**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;**

|  |
| --- |
| Barang  Primary key |
| Id\_bar |
| nama\_bar  Foreign key |
| id\_mac |

|  |
| --- |
| Macam\_barang |
| id\_mac  Primary key |
| Nama\_mac |
| Keterangan\_mac |

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’);**

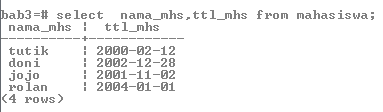
Berikut penjelasan :

* 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

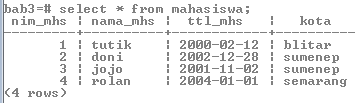
Dalam database, perintah SQL untuk menampilkan data sebuah tabel bisa menggunakan SELECT. Berikut struktur SQL untuk penampilan data berdasarkan kolom tertentu :

**Select nama\_kolom from nama\_tabel;**

Contoh :   


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

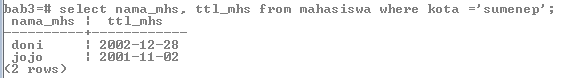
Contoh :



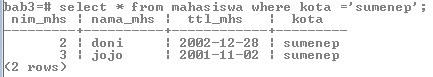
Kemudian, bila ingin menampilkan berdasarkan baris tertentu bisa menggunakan perintah WHERE yang diletakkan dibelakang nama tabel dan isikan nama kolom beserta kata kunci sebagai dasar atau syarat dari pencarian baris. Untuk lebih jelasnya berikut struktur perintah SQL nya :

**select nama\_kolom from nama\_tabel where nama\_kolom = kata kunci;**

Contoh :



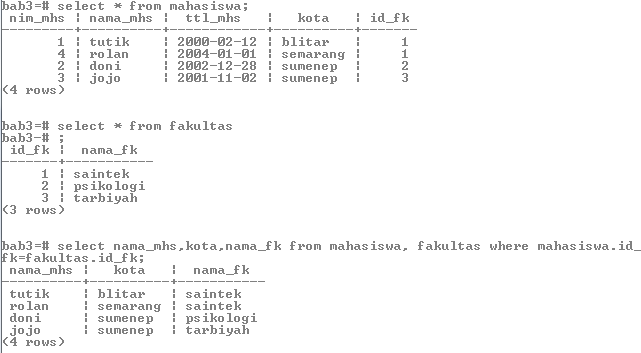
Atau



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 :



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

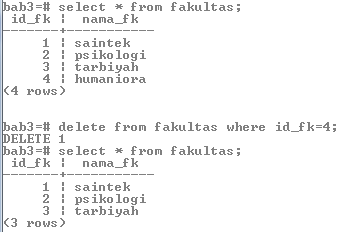
Pada database PostgreSQL, DELETE digunakan untuk menghapus data pada sebuah tabel. Berikut Struktur yang digunakan :

**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 :**

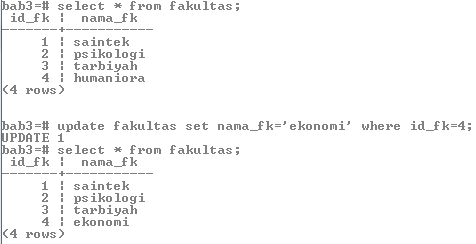
****

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 :**

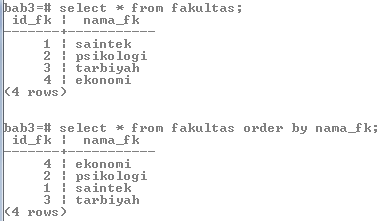
****

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.*

****

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**.