

BAB IV

Data Manipulation Language menggunakan MariaDB

3.1 Materi

3.1.1 Hubungan Antar Tabel (*Reference*)

Dalam hal hubungan antar tabel, ada beberapa syarat yang harus terpenuhi seperti, antara tabel yang ingin dihubungkan harus memiliki sebuah *field* atau kolom dengan tipe data yang sama. Dengan kata lain ada tabel yang memiliki *primary key* sebagai kunci untuk menghubungkan ke tabel yang lain dimana terdapat *field* yang mempunyai tipe data yang sama dengan kata lain yang disebut *foreign key*. Misalnya, terdapat tabel barang dan macam_barang. Berikut *query* kedua tabel tersebut :

- **create table macam_barang (id_mac integer not null, nama_mac varchar(15), guna_mac varchar(40), primary key (id_mac));**
- **create table barang (id_bar integer not null, nama_bar varchar(15), id_mac integer not null references macam_barang, primary key (id_bar, id_mac));**

```
MySQL [(none)]> use mariadbml;
Database changed
MySQL [mariadbml]> create table macam_barang (id_mac integer not null, nama_mac varchar(15), guna_mac varchar(40), primary key (id_mac));
Query OK, 0 rows affected (0.17 sec)

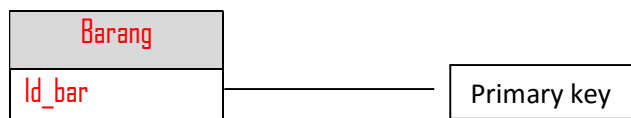
MySQL [mariadbml]> show tables;
+-----+
| Tables_in_mariadbml |
+-----+
| macam_barang         |
+-----+
1 row in set (0.00 sec)

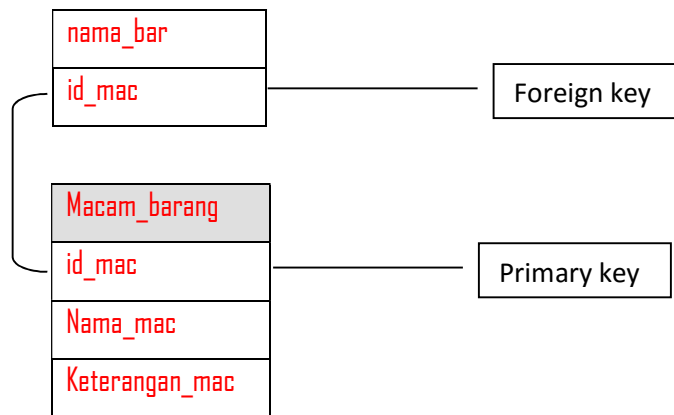
MySQL [mariadbml]> create table barang (id_bar integer not null, nama_bar varchar(15), id_mac integer not null references macam_barang, primary key (id_bar, id_mac));
Query OK, 0 rows affected (0.17 sec)

MySQL [mariadbml]> show tables;
+-----+
| Tables_in_mariadbml |
+-----+
| barang              |
| macam_barang        |
+-----+
2 rows in set (0.00 sec)
```

Atau ketika kita lupa memberikan *references* pada tabel macam barang kita dapat memberikan *query* sebagai berikut :

- **alter table barang add constraint relasi_barang foreign key (id_mac) references macam_barang (id_mac) on update no action on delete no action;**





3.2.2 Memasukkan Data

Sebuah data dapat dimasukkan ke dalam tabel menggunakan perintah INSERT. Namun hal yang perlu diperhatikan yaitu struktur tabel yang akan digunakan untuk memasukkan data. Sebagai contoh memasukkan data ke dalam tabel dengan nama identitas :

IDENTITAS
id_idn
Nama_idn
Ttl_idn
Menikah_idn
Keahlian_idn

insert into identitas values (1, 'wahyu' , '1992-07-23' , false , 'programer');

Berikut penjelasan :

- Kolom pertama berisikan (1), karena tipe datanya adalah integer maka tanda petik tidak digunakan.
- Kolom kedua ('wahyu') dan kelima ('programer'), karena tipe datanya varchar maka tanda petik diperlukan, bila tanpa ada tanda petik maka dianggap salah.
- Kolom ketiga ('1992-07-23'), tipe data Date dianggap sebagai sebuah string sehingga harus menggunakan tanda petik.
- Kolom keempat (false), tipe data kolom ini Boolean yang hanya mengijinkan inputan true atau false dan tidak perlu tanda petik.

3.2.3 Menampilkan data

Dalam database, perintah SQL untuk menampilkan data sebuah tabel bisa menggunakan SELECT. Berikut struktur SQL untuk penampilan data berdasarkan kolom tertentu :

Select nama_kolom from nama_tabel;

Contoh :

```
MySQL [mariadbmysql]> select nama_bar from barang;
+-----+
| nama_bar |
+-----+
| buku garis |
+-----+
1 row in set (0.00 sec)
```

Jika ingin menampilkan keseluruhan kolom dan keseluruhan baris suatu tabel bisa mengganti **nama_kolom** menggunakan tanda asterisk (*).

Contoh :

```
MySQL [mariadbmysql]> select * from barang;
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
|      1 | buku garis |      1 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Kemudian, bila ingin menampilkan berdasarkan baris tertentu bisa menggunakan perintah **WHERE** yang diletakkan dibelakang nama tabel dan isikan nama kolom beserta kata kunci sebagai dasar atau syarat dari pencarian baris. Untuk lebih jelasnya berikut struktur perintah SQL nya :

select nama_kolom from nama_tabel where nama_kolom = kata kunci;

Contoh :

```
MySQL [mariadbmysql]> select * from barang where id_mac=1;
+-----+-----+-----+
| id_bar | nama_bar | id_mac |
+-----+-----+-----+
|      1 | buku garis |      1 |
+-----+-----+-----+
1 row in set (0.04 sec)
```

Apabila ingin menampilkan data dari lebih dari satu tabel, struktur SQL nya terdapat sedikit perbedaan. Misalnya memakai tabel yang telah dibuat pada praktikum ke-2 yaitu mahasiswa dan fakultas (Dengan syarat adanya relasi antara kedua tabel tersebut). Berikut Struktur SQL nya :

**select nama_kolom, nama_kolom, nama_kolom, nama_kolom from nama_tabel_1, nama_tabel_2
where nama_tabel_1.nama_kolom = nama_tabel_2.nama_kolom;**

Contoh :

```

MySQL [mariadbmysql]> insert into macam_barang values (125, 'JenisHapus' , 'gunaHapus'),( 124, 'JenisAlatTulis','gunaMenulis');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

MySQL [mariadbmysql]> select*from macam_barang;
+-----+-----+-----+
| id_mac | nama_mac | guna_mac |
+-----+-----+-----+
| 123 | JenisBuku | gunaMenulis |
| 125 | JenisHapus | gunaHapus |
| 124 | JenisAlatTulis | gunaMenulis |
+-----+-----+-----+
3 rows in set (0.00 sec)

MySQL [mariadbmysql]> select*from macam_barang where guna_mac='gunaMenulis';
+-----+-----+-----+
| id_mac | nama_mac | guna_mac |
+-----+-----+-----+
| 123 | JenisBuku | gunaMenulis |
| 124 | JenisAlatTulis | gunaMenulis |
+-----+-----+-----+
2 rows in set (0.00 sec)

```

3.2.4 Menghapus Data

Pada database PostgreSQL, DELETE digunakan untuk menghapus data pada sebuah tabel. Berikut Struktur yang digunakan :

delete from namatabel ;

Arti dari struktur diatas yaitu menghapus semua baris data dari nama tabel yang tertulis. Namun bisa juga digunakan penghapusan berdasarkan spesifikasi baris atau baris tertentu saja. Berikut Strukturnya :

delete from nama tabel where namakolom = katakunci;

contoh :

```

MySQL [mariadbmysql]> show tables;
+-----+
| Tables_in_mariadbmysql |
+-----+
| barang |
| baranghapus |
| macam_barang |
+-----+
3 rows in set (0.00 sec)

MySQL [mariadbmysql]> select * from barangHapus;
+-----+-----+
| id_bar | nama_bar |
+-----+-----+
| 10 | kotakPensil |
| 11 | BukuGambar |
| 12 | Penggaris |
+-----+-----+
3 rows in set (0.00 sec)

MySQL [mariadbmysql]> select * from barangHapus where nama_bar='Penggaris';
+-----+-----+
| id_bar | nama_bar |
+-----+-----+
| 12 | Penggaris |
+-----+-----+
1 row in set (0.00 sec)

```

3.2.5 Modifikasi Data

Dalam memodifikasi database bisa menggunakan perintah UPDATE. Berikut contoh struktur SQL untuk memodifikasi data pada kolom tertentu berdasarkan baris tertentu :

update namatabel set namakolom = isidata where namakolom = katakunci;

Contoh :

```
MySQL [mariadbml]> update barangHapus set nama_bar='kotakPensilBiru' where id_bar=10;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MySQL [mariadbml]> select * from barangHapus;
+-----+-----+
| id_bar | nama_bar |
+-----+-----+
|      10 | kotakPensilBiru |
|      11 | BukuGambar |
|      12 | Penggaris |
+-----+-----+
3 rows in set (0.00 sec)
```

3.2.6 Pengurutan Data

Maksud dari pengurutan data dengan perintah ORDER BY adalah jika data sebuah tabel diurutkan dengan perintah tersebut, maka data tadi akan diurutkan dari atas ke bawah berdasarkan abjad. Hal ini juga berlaku jika data yang ada dalam bentuk angka. Berikut struktur perintah SQL untuk pengurutan berdasarkan kolom :

select * from nama_tabel order by nama_kolom;

Contoh : data fakultas diurutkan berdasarkan kolom nama fakultas secara *ascending*.

```
MySQL [mariadbml]> select * from barangHapus;
+-----+-----+
| id_bar | nama_bar |
+-----+-----+
|      10 | kotakPensilBiru |
|      11 | BukuGambar |
|      12 | Penggaris |
+-----+-----+
3 rows in set (0.00 sec)

MySQL [mariadbml]> select * from barangHapus order by nama_bar;
+-----+-----+
| id_bar | nama_bar |
+-----+-----+
|      11 | BukuGambar |
|      10 | kotakPensilBiru |
|      12 | Penggaris |
+-----+-----+
3 rows in set (0.00 sec)
```

Berikut struktur perintah SQL untuk pengurutan berdasarkan kolom secara descending:

select * from nama_tabel order by nama_kolom desc;

Berikut struktur perintah SQL untuk pengurutan berdasarkan beberapa kolom :

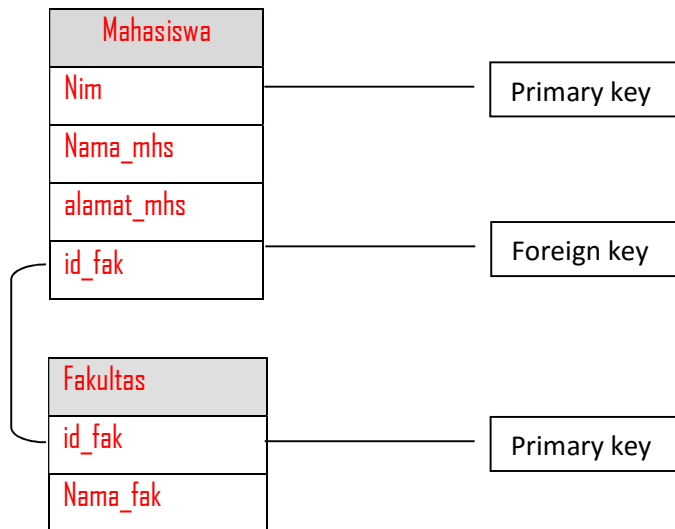
select * from nama_tabel order by nama_kolom asc, nama_kolom desc;

Berikut struktur perintah SQL untuk pengurutan berdasarkan nomor kolom :

select * from nama_tabel order by nomor_kolom desc;

*catatan : nomor_kolom dimulai dari kiri, misal terdapat kolom nim, nama. Maka jika ingin mengurutkan berdasarkan nama kita perlu mengganti **nomor_kolom** menjadi 2.

Tugas Praktikum



Gambar 1 Diagram Database Mahasiswa Baru

1. Buatlah tabel seperti pada gambar 1 diagram database mahasiswa baru
2. Isi tabel dengan 3 data mahasiswa sebagai berikut:

Nim : 12

Nama : edi

Alamat : malang

No.telp : 0856453428

Fakultas : SAINTEK

Nim : 13

Nama : sinta

Alamat : jogja

No.telp : 0813347565675

Fakultas : SAINTEK

Nim : 14

Nama : luki

Alamat : ponorogo

No.telp : 082576568798

Fakultas : PSIKOLOGI

3. Tampilkan data mahasiswa berupa nim dan alamat
4. Tampilkan data mahasiswa berdasarkan nim salah satu mahasiswa. Sehingga yang muncul hanya salah satu data mahasiswa.
5. Tampilkan data mahasiswa dan fakultas. Ambil berdasarkan nim, nama mahasiswa, nama fakultas.
6. Hapus data mahasiswa yang memiliki nim : 13
7. Ubah data berupa alamat : solo dan no.telp : 0857688788 yang memiliki nim 12
8. Tampilkan data mahasiswa dan urutkan dari besar ke kecil berdasarkan nim mahasiswa. Kemudian tampilkan data fakultas urutkan secara *ascending* berdasarkan kolom ke-2.
9. Tampilkan seluruh data mahasiswa beserta seluruh data fakultasnya.
10. Tampilkan seluruh mahasiswa yang berada di fakultas "SAINTEK".
11. Tampilkan seluruh mahasiswa yang tidak berada di fakultas "SAINTEK".

Next week ***Operator Dasar di dalam MariaDB***