanggan
<input type="checkbox"/>	P01	Putu
<input type="checkbox"/>	P02	Made
<input type="checkbox"/>	P03	Komang
<input type="checkbox"/>	P04	Ketut
*	(NULL)	(NULL)

Gambar 2.3 Hasil dari penambahan pada tabel Pelanggan.

UPDATE

Perintah UPDATE digunakan untuk memodifikasi data pada tabel dalam basis data.

```
UPDATE [tabel]
  SET [field] = '[value baru]'
  WHERE [field acuan] = '[value]';
```

Keterangan :

SET : Untuk menentukan kolom yang akan diubah dan isi penggantinya.

WHERE : Menentukan kondisi atau syarat dari baris-baris yang akan diganti.

Berikut ini perintah untuk meng-*update field* nama barang pada tabel barang dengan nama “LG” yang awalnya bernama “HTC”.

```
UPDATE barang
  SET nama_barang='LG'
  WHERE id_barang='BRG03';
```

<input type="checkbox"/>	▲ id_barang	nama_barang	tanggal_terima	stok_barang
<input type="checkbox"/>	BRG01	Sony	2015-04-03 00:00:00	0
<input type="checkbox"/>	BRG02	HTC	0000-00-00 00:00:00	0
<input type="checkbox"/>	BRG03	LG	2015-03-08 00:00:00	0
<input type="checkbox"/>	BRG04	Microsoft	2015-03-10 00:00:00	0
<input type="checkbox"/>	BRG05	Motorola	2015-03-15 00:00:00	0
<input type="checkbox"/>	BRG06	Xiaomi	2015-03-17 00:00:00	0
*	(NULL)	(NULL)	(NULL)	0

Gambar 2.4 Hasil perintah UPDATE pada tabel barang

DELETE

Pernyataan DELETE digunakan untuk menghapus baris pada tabel, bentuk perintahnya sebagai berikut.

```
DELETE FROM [tabel]
  WHERE [field] = '[values]';
```

Pada contoh ini akan menghapus *record* dengan ID_Barang = 'BRG01' pada tabel pembelian. Untuk mempraktekkannya tulislah perintah dibawah ini!

```
DELETE FROM pembelian
  WHERE id_barang = 'BRG03';
```

<input type="checkbox"/>	id_pembeli	id_barang	tanggal_beli	nama_pembeli	jumlah_pembelian
<input type="checkbox"/>	P02	BRG02	2015-04-06 00:00:00	Made	2
<input type="checkbox"/>	P03	BRG06	2015-05-08 00:00:00	Komang	2
<input type="checkbox"/>	P04	BRG06	2015-06-10 00:00:00	Ketut	1
*	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Gambar 2.5 Hasil dari perintah DELETE pada tabel pembelian.

Dalam perintah DELETE jika ingin menghapus semua data pada tabel tanpa menghapus tabel, maka klausa WHERE tidak perlu disebutkan, berikut perintahnya.

```
DELETE FROM [tabel];
atau
DELETE * FROM [tabel];
atau
TRUNCATE [tabel];
```

Untuk mengambil, mengubah, atau menghapus data, dapat digunakan where untuk mencari beberapa record saja sesuai kriteria yang ditetapkan. Misal

```
SELECT * FROM barang WHERE id_barang='BRG03';
```

Untuk mendapatkan data yang memenuhi beber nilai untuk sebuah kriteria, dapat digunakan in

```
SELECT * FROM barang WHERE id_barang IN ('BRG04', 'BRG05');
```

Untuk mendapatkan data dengan kriteria tidak sama persis, digunakan like, contohnya:

```
SELECT * FROM pelanggan WHERE nama_pembeli LIKE %tu%';
```

Tanda persen (%) dapat diartikan sebagai apa saja, jadi dengan menulis '%tu%' berarti akan mencari semua yang mengandung 'tu' apapun huruf di depan dan belakangnya. Namun, jika ditulis '%tu' maka akan dicari hanya yang berakhiran dengan 'tu'.

TUGAS

1. Isilah tabel jurusan dan tabel biodata yang anda buat dengan ketentuan sebagai berikut: Misalkan isi tabel jurusan:

Kode Jurusan	Nama Jurusan	Ketua Jurusan
KD01	Sistem Informasi	Harnaningrum,S.Si
KD02	Teknik Informatika	EnnySela,S.Kom.,M.Kom
KD03	Tekhnik Komputer	Berta Bednar,S.Si,M.T.

Isi tabel biodata:

No Mahasiswa	Kode Jurusan	Nama Mahasiswa	Alamat	IPK
210089	KD01	Rina Gunawan	Denpasar	3
210090	KD03	Gani Suprpto	Singaraja	3.5
210012	KD02	Alexandra	Nusa dua	3
210099	KD02	Nadine	Gianyar	3.2
210002	KD01	Rizal Samurai	Denpasar	3.7

- Masukkan data baru pada tabel biodata dengan kode jurusan “KD04”!
- Dengan menggunakan perintah UPDATE, cobalah merubah isi tabel jurusan dan biodata pada basis data mahasiswa yang anda buat dengan ketentuan sebagai berikut :
 - Mengganti nama mahasiswa pada tabel biodata “Rina Gunawan” menjadi “Rina Gunawan Astuti”!
 - Mengganti kode jurusan pada tabel jurusan “KD01” menjadi “KM01”!
 - Mengganti no mahasiswa pada tabel biodata “210089” menjadi “210098”!
 - Mengganti nilai pada tabel biodata “3” menjadi “3.3”!
 - Mengganti kode jurusan pada tabel biodata “KD03” menjadi “KD05”!
- Buatlah kesimpulan mengenai soal no.2 dan no.3!