

<http://fajardelphiscript.blogspot.com>

**WIKI**

**WIKI**

**with** **WIKI** **on**

**WIKI**

Author:Fajar Khairil

Copyright (C) fajardelphiscript 2009

## ***Membuat Database dengan SQL di Delphi***

E-Book ini merupakan kelanjutan dari E-Book yang sebelumnya yaitu E-Book Membuat database dengan Delphi. Jika E-Book yang lalu pemanipulasian masih menggunakan table di E-book ini anda akan mempelajari memanipulasi database dengan perintah-perintah SQL ☺, mulai dari Perkenalan dengan SQL, menggunakan perintah SQL di Delphi, sampai Contoh Program nya, kurang baik apa saya coba ☺..... Lanjutin bacanya ya...

### **SEKILAS SQL(Structured Query Language)**

SQL adalah Singkatan dari (Structured Query Language). SQL adalah bahasa yang digunakan untuk berkomunikasi dengan RDBMS(Relational Database Management System). SQL digunakan untuk membuat objek-objek database dan memanipulasi data pada database.

SQL pertama kali dikembangkan pada tahun 1970`an di laboratorium IBM-San Jose California Pertama kali dikembangkan sebagai bahasa di produk database DB2 yang sampai saat ini merupakan produk database andalan IBM. SQL sering di lafalkan dengan "sequel". SQL selanjutnya berkembang menjadi berbagai macam dan bentuk. Setiap Vendor RDBMS seperti Ms. Access dkk mengembangkan versi SQL nya masing-masing. beberapa standart SQL yang diterima luas adlah SQL ANSI 92 dan SQL ANSI 99.

Perintah-perintah SQL dikelompokan menjadi 2 bagian penting yaitu DML(Data Manipulation Language) dan DDF(Data Definition Language).

**DDF**→ merupakan perintah SQL yang berhubungan dengan pembentukan objek-objek database, seperti Triger, Stored Procedure de-el.

**DMF**→ merupakan perintah SQL yang berhubungan dengan manipulasi data seperti memperoleh data, memasukan, mengedit, mengupdate, dan menghapus

data.permintaan/penggunaan perintah SQL pada suatu database disebut dengan *Query*.

Query-Query Dasar:

\*SELECT, untuk memperoleh data dari tabel.

\*INSERT, untuk menambah data ke tabel.

\*UPDATE, untuk mengupdate data yang ada pada tabel.

\*DELETE, untuk menghapus suatu record pada tabel.

## PERINTAH-PERINTAH SQL

\*Query dasar

Perintah SELECT digunakan untuk menampilkan field-field tertentu dari suatu tabel.

Contoh: (contoh ini menggunakan database DBDMOS pada C:\Program Files\Common Files\Borland Shared\Data\dbdemos.mdb)

```
Select neme, capital from country;
```

Ket: name dan capital adl nama field dan country adalah nama tabel.

Untuk memperoleh semua data yang ada pada tabel, contoh:

```
Select * from country;
```

Diberikan tanda asterik setelah select.

\***Memfilter Data**

Menggunakan perintah WHERE untuk memperoleh data yang diinginkan.

Contoh untuk menampilkan data semua negara yang memiliki jumlah penduduk lebih dari 100 juta:

```
Select* from country where population>100000000;
```

Gunakan operator Logika dan tanda kurung secukupnya untuk menggabungkan beberapa kriteria.

```
Select * from country where (population>=100000000) and  
(population<=1000000000);
```

### \*Mengurutkan Data

perintah ORDER BY digunakan untuk mengurutkan data berdasarkan satu field atau lebih.

```
Select * from country order by asc;
```

Asc(Ascending A→Z) Desc(Descending Z→A).

### \*Operator Query

#### Operator Query

Gunakan Operator Like untuk memfilter record yang nilai suatu field nya berawalan,berakhiran atau mengandung kata/huruf tertentu.

Perintah berikut adalah untuk memilih negara yang memiliki huruf awal "C".

```
Select * from country where name like "C%";
```

Perintah berikut adalah untuk memilih negara yang memiliki huruf akhir "C".

```
Select * from country where name like "%C";
```

Perintah berikut untuk memilih negara yang mengandung kata "agu".

```
Select * from country where name like "%agu%";
```

Perhatikan perbedaan 3 contoh di atas!

### \*Operator BETWEEN

Operator BETWEEN digunakan untuk membandingkan apakah suatu nilai berada dalam suatu range nilai. Contoh:

```
Select * from country where population between 1000000 and 2000000;
```

### \*Operataor IN

Operator In digunakan untuk melihat apakah suatu nilai berada dalam suatu list nilai.

```
Select * from country where capital in ('Lima','Quito');
```

### \*Operator Is Null

Operator Is Null digunakan untuk membandingkan apakah suatu nilai bernilai *Null* atau 0.

```
Select * from country where population is Null;
```

Menampilkan data yang populasinya Null/0, nah untuk menampilkan yang tidak bernilai null cukup ditambahkan **not**.

```
Select * from country where population is not Null;
```

### **\*Aggregate**

SQL mempunyai beberapa buah fungsi aggregate untuk keperluan tertentu yaitu:

- **Sum()**, untuk menghitung jumlah suatu field dalam suatu record set.
- **Avg()**, untuk menghitung rata-rata dari suatu field.
- **Min()**, untuk mencari nilai minimum dari sebuah field.
- **Max()**, untuk mencari nilai maximum dari sebuah field.
- **Count()**, untuk menghitung jumlah record dalam suatu record set.

Semua fungsi diatas membutuhkan suatu parameter yaitu nama field yang akan dikalkulasi. Khusus untuk fungsi aggregate **Count()**, parameternya dapat sebuah field atau sebuah *wildcard*.

Contoh berikut menghitung jumlah gaji pokok yang dibayarkan untuk sebuah Departemen pada bulan dan tahun tertentu.

```
Select sum(gajipokok) as jumlahgajipokok from gaji where departemen='marketing'  
and bulan=11 and tahun=2004;
```

### **\*GROUPING**

query **Group by** dapat digunakan untuk mengelompokkan record berdasarkan field-field tertentu. langsung aja ke contohnya biar lebih jelas:

```
select departmen, sum(gajipokok) as JumlahGajiPokok, avg(gajipokok) as  
RatarataGajiPokok from gaji where bulan=11 and tahun=2004 group by departmen;
```

hasil query di atas, menghasilkan tabel seperti ini

| Departmen   | JumlahGajiPokok | RatarataGajiPokok |
|-------------|-----------------|-------------------|
| Mangemen    | 27.000.000      | 5.000.000         |
| Operasional | 52.000.000      | 3.250.000         |
| Personalia  | 18.000.000      | 3.000.000         |
| Marketing   | 43.000.000      | 3.500.000         |

### \*Join

Anda dapat melakukan Query data yang berasal lebih dari satu tabel.record-record dari sebuah tabel harus di relasikan dengan record-record tabel lain agar dapat menghasilkan hasil yang diharapkan.

Berikut contoh 3 buah tabel yang akan direlasikan

#### Tabel Karyawan

| IDKaryawan | NamaKaryawan | IDDept | IDStatus |
|------------|--------------|--------|----------|
| 001        | Andi         | A      | T        |
| 002        | Budi         | B      | T        |
| 003        | Iwan         | B      | K        |
| 004        | Sari         | D      | T        |
| 005        | Wati         | D      | K        |

#### Tabel Departemen

| IDDept | NamaDept    |
|--------|-------------|
| A      | Managemen   |
| B      | Operasional |
| C      | Personalia  |
| D      | Marketing   |

#### Tabel Status

| IDStatus | KetStatus      |
|----------|----------------|
| T        | Karyawan Tetap |

|   |                  |
|---|------------------|
| K | Karyawan Kontrak |
|---|------------------|

Perhatikan Contoh Query berikut!

**Select**

**karyawan.idkaryawan, karyawan.namakaryawan, departemen.iddept, departmen.namadept from karyawan, departmen;**

Anda juga dapat membuat Alias untuk menyederhanakan penulisan, contoh

**Select k.idkaryawan, k.namakaryawan, d.iddept, d.namadept from karyawan k, departmen d;**

Hasil Query di atas kira-kira akan menghasilkan tabel berikut:

| IDKaryawan | NamaKaryawan | IDDept | IDDept2 | NamaDept    |
|------------|--------------|--------|---------|-------------|
| 001        | Andi         | A      | A       | Managemen   |
| 002        | Budi         | B      | A       | Managemen   |
| 003        | Iwan         | B      | A       | Managemen   |
| 004        | Sari         | D      | A       | Managemen   |
| 005        | Wati         | D      | A       | Managemen   |
| 001        | Andi         | A      | B       | Operasional |
| 002        | Budi         | B      | B       | Operasional |
| 003        | Iwan         | B      | B       | Operasional |
| 004        | Sari         | D      | B       | Operasional |
| 005        | Wati         | D      | B       | Operasional |
| 001        | Andi         | A      | C       | Personalia  |
| 002        | Budi         | B      | C       | Personalia  |
| 003        | Iwan         | B      | C       | Personalia  |
| 004        | Sari         | D      | C       | Personalia  |
| 005        | Wati         | D      | C       | Personalia  |
| 001        | Andi         | A      | D       | Marketing   |

|            |             |          |          |                  |
|------------|-------------|----------|----------|------------------|
| 002        | Budi        | B        | D        | Marketing        |
| 003        | Iwan        | B        | D        | Marketing        |
| <b>004</b> | <b>Sari</b> | <b>D</b> | <b>D</b> | <b>Marketing</b> |
| <b>005</b> | <b>Wati</b> | <b>D</b> | <b>D</b> | <b>Marketing</b> |

Contoh diatas menghasilkan kombinasi pasangan setiap record tabel karyawan dan record tabel departmen.

Jumlah record yang dihasilkan adalah=Jumlah record di tabel karyawan dan Jumlah record di tabel Departmen. Tabel di atas akan memperjelas pembahsan selanjutnya mengenai Join.

### Inner Join

Query yang melibatkan 2 tabel seperti contoh sebelumnya tentu jurang bahkan tidak berguna karena menampilkan semua bentuk kombinasi pasangan setiap record dari kedua tabel. agar hasil query lebih efisien, kita harus mendefinisikan relasi antar 2 tabel tsb. contoh

```
Select k.idkaryawan,k.namakaryawan,d.iddept,d.namadept from karyawan  
k,departmen d where k.iddept=d.iddept;
```

Dengan query di atas akan menghasilkan tabel sbb:

| IDKaryawan | NamaKaryawan | IDDept | NamaDept    |
|------------|--------------|--------|-------------|
| 001        | Andi         | A      | Managemen   |
| 002        | Budi         | B      | Operasional |
| 003        | Iwan         | B      | Operasional |
| 004        | Sari         | C      | Personalia  |
| 005        | Wati         | D      | Marketing   |

Karyawan yang bernama "Wati" tidak termasuk record set karena iddept nya bernilai null atau kosong. contoh di atas disebut *Inner equi -join*.



## \*Manipulasi Data

### Insert

Insert digunakan untuk menambah data ke suatu tabel. contoh nya:

```
Insert into country(name,capital,continent) values('Indonesia','Jakarta','Asia');
```

Maka record pada tabel akan bertambah dengan nilai-nilai yang berada pada values di atas.

### Update

Update digunakan untuk mengubah isi dari record yang akan di update seperti kondisi setelah query where pada query select.

Contoh:

```
Update country set population=20000000000 where name='Indonesia';
```

### Delete

Seperti namanya query delete ya gunanya untuk menghapus suatu record dari tabel. contoh:

```
Delete from country where continent='North America';
```

Setelah anda mengetahui fungsi-fungsi dan query-query dari SQL, kini anda dan saya akan mempelajari bagaimana sih penggunaannya di Delphi+Contoh Program nya ☺.

## CONTOH PROGRAM nya di DELPHI

Untuk mempersingkat halaman dan waktu langsung aja ya ga usah basa-basi. bikin interface yang kaya gini:



untuk componentnya udah pada tau kan??kalo gak tau kebangetan!!!

Tambain ADOConnection,2 ADOQuery pada palette ADO,DataSource pada palette DataAccess,DBGrid dan DBNavigator pada palette DataControls yang lainnya pasti tau deh,atur jadi kaya di atas ya....info 2 component ADO Query agar mempermudah pemanipulasian data dan Query untuk DDL dan DML.

Untuk database nya kita pake aja database dbdemos yang terdapat di  
**C:\Program Files\Common Files\Borland Shared\Data\dbdemos.mdb**

Tentunya anda harus punya delphi dulu.Langkah-langkah:

- sedikit tips dari saya agar lebih mudah dan mempersingkat penulisan coding dan penulisan saya di E-Book ini ☺ ubah nama ADOConnection dan ADOQuery menjadi con untuk ADOConnection dan query dan query1 untuk ADOQuery.untuk selanjutnya ADOConnection akan disebut con dan ADOQuery akan disebut query so..biasakan
- double click pada con→click Build pilih *Microsoft.Jet.OLEDB.4.0* lalu hubungkan dengan database yang berada di directory yang saya sebutkan diatas.
- Atur Properties sbb:

| Component | Properties | Nilai |
|-----------|------------|-------|
|-----------|------------|-------|

|             |              |             |
|-------------|--------------|-------------|
| Con         | Login Prompt | False       |
| Query       | Connection   | Con         |
| Query1      | Connection   | Connection  |
| Datasource  | Connection   | Query       |
| DBGrid      | DataSource   | DataSource1 |
| DBNavigator | DataSource   | DataSource1 |

-Pada Project seperti ini akan banyak perocedure yang menggunakan procedure yang sama,agar lebih efisien maka dibuat 2 Procedure yang kita definisikan sendiri,yaitu procedure *Show* untuk untuk menampilkan semua record dari database dan procedure *Delete* untuk menghapus Record.Masuk Code Editor atau pencet aja F12..pada bagian *{Public Declaration}* ketikan code berikut:

```
private
{ Private declarations }
public
{ Public declarations }
Procedure delete;
procedure show;
end;
```

-Lalu di bawah Implementation *{\$R \*.dfm}* ketikan Procedurennya sbb: implementation

```
{$R *.dfm}
procedure TForm1.delete;
Begin
if messagedlg('yakin akan menghaspus record ini?',mtconfirmation,
[mbYes,mbCancel],0)=mrYes then
begin
con.BeginTrans;//memulai transaksi
with query1 do begin
sql.Clear;
sql.Text:='delete from country where name=:parName';
parameters.ParamByName('parName').Value:=query.fieldbyname('name'
).AsString;
execsql;//eksekusi perintah
end;
```

```
query.Requery();//mengembalikan semula query  
con.CommitTrans;mengakhiri transaksi  
end;  
end;
```

```
Procedure TForm1.show;  
Begin  
with query do begin  
sql.clear;//selalu menghapus query yang masih ada jika ada  
sql.Text:='select * from country order by Name asc';  
open;//membuka query  
end;  
end;
```

-Double click pada bagian form maka akan muncul code editornya tambahkan code berikut:

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
con.connection:=true;//menghubungkan dengan database  
end;
```

-pada form1 di bagian yang kosong double click button "tambah" ketikan "form2.showmodal;" tanpa tanda kutip.

pencet F12

-tambahkan form baru dan atur jadi kaya gini:



-pada form2 double click button "tambahkan" dan ketikan code berikut:

```

procedure TForm2.Button1Click(Sender: TObject);
begin
form1.con.BeginTrans;
with form1.query do begin
sql.Clear;
sql.Text:='insert into
country(Name,Capital,Continent,Area,Population)'+

'Values(:parName,:parCapital,:parContinent,:parArea,:parPopulatio
n)';
prepared;//mempersiapkan
parameters.ParamByName('parName').Value:=Edit1.text;
parameters.ParamByName('parCapital').Value:=Edit2.text;
parameters.ParamByName('parContinent').Value:=Edit3.text;
parameters.ParamByName('parArea').Value:=Edit4.text;
parameters.ParamByName('parPopulation').Value:=Edit5.text;
execsql;//untuk mengeksekusi perintah SQL
form1.show;
end;
form1.con.CommitTrans;
button1.Enabled:=false;
end;

```

perintah DML(Data Manipulation Language)jika ingin di kerjakan harus di beri perintah "execSQL",setelah DML di execute maka SQL akan menutup tabel dengan otomatis untuk itu perlu ditambahkan procedure form1.show yang tadi kita buat: untuk kembali memunculkan tabel.

-masih pada form2,double click button baru ketikan code berikut.

```

procedure TForm2.Button2Click(Sender: TObject);
begin
Edit1.clear;
Edit2.clear;
Edit3.clear;
Edit4.clear;
Edit5.clear;
edit1.SetFocus;
button1.Enabled:=true;
end;

```

-kalo untuk button keluar pasti pada tau kan?Cuma ngetik "Close;"

-Kembali pada form1 double click button Search ketikan code berikut:

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
if edit1.Text='' then  
show else  
with query do begin  
sql.Clear;  
sql.Text:='select* from country where name  
like'+quotedstr('%'+Edit1.Text+'%');  
open;  
end;  
end;
```

-Double Pada Button "Hapus" di Form1,ketikan code berikut:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
delete;  
end;
```

-Nah Beres dah,pencet dech F9!!!



Ya selanjutnya coba aja deh Programnya ☺

Berikut Listing Lengkapnya:

### UNIT 1:

```
unit Unit1;
interface
uses
  windows, Messages, SysUtils, Variants, Classes, Graphics,
  Controls, Forms,
  Dialogs, StdCtrls, Grids, DBGrids, DB, ADODB, ExtCtrls, DBCtrls;

type
  TForm1 = class(TForm)
    query: TADOQuery;
    con: TADOConnection;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    Button1: TButton;
    Label1: TLabel;
    Button2: TButton;
    DBNavigator1: TDBNavigator;
    Edit1: TEdit;
    Button3: TButton;
    Query1: TADOQuery;
    procedure FormCreate(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action:
    TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
    Procedure delete;
    procedure show;
  end;

var
  Form1: TForm1;
```

implementation

```
uses Unit2;
```

```
{$R *.dfm}
```

```
procedure TForm1.delete;
```

```
begin
```

```
if messagedlg('yakin akan menghapus record ini?',mtconfirmation,  
[mbYes,mbCancel],0)=mrYes then
```

```
begin
```

```
con.BeginTrans;
```

```
with query1 do begin
```

```
sql.Clear;
```

```
sql.Text:='delete from country where name=:parName';
```

```
parameters.ParamByName('parName').Value:=query.fieldbyname('name'  
)AsString;
```

```
execsql;
```

```
end;
```

```
query.Requery();
```

```
con.CommitTrans;
```

```
end;
```

```
end;
```

```
Procedure TForm1.show;
```

```
begin
```

```
with query do begin
```

```
sql.clear;//menghapus query yang masih ada jika ada
```

```
sql.Text:='select * from country order by Name asc';
```

```
open;//membuka query
```

```
end;
```

```
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
con.connected:=true;
```

```
Show;
```

```
end;
```



```
procedure TForm1.Button3Click(Sender: TObject);
begin
if edit1.Text='' then
show else
with query do begin
sql.Clear;
sql.Text:='select* from country where name
like'+quotedstr('%'+Edit1.Text+'%');
open;
end;
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
form2.showmodal;
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);
begin
delete;
end;
```

```
procedure TForm1.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
con.Connected:=false;
end;
end.
```

## Unit 2

```
unit Unit2;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms,
```

```
Dialogs, StdCtrls;
```

```
type
```

```
TForm2 = class(TForm)
```

```
Label1: TLabel;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
Label5: TLabel;
```

```
Edit1: TEdit;
```

```
Edit2: TEdit;
```

```
Edit3: TEdit;
```

```
Edit4: TEdit;
```

```
Edit5: TEdit;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Button3Click(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
Form2: TForm2;
```

```
implementation
```

```
uses Unit1;
```

```
{$R *.dfm}
```

```
procedure TForm2.Button1Click(Sender: TObject);
```

```
begin
```

```
form1.con.BeginTrans;
```

```
with form1.query do begin
```

```

sql.Clear;
sql.Text:='insert into
country(Name,Capital,Continent,Area,Population)'+
'Values(:parName,:parCapital,:parContinent,:parArea,:parPopulatio
n)';
prepared;//mempersiapkan
parameters.ParamByName('parName').Value:=Edit1.text;
parameters.ParamByName('parCapital').Value:=Edit2.text;
parameters.ParamByName('parContinent').Value:=Edit3.text;
parameters.ParamByName('parArea').Value:=Edit4.text;
parameters.ParamByName('parPopulation').Value:=Edit5.text;
execsql;//untuk mengeksekusi perintah SQL
form1.show;
end;
form1.con.CommitTrans;
button1.Enabled:=false;
end;

procedure TForm2.Button2Click(Sender: TObject);
begin
Edit1.clear;
Edit2.clear;
Edit3.clear;
Edit4.clear;
Edit5.clear;
edit1.SetFocus;
button1.Enabled:=true;
end;

procedure TForm2.Button3Click(Sender: TObject);
begin
close
end;

end.

```

Gampang buanget kan???

## Biografi Penulis



Fajar Khairil Amin Lahir di Tasikmalaya 2 Maret 1990,Lulus SMA tahun 2008 di SMAN1 Cikatomas-Tasikmalaya ,Melanjutkan ke Bina Sarana Informatika(BSI)Tangerang Satu Semester kemudian pindah ke Universitas Budi Luhur,Senang belajar semua tentang komputer terutama Programing menggunakan bahasa Pemrograman seperti Borland Delphi,, Aktivitas masih belajar. Penulis dapat dihubungi lewat:

E-Mail:[fajar90alone@gmail.com](mailto:fajar90alone@gmail.com)

[fajaralonekhairil@yahoo.com](mailto:fajaralonekhairil@yahoo.com)

URL:<http://www.fajardelphiscript.blogspot.com>