**BAB IX**

**Komponen Query Atau SQL (Strutured Query Language)**

# Sekilas Mengenai SQL

**SQL** adalah bahasa standar untuk query yang difungsikan untuk memanipulasi suatu data pada Database. Hal itu meluputi DDL (*Data Definition Language)* meliputi pembuatan Data Base ataupun pembuatan tabel dan DML (*Data Manipulation Language)* meliputi perintah – perintah standar query.

# Dasar – Dasar mengenai Metode SQL

## DDL (Data Defenition Language)

a. Membuat Tabel

Pembuatan tabel dengan perintah SQL dapat dilakukan dengan perintah Create diikuti dengan nama tabel dan field yang dibutuhkan

# Bentuk Umum

Create table nama\_tabel ( Nm\_filed1 tipedata1, Nm\_field2 tipedata2,

…………………..

…………………… nm\_fieldN tipedataN

);

# Contoh

Create table barang (

Kdbarang varchar(5) not null primary key, Nmbarang varchar(15)

);

* + 1. **DML (*Data Manipulation Language)***

# Metode Select

Metode Select digunakan untuk menampikan dan memilih suatu data dengan kondisi ataupun syarat yang sudah ditentukan dari satu atau beberapa tabel sekaligus dalam satu data base.

# Select Tanpa Syarat

Merupakan perintah pencarian data tanpa diikuti perintah dengan kondisi ataupun persyaratan.

# Bentuk Umum

Select *Daftar Field* From *Nama Tabel*

# Contoh

Select \* From barang

Perintah tersebut adalah digunakan untuk menampilkan semua data pada tabel barang. Untuk menampilkan data pada tabel dengan tidak menampilkan semua field pada tabel dengan menggunakan perintah **Select** diikuti dengan daftar field.

# Contoh

Select kdbarang, nmbarang From barang

Perintah yang digunakan untuk menampilkan data Kode Barang dan Nama Barang pada tabel Barang.

# Select Dengan Syarat

Adalah perintah menampilkan data yang diikuti dengan kondisi yang harus terpenuhi.

# Bentuk Umum

Select *daftar\_field* From *nama\_tabel* Where *kondisi/persyaratan*

# Contoh

Select \* From barang where kdbarang = ‘A001’;

Adalah menampilkan semua data barang dengan kode barang = A001

Select \* From barang where stok < 100;

Adalah menampilkan semua data barang dengan kondisi stok yang lebih kecil dari 100.

Select \* From barang where hargabeli < = 2000 and Stok >= 5

Adalah menampilkan seluruh data pada tabel barang untuk harga beli dibawah atau sama dengan 2000 dan Stok diatas sama dengan 5.

Select \* From barang where Harga\_beli Between 5000 And 10000

Adalah semua data pada tabel barang dengan harga beli diatara 5000 dan 10000

Select \* From barang where nmbarang like ‘mie%’

Adalah menampilkan seluruh data dari tabel barang dengan nama barang yang berawalan mie

Select \* From barang where nmbarang Like ‘%mie’

Adalah menampilkan data dari tabel barang dengan nama barang yang berakhiran dengan kata mie.

# Mengurutkan Data (Order By)

Fungsi ini digunakan untuk menurutkan data berdasarkan kondisi tertentu terhadap hasil Query.

# Bentuk Umum

Select *daftar\_field* From *nama\_tabel* order by *nama\_field metode\_pengurutan*

# Contoh

Select \* From barang order by nmbarang ASC

Menampilkan seluruh data barang diurutkan berdasarkan nama barang secara Ascending

# Mengelompokan Data (Group By)

Fungsi ini digunakan untuk mengelompokan data berdasarkan field terpilih.

# Bentuk Umum

Select \* From barang Group By kdbarang

# Contoh

Select \* From barang group by kdbarang

Menampilkan seluruh data barang dengan dikelompokan berdasarkan kode barang.

# Fungsi Angregasi

Fungsi Agregasi adalah fungsi matematika yang digunakan bersamaan dengan perintah

**Select.** Berbagai macam agregasi yang digunakan bersamaan dengan peritah **Select**

|  |  |
| --- | --- |
| **Fungsi** | **Kegunaan** |
| Count | Untuk memperoleh jumlah record hasil Query |
| Sum | Untuk memperoleh total nilai dari suatu field |
| Avg | Untuk memperoleh nilai rata – rata |
| Max | Untuk memperoleh nilai terbesar |
| Min | Untuk memperoleh nilai terkecil |

# Bentuk Umum

**Select** *Fungsi\_agregasi (nama\_field)* **From** *nama\_tabel*

# Contoh

**Select Sum(**stok) **From** barang

# Query Untuk banyak Tabel

Adalah fungsi Query yang digunakan untuk menampilkan lebih dari satu tabel

# Bentuk Umum

**Select** *index1.daftar\_field\_tabel1, index2.daftar\_field\_tabel2* **From** *tabel1 index1, tabel2 index2* **Where** *index2.****tabel1 = index1.tabel2***

Atau

**Select Distinct** *tabel1.daftar\_field, tabel2.Daftar\_field* **From *tabel1, tabel2* Where**

*tabel1.nama\_field = tabel2.nama\_field*

# Contoh

**Select Distinct** *barang.nmbarang, barang.harga\_bel, pemasok.nmpemasok, pemasok.alamat* **From** *Pemasok, barang* **Where *barang.kdpemasok = pemasok.kdpemasok.***

Perintah tersebut akan menampilkan nama barang dari tabel barang, harga beli dari tabel barang, nama pemasok dari tabel pemasok, dan alamat dari tabel pemasok dari tabel pemasok dan tabel barang dengan kondisi dimana kdpemasok di tabel barang sama dengan kode pemasok pada tabel pemasok.

# Manipulasi Data

Perintah SQL yang digunakan untuk memanipulasi data pada sebuah tabel. Hal ini meliputi = menambah data, mengedit data ataupun menghapus data.

Ada tiga perintah yang sering digunakan untuk perintah SQl dalam hal memanipulasi Data

# Insert

Perintah SQl yang digunakan untuk menambahkan data pada tabel. Bentuk umum penulisan perintah **Insert.**

**Insert Into** *nama\_tabel (field1, field2, field3,…….fieldn)*

## Values (nilai1, nilai2, nilai3, nilain)

Yang perlu diperhatikan adalah jumlah semua field dengan jumlah nilai yang akan dimasukan adalah sama. Untuk field dengan tipe Alpha (string) maka diantara nilai yang diinput diberikan dengan tanda kutif (‘ ‘).

# Contoh

**Insert Into** *barang(kdbarang, nmbarang, hargabeli, hargajual, stok)*

**Values** *(‘A001’,’Indomei rasa Soto’,20000,30000,23)*

# Update

Perintah SQL yang digunakan untuk mengedit data yang sudah ada sebelumnya pada tabel.

**Update *nama\_tabel***

**Set** *field1 = nilai1, field2=nilai2, field3 = nilai3, fieldn = nilain)*

**Where *syarat***

**Contoh Update** barang

**Set** kdbarang = ‘A001’, nmbarang = ‘Idomie rasa Ayam’, hargabeli = 2000, hargajual = 3000, stok = 34 **where** kdbarang = ‘A002’

Perintah SQl tersebut diatas adalah mengganti seluruh data yang ada ditabel barang untuk kode barang = A002 menjadi A001.

# Delete

Perintah SQL digunakan untuk menghapus data yang ada ditabel. Bentuk umum penulisan perintah **Delete.**

**Delete From** *nama\_tabel* **where** *syarat* **(**digunakan untuk menghapus record dengan kondisi yang diinginkan).

# Atau

**Delete From** *nama\_tabel* (digunakan untuk menghpus seluruh record pada tabel terpilih)

# Contoh

**Delete From** barang **where** kdbarang = ‘A001’

Merupakan kondisi dimana ada perintah untuk menghapus data barang untuk kode barang = A001

# 8.2.3. Penerapan Konsep SQL pada Aplikasi

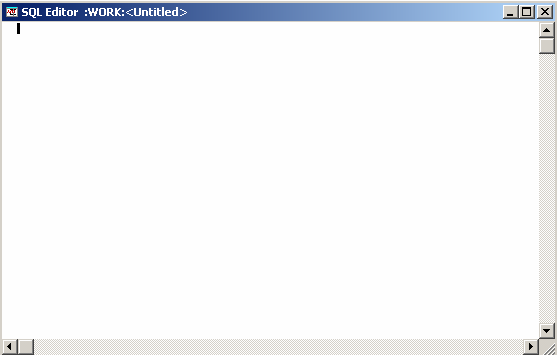
Penerapan konsep pemograman SQl pada Delphi dapat kita implementasikan dengan dua konsep yaitu dengan konsep Data Base Dekstop dan Konsep pemograman

# 9.2.3.1 Konsep Database Dekstop

Penerapan konsep ini dengan memanfaatkan jendela dari data base Dekstop dan SQL File.

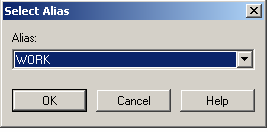
# Langkah - langkahnya

1. Aktifkan Data Base Dekstop
2. Dari Menu **File | New | SQL File**. Jendela SQL Editor akan ditampilkan



Gambar 10.1 Jendela SQL Editor

1. Dari Menu **SQl** pilih **Select Alias (**untuk tabel yang ada dalam folder penyimpanan), sehingga pada layer akan ditampilkan jendela **Select Alias.**



Gambar 9.2. Jendela Select Alias

1. Dari jendela Select Alias pilih dan aktifkan nama alias yang sudah dibuat sebelumnya

**(TOKO) .** Setelah selesai klik **OK.**

1. Pada jendela editor tulisakn perintah SQL dan jalankan dengan memilih icon **Run SQL.**

# Untuk mencoba kita gunakan beberapa kasus berikut ini

1. Menampikan semua data barang

# Sintaks Penulisan

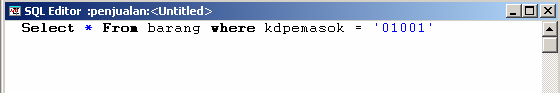


**Hasil**

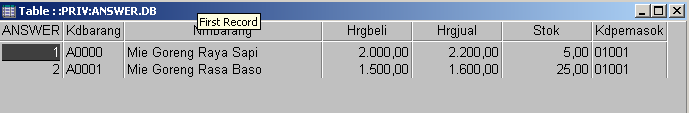


1. Menampilkan Semua Data barang dengan kode pemasok = 01001

# Sintaks Penulisan



**Hasil**

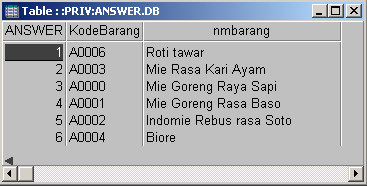


1. Menampilkan kode barang dan nama barang dari tabel barang dengan pengurutan secara descending berdasar nama barang serta mengganti judul kdbarang menjadi KodeBarang.

# Sintaks Penulisan

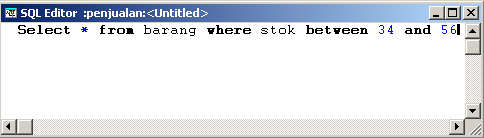


**Hasil**



1. Menampilkan Semua Data barang untuk stok diantara 34 sampai dengan 56

# Sintaks Penulisan



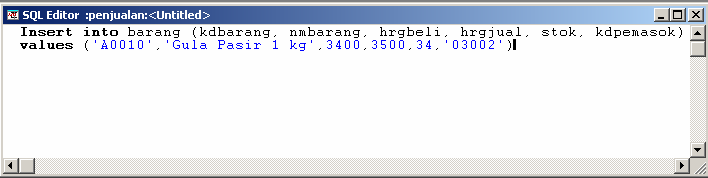
**Hasil**



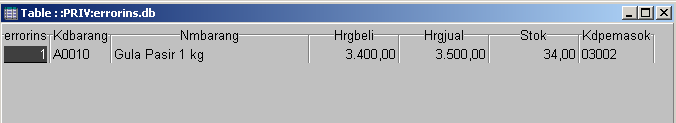
1. Menambahkan data barang dengan data sebagai berikut =

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kdbarang | Nmbarang | Harga Beli | Harga Jual | Stok | Kdpemasok |
| A0010 | Gula Pasir 1 Kg | 3400 | 3500 | 34 | 03002 |
|  |  |  |  |  |  |

# Sintaks Penulisan



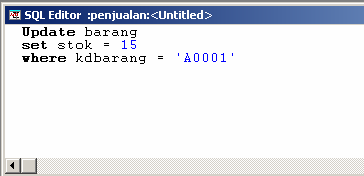
**Hasil**



Untuk melihat semua data gunakan perintah menampilkan data seluruhnya

1. Mengganti Stok barang untuk kode A001 menjadi 15

# Sintaks Penulisan



**Hasil**

