

## BAB III

### PERINTAH DASAR SQL

#### 3.1 Bahasan dan Sasaran

##### 3.1.1 Bahasan

- Perintah-perintah dasar dalam standar query sql
- Query untuk menampilkan, menambah, menghapus dan memanipulasi data dalam tabel.

##### 3.1.1 Sasaran

1. Mahasiswa memahami perintah dasar dalam standar query sql.
2. Mahasiswa memahami cara menambah, menampilkan, manghapus dan memanipulasi data.

#### 3.2 Materi

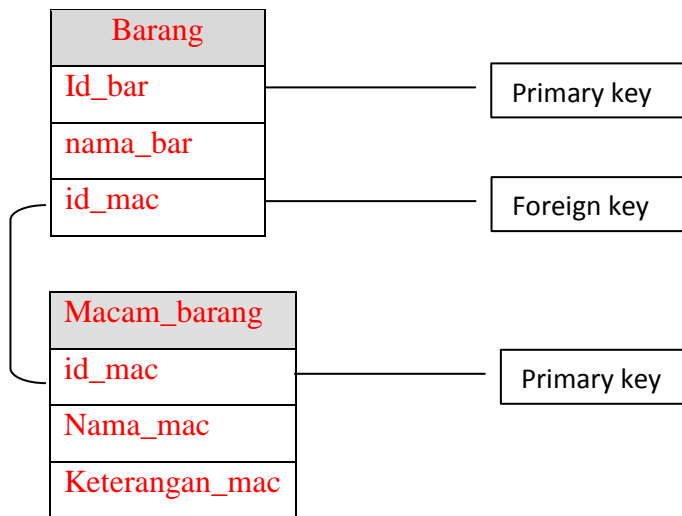
##### 3.2.1 Hubungan Antar Tabel (Reference)

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));**

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 :

```
bab3=# select nama_mhs,tgl_mhs from mahasiswa;
nama_mhs | tgl_mhs
-----+-----
 tutik    | 2000-02-12
 doni     | 2002-12-28
 jojo     | 2001-11-02
 rolan    | 2004-01-01
(4 rows)
```

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

Contoh :

```
bab3=# select * from mahasiswa;
nim_mhs | nama_mhs | tgl_mhs | kota
-----+-----+-----+-----
      1 | tutik    | 2000-02-12 | blitar
      2 | doni     | 2002-12-28 | sumenep
      3 | jojo     | 2001-11-02 | sumenep
      4 | rolan    | 2004-01-01 | semarang
(4 rows)
```

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 :

```
bab3=# select nama_mhs, tgl_mhs from mahasiswa where kota ='sumenep';
nama_mhs | tgl_mhs
-----+-----
 doni     | 2002-12-28
 jojo     | 2001-11-02
(2 rows)
```

Atau

```
bab3=# select * from mahasiswa where kota ='sumenep';
nim_mhs | nama_mhs | tgl_mhs | kota
-----+-----+-----+-----
      2 | doni     | 2002-12-28 | sumenep
      3 | jojo     | 2001-11-02 | sumenep
(2 rows)
```

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 :

```
bab3=# select * from mahasiswa;
 nim_mhs | nama_mhs | ttl_mhs | kota | id_fk
-----+-----+-----+-----+-----
      1 | tutik    | 2000-02-12 | blitar | 1
      4 | rolan    | 2004-01-01 | semarang | 1
      2 | doni     | 2002-12-28 | sumenep | 2
      3 | jojo     | 2001-11-02 | sumenep | 3
(4 rows)

bab3=# select * from fakultas
bab3=# ;
 id_fk | nama_fk
-----+-----
      1 | saintek
      2 | psikologi
      3 | tarbiyah
(3 rows)

bab3=# select nama_mhs,kota,nama_fk from mahasiswa, fakultas where mahasiswa.id_
fk=fakultas.id_fk;
 nama_mhs | kota | nama_fk
-----+-----+-----
 tutik    | blitar | saintek
 rolan    | semarang | saintek
 doni     | sumenep | psikologi
 jojo     | sumenep | tarbiyah
(4 rows)
```

Perintah SQL diatas akan menampilkan nim mahasiswa, nama mahasiswa, nomor fakultas, dan nama fakultas. Pada perintah setelah WHERE terdapat 2 nama kolom **id\_fk**, hal tersebut dimaksudkan penyamaan antara *primary key* milik dari tabel fakultas dengan *foreign key* milik dari tabel mahasiswa.

### 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 :**

```
bab3=# select * from fakultas;
 id_fk |  nama_fk
-----+-----
      1 | saintek
      2 | psikologi
      3 | tarbiyah
      4 | humaniora
(4 rows)

bab3=# delete from fakultas where id_fk=4;
DELETE 1
bab3=# select * from fakultas;
 id_fk |  nama_fk
-----+-----
      1 | saintek
      2 | psikologi
      3 | tarbiyah
(3 rows)
```

### 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 :**

```
bab3=# select * from fakultas;
 id_fk |  nama_fk
-----+-----
      1 | saintek
      2 | psikologi
      3 | tarbiyah
      4 | humaniora
(4 rows)

bab3=# update fakultas set nama_fk='ekonomi' where id_fk=4;
UPDATE 1
bab3=# select * from fakultas;
 id_fk |  nama_fk
-----+-----
      1 | saintek
      2 | psikologi
      3 | tarbiyah
      4 | ekonomi
(4 rows)
```

### 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*.

```
bab3=# select * from fakultas;
 id_fk |  nama_fk
-----+-----
      1 | saintek
      2 | psikologi
      3 | tarbiyah
      4 | ekonomi
(4 rows)

bab3=# select * from fakultas order by nama_fk;
 id_fk |  nama_fk
-----+-----
      4 | ekonomi
      2 | psikologi
      1 | saintek
      3 | tarbiyah
(4 rows)
```

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 :

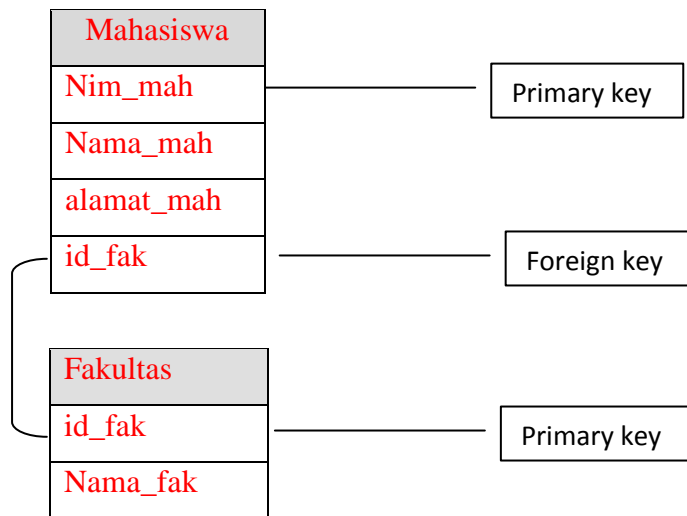
**select \* from nama\_tabel order by nama\_kolom asc, nama\_kolom desc;**

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



1. Gunakan tabel mahasiswa, tambahkan kolom yang diperlukan kemudian masukkan 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

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

### **Tugas Rumah**

1. Buat laporan, Tampilkan perintah-perintah SQL menggunakan DBMS MYSQL beserta hasilnya di blog anda. Perintahnya sesuai dengan tugas praktikum 1-11.
2. Jelaskan evaluasi dari DBMS MySQL dan PostgreSQL dalam blog anda berdasarkan tugas praktikum 1-11. (evaluasi meliputi perbedaan atau kesamaan, kelebihan atau kekurangan, tambahan dari penulis)