

Bab IX Data Base Dekstop

9.1. Pengertian Data Base Dekstop

Data Base Dekstop adalah merupakan sebuah system aplikasi database yang sudah disertakan pada saat penginstalan Delphi. Komponen Data Base Dekstop yang merupakan bawaan Delphi meliputi = Paradox, dBse, MsSQL, Oracle, Maccess, Excel dan Lain - lain.

9.2. Langkah - Langkah pembuatan Data Base Dekstop

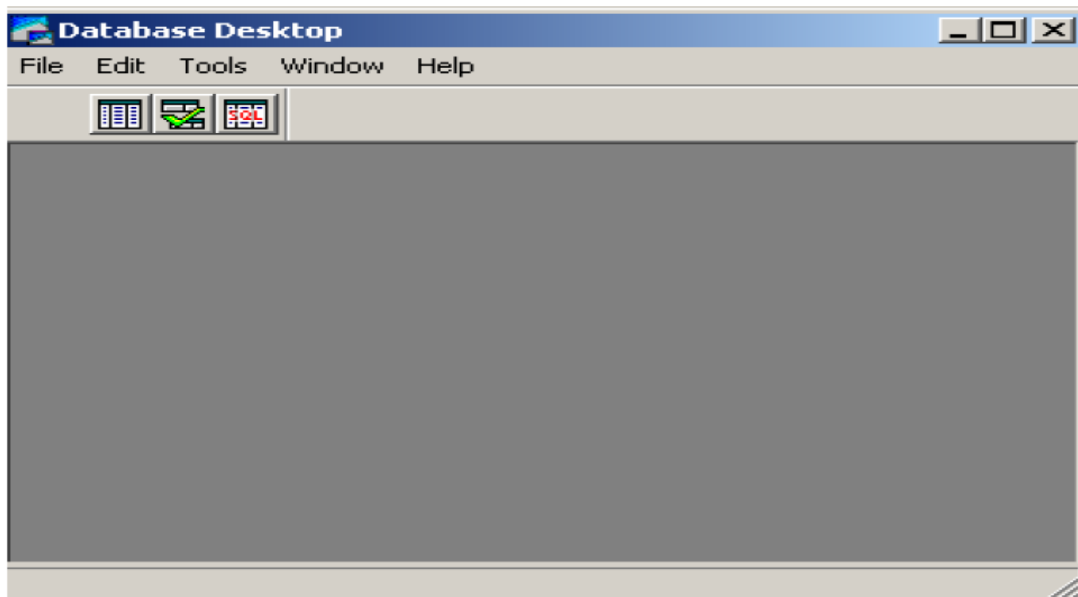
Ada dua cara untuk mengaktifkan database Dekstop

a. Melalui Icon Start

Klik start I Pilih programs I Pilih Borland Delphi I Klik Data Base Dekstop

b. Melalui IDE Delphi

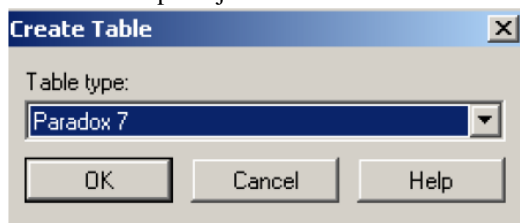
- Pilih dan klik Menu Tools I pilih dan klik Database Dekstop



9.3. Membuat Tabel

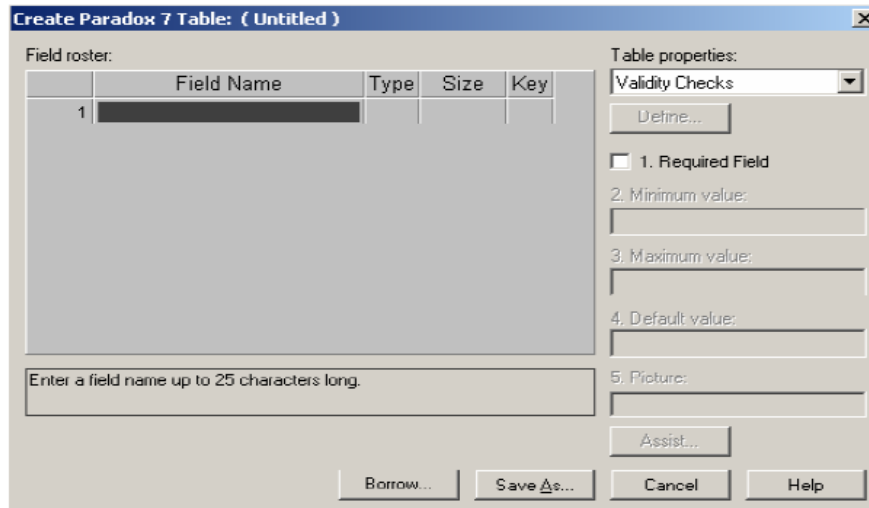
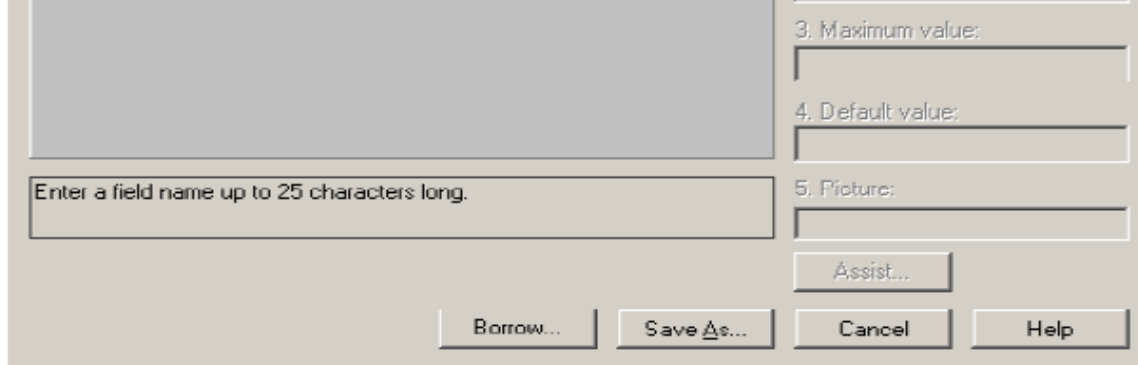
Setelah DBD diaktifkan, langkah berikut untuk membuat tabel:

1. Klik Menu File I New I table
2. Perhatikan tampilan jendela create table



Gambar 9.2 Gambar Create tabel

3. Dari Jendela create table pilih dan klik pilihan table pada table type (mis = Paradox)
4. Klik Ok



Gambar 9.3 Gambar Desain Tabel

Penjelasan Untuk Struktur tabel

Field Name = digunakan untuk menuliskan nama field

Syarat Penulisan Nama Field

- > Unik dan mudah diingat
- > Diawali dengan Karakter
- > Tidak diperbolehkan penulisan dengan tanda baca (! , . ? / + & % # ©)
- > Untuk field dengan dua kata atau lebih dihubungkan dengan Undecscare (_)

Type = Digunakan untuk menentukan tipe data pada Field

Type Data yang Dikenal Pada Paradox adalah

- > Alpha = tipe data yang menampung semua karakter baik berupa huruf, angka maupun tanda baca.
- > Number = tipe data yang menampung angka (numeric), jangkauannya mempunyai range untuk bilangan negative dan bilangan positif, (-10307 sampai dengan 10308). Dengan digit maksimal adalah 15 digit.
- > Money = Tipe data yang hanya untuk angka. Sama seperti dengan Tipe Data *Number* hanya berbeda pada jangkauan. Pada tipe data number tidak mengenal deddsimal, tipe data money mengenal adanya decimal dan format mata uang.
- > Short = tipe data angka yang hanya mempunyai jangkauan -32,767 to 32,767
- > Long Integer = Sama seperti tipe data integer dengan jangkauan lebih luas (-2147483648 to 2147483647)
- > Date = Tipe Data untuk tanggal (1 Januari 9999 BC to 31 Desember 9999 AD).
- > Time = Tipe Data yang digunakan untuk setup waktu.
- > Memo = Tipe Data untuk semua unit karakter dengan jangkauan 1 sampai dengan 240 karakter.
- > Graphic = Tipe Data untuk gambar. (..BMP, .PCX, .TIF, .GIF, and .EPS file formats).
- > Logical = Tipe data yang hanya mengenal benar atau salah (Yes or No).

Size = Digunakan untuk menentukan size terhadap suatu field

Catatan : Size yang kompatibel untuk diganti hanya tipe data Alpha dan memo.

Key = Digunakan untuk menentukan kunci utama (primary key)

Catatan = untuk memberikan primary key cukup ditekan space bar atau double klik pada field yang akan dibuat sebagai key (ditandai dengan *)

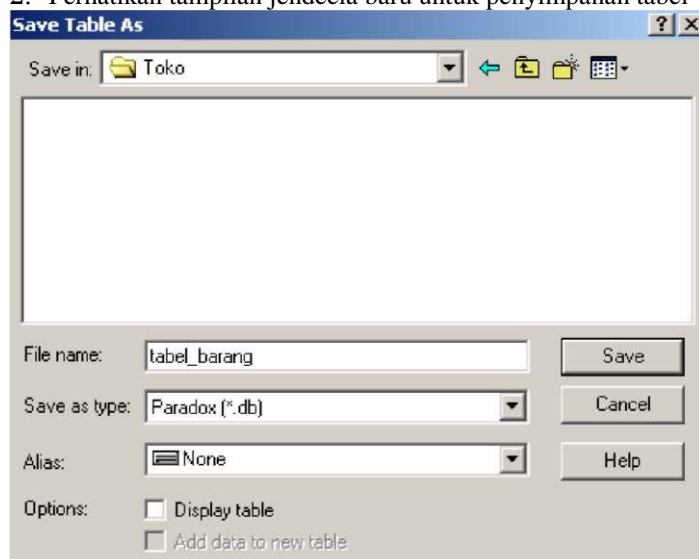
Misal = Buatlah Struktur tabel untuk tabel Mahasiswa seperti berikut ini

Field	Tipe Data	Size	Key
Kdbarang	Alpha	5	*
Nmbarang	Alpha	30	
Hrgbeli	Number		
Hrgjual	Number		
Stok	Number		

9.4. Menyimpan Tabel

langkah - langkah penyimpanan tabel (missal dari tersebut diatas akan kita simpan pada folder took dengan nama tabel Tabel_barang)

1. Pilih dan Klik **Save As**
2. Perhatikan tampilan jendeela baru untuk penyimpanan tabel



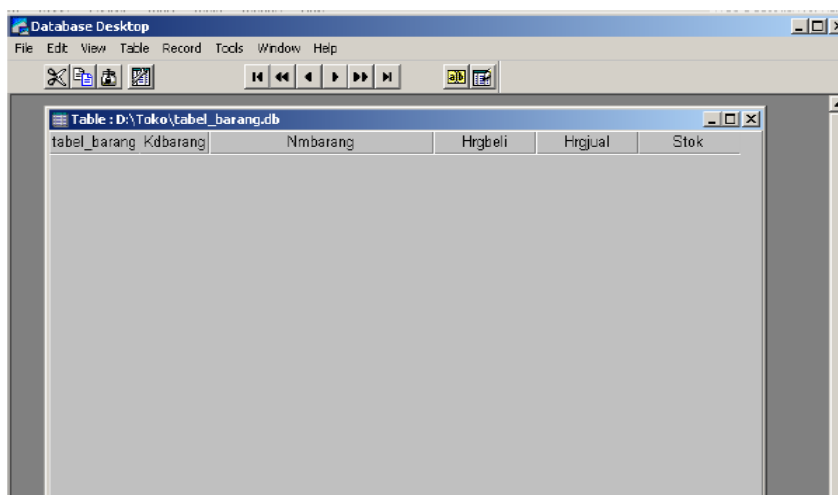
Gambar 9.4. Gambar Save table

3. Perhatikan cara penyimpanan seperti contoh diatas, nama folder harus sesuai dan nama field diketikan pada text box file name.
4. Setelah selesai klik Save.

9.5. Membuka tabel

Langkah - langkahnya adalah

1. Dari Data Base Dekstop, pilih dan klik File I Open



2. Cari tempat penyimpanan tabel dan setelah ditemukan nama tabel pilih dan klik Open
3. Perhatikan tampilan jendela baru untuk Data Base Dekstop

Dari tampilan tersebut diatas ada beberapa hal yang bisa kita lakukan

a. Kembali ke Struktur Tabel

Langkahnya = Dari jendela tersebut klik icon *restructure*

b. Mengisi Data

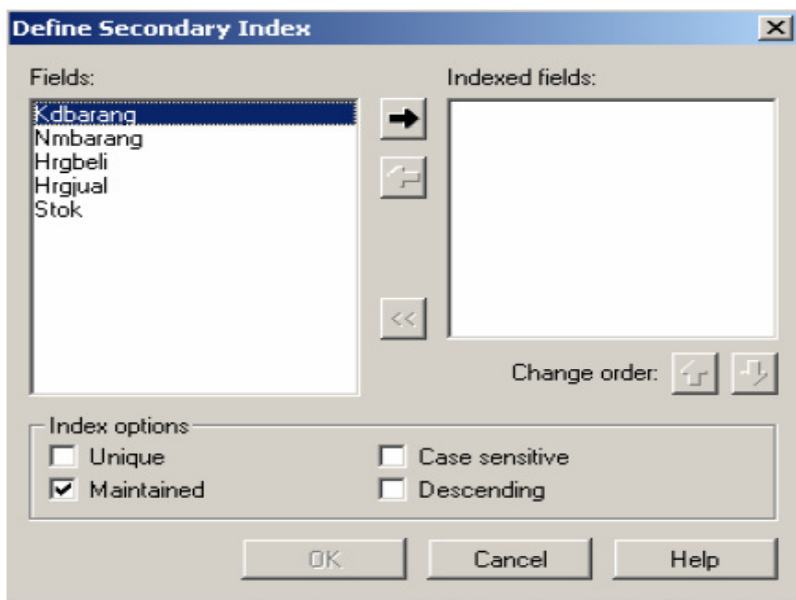
Langkahnya = Dari Jendela tersebut diatas klik icon *Edit Data JMI*

9.6. Membuat Index (secondary Index)

Index digunakan sebagai kunci untuk pencarian data ataupun digunakan untuk mengurutan data pada tabel. Didalam satu tabel diperkenankan menggunakan lebih dari satu index secondary.

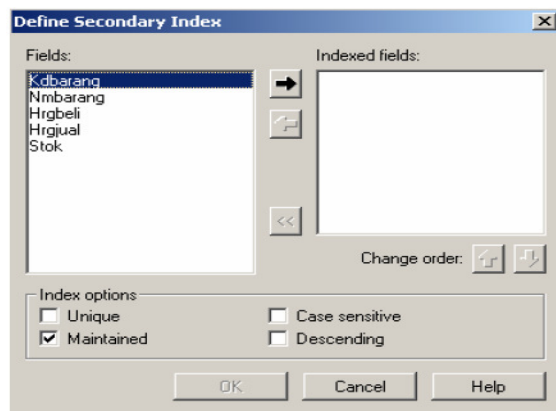
Langkahnya

- > Aktifkan tabel barang
- > Dari DBD pilih table I restructure atau icon Restructure
- > Klik combo table properties dan pilih Secondary Index
- > Klik tombol Define

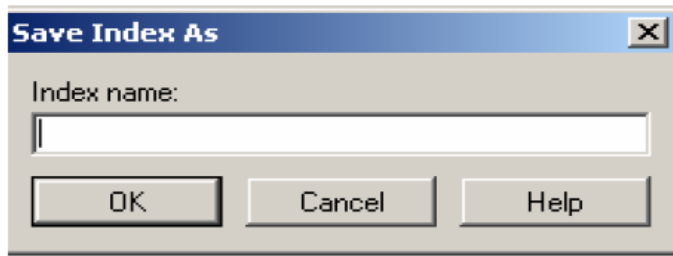


Gambar 9.5 Gambar Create Secondary Index

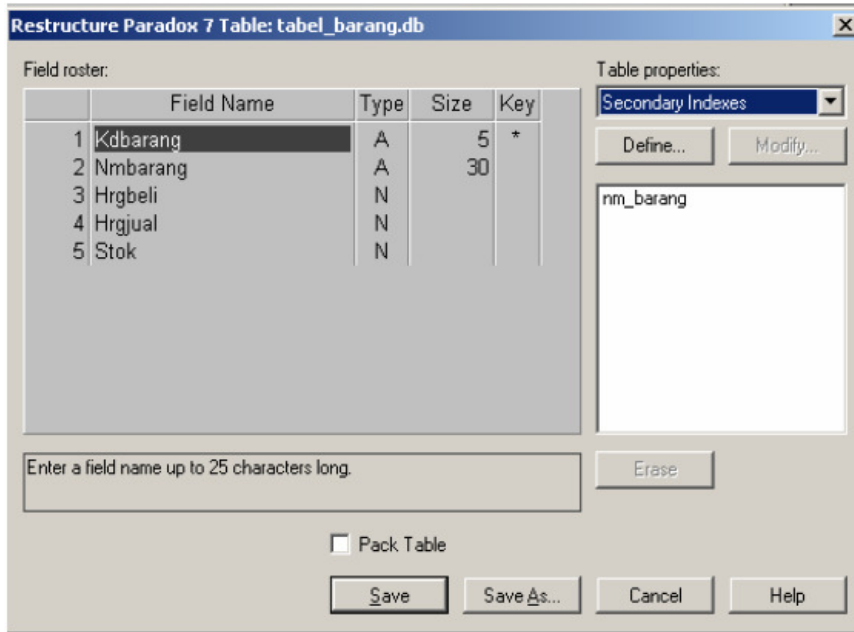
- > Pilih dan klik field yang akan dijadikan secondary index (**Misal nama barang**)
- > Klik tanda panah kekanan sehingga akan ditampilkan seperti gambar berikut



- > Klik OK untuk mengakhiri pembuatan Secondary index
- > Berikan nama index (**missal Nmbarang**) ketika ditampilkan jendela sebagai berikut



> Klik Ok untuk menutup form Save index dan kembali ke menu table barang, seperti berikut:



Catatan : Secondary Index dalam satu tabel boleh lebih dari satu dan yang perlu diingat dalam pemberian nama index tidak boleh sama dengan nama field.

9.7. Mengisi Data Pada Tabel

Langkah - langkah Pengisian data pada tabel

- > Aktifkan database Dekstop I buka tabel barang
- > Pada tampilan data pilih dan klik edit Data
- > Isi Data Sebagai Berikut:

Kdbarang	Nama Barang	Harga Beli	Harga Jual	Kdpemasok	Stok
A0000	Mie Goreng Raya Sapi	2.000,00	2.200,00	01001	2
A0001	Mie Goreng Rasa Baso	1.500,00	1.600,00	01001	25
A0002	Indomie Rebus rasa Soto	2.100,00	2.300,00	03001	34
A0003	Mie Rasa Kari Ayam	1.600,00	1.700,00	02001	36
A0004	Biore	4.500,00	5.000,00	03001	56
A0006	Roti tawar	6.000,00	6.500,00	02002	56

9.8. Membuat Alias Manager

Fungsinya digunakan untuk menggantikan fungsi path ketika tabel yang dibuat akan dipanggil pada jendela program.

Contoh =Tanpa Alias

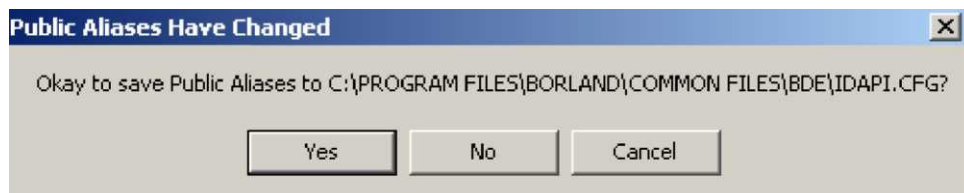
Ketika kita akan mengaktifkan tabel yang disimpan di C didalam Folder Toko dengan nama tabel Barang, maka kita kita harus menuliskan program sebagai berikut : **C:\Toko\barang.db**. (itu bisa semakin panjang jika folder penyimpanan data semakin bertingkat)

Dengan Alias

Pemanggilan program dengan **alias** kita tinggal panggil nama Alias baru kita tuliskan nama tabel. Dengan alias tidak mengenal penggunaan folder yang bertumpuk, karena **Alias** ditempatkan pada system **Sistem Operasi**

Langkah - langkahnya

1. Dari jendela Data base desktop pilih tools
2. pilih dan klik **alias manager**, perhatikan tampilan jendela Alias manager
3. Pada Jendela Alias Manger pilih dan klik New
4. Ketikan nama alias pada database alias, misal toko
5. Pada Database Type pilih Standard
6. Pada jendela path ketikan path tempat penyimpanan field tabel atau klik browse untuk pencarian secara otomatis.
7. Klik OK
8. Jika Yes jika muncul tampilan sebagai berikut



9. Jika tidak tampil Public Aliases Have Changed berarti pada kesalahan dalam menuliskan path pada jendela Alias Manager.

Catatan = dalam pembuatan nama alias cukup dibuat satu kali untuk satu project program, walaupun ada penambahan tabel setelah pembuatan Alias karena fungsi Alias sama seperti fungsi dari database.

9.9. Latihan

Tambahkan tabel baru dan simpan pada folder yang sama dengan nama = **Pemasok Struktur Tabel**

Field	Type Data	Size	Key
Kdpemasok	Alpha	5	*
Nmpemasok	Alpha	30	
Almpemasok	Alpha	50	
Ktpemasok	Alpha	20	
Tlppemasok	Alpha	10	

Bab X

Kontrol Data Set Dan Navigator



Kontrol Dataset merupakan control yang diberikan kepada tabel dalam suatu data base. Hal ini meliputi bagaimana data terhubung dengan tabel dari suatu data base. Setelah terhubung bagaimana proses menggerakkan record, entah itu keawal, ke akhir, ke record sesudah maupun ke record sebelumnya.

10.1. Komponen Table

a. Borland Database Engine(BDE)

Komponen Borland Database Engine merupakan media penghubung antara database dengan aplikasi program. Pada awalnya BDE digunakan untuk paradox. Komponen BDE mendukung akses database yang merupakan bawaan dari Delphi walaupun tidak menutup kemungkinan bisa juga digunakan untuk pengaksesan database yang bersifat client/server. Salah satu kelebihan BDE adalah integrasi yang sudah sangat baik dengan Delphi. Sisi lain penggunaan BDE hanya maksimal untuk pemrograman yang bersifat standalone.


Bentuk Komponen BDE pada Delphi

Komponen Delphi	Keterangan
	Komponen Data Source = Komponen ini digunakan untuk mengakses Data base (terdapat dalam tab Data Access)
	Komponen table = Komponen ini digunakan untuk mengakses tabel yang terdapat dalam Database (terdapat dalam tab BDE)

b. ActiveX Data Object (ADO)

ADO merupakan salah satu teknologi Akses Data Base tingkat tinggi. Kelebihan antar muka ADO adalah merupakan teknologi Akses Independen terhadap setiap aplikasi program. ADO mendukung aplikasi yang bersifat local maupun yang berbasis Client Server. Komponen ADO terdapat dalam tab ADO pada Component Palette.


Bentuk Komponen ADO

Komponen Delphi	Keterangan
	Komponen ADO Connection = Komponen ini digunakan untuk mengakses Data base (terdapat dalam Tab ADO)

c. dbexpress


merupakan teknologi pengaksesan database dengan kemampuan yang bersifat terbatas pada teknologi akses untuk database yang bersifat client/Server. Kemampuan terbaik untuk dbexpress adalah didalam pembuatan laporan.

Bentuk Komponen DBExpress

Komponen Delphi	Keterangan
	Komponen SQL Connection = Komponen ini digunakan untuk mengakses Data base (terdapat dalam tab Dbexpress)

d. Interbase Express (IBX)

Merupakan teknologi control database yang bersifat open source. IBX merupakan teknologi akses database server yang bersifat khusus.

Komponen Delphi	Keterangan
	Komponen IBDatabase = Komponen ini digunakan untuk mengakses Data base (terdapat dalam tab Interbase)

10.2. Kontrol Tabel

Kontrol tabel adalah komponen yang menyediakan akses ke dalam sekumpulan record terdapat didalam suatu tabel. Kontrol ini merupakan konsep bagaimana Akses ke dalam suatu tabel menjadi lebih mudah. Salah satu Kontrol Tabel yang bisa digunakan adalah dengan memanfaatkan fasilitas yang ada di page BDE.

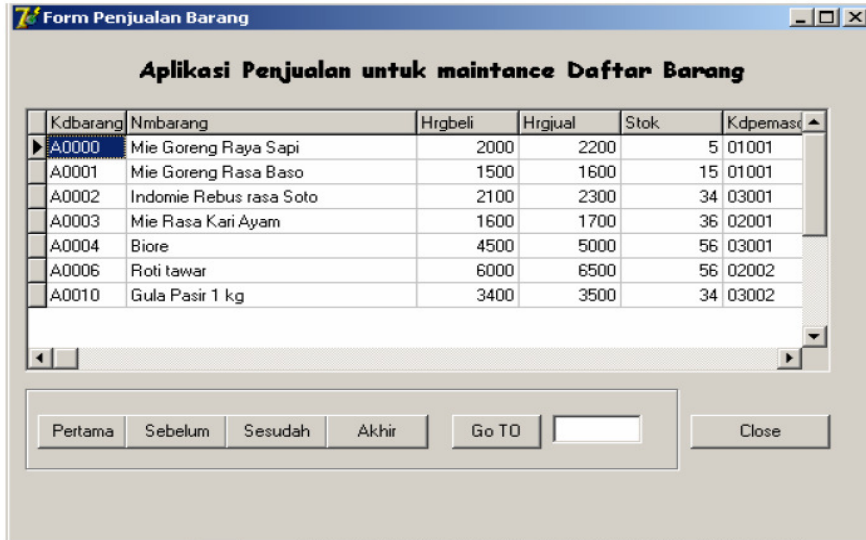
- a. Ttable = Digunakan untuk menghubungkan ke tabel dalam suatu Data.
- b. Tquery = Digunakan untuk menempatkan hasil dari sebuah Query.

Kontrol Tabel memiliki dua Properti

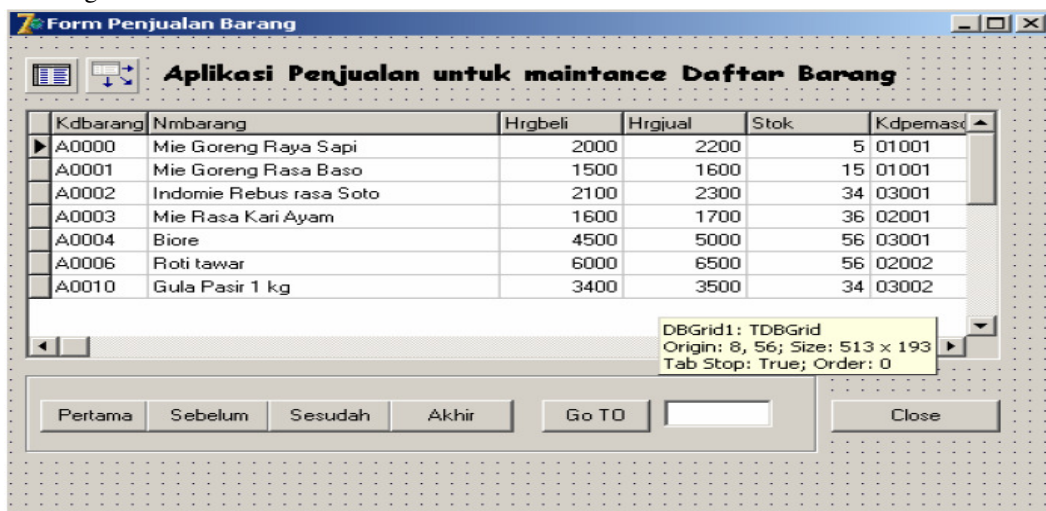
- a. Properti BOF (*Begin Of File*)
Menunjukkan bahwa control data di awal record
- b. Properti EOF(*end Of File*)
Menunjukkan bahwa control data di akhir record

10.3. Merancang Program Dengan Kontrol Tabel

- a. Hasil yang diinginkan



- b. Design Form



c. Komponen Tambahan yang dibutuhkan

1. 6 button dan 1 edit text
2. 1 Dbgrid (dalam komponen **Data Control**) Digunakan untuk menampilkan data tabel pada form.
3. 1 table (dalam **komponen BDE**)
Digunakan untuk menghugnkan antara tabel dengan database yang dibutuhkan
4. 1 datasource (dalam **Komponen Data Access**)
Digunakan untuk koneksi Database / Alias Data Base Didalam suatu folder.

d. Setting Properties

Object	Properties	
	Name	Caption/text
Label 1	Label 1	AplikasiPenj
Button 1	Bawal	Pertama
Button2	Bsebelum	Sebelum
Button3	BSesudah	Sesudah
Button4	Bakhir	Akhir
Button5	bLoncat	Go To
Button6	Bclose	Close
Edit1	Eloncat	-
Object	Data Base Name	Table Name
Table 1	Toko	Barang.db
Object	Data Set	
Data Source 1	Table 1	
Object	Data Source	
Dbgrid 1	Data Source 1	
Object	Active	
Table 1	True	

Ketikan program berikut ini

```
procedure TForm1.bpertamaClick(Sender: TObject); begin
table 1. Fir st end;
```

```
procedure TForm1.bsebelumClick(Sender: TObject); begin
if tabel.Bof then
  showmessage('Anda diawal record') else
  TAB LEI.Prior; end;
```

```
procedure TForm1.bsesudahClick(Sender: TObject); begin
if tabel.Eof then
  showmessage('Anda berada pada record terakhir') else
  table 1.Last; end;
```

```
procedure TForm1.bakhirClick(Sender: TObject); begin table 1.Next;
end;
```

```
procedure TForm1.bloncatClick(Sender: TObject); begin
table1.MoveBy(strtoint(eloncat.Text)); end;
```

```
procedure TForm1.bcloseClick(Sender: TObject); begin
if (application.MessageBox('Anda yakin form akan ditutup','Info',MB_YESNO)= ID YES) then
  close;
end;
```

BAB XI MANIPULASI DATA/TABEL MASTER

11.1. Program Tambah Data

Konsep program ini meliputi bagaimana proses pencarian data yang digunakan untuk memberikan batasan agar data yang disimpan bukan merupakan data yang sama, proses penyimpanan dan membatalkan penambahan data.

11.2. Event Yang Digunakan

- findkey* : digunakan untuk pencarian data, apakah data sudah ada atau belum.
- Append*: digunakan untuk membuka record kosong pada tabel
- Post*: event pada tabel ini digunakan untuk menyimpan data pada tabel.
- Databasename* : digunakan untuk memanggil nama data base.
- Table* : digunakan untuk mengaktifkan tabel.
- Dataset*: digunakan untuk menghubungkan antara datasource dengan tabel.
- Datasource* : digunakan untuk menghubungkan grid dengan tabel melalui datasource.

11.3. Aplikasi Tambah Data Barang

a. Hasil Form Setelah Dijalankan

Kdbarang	Nmbarang	Hrgbeli	Hrgju
A0000	Mie Goreng Raya Sapi	2000	
A0001	Mie Goreng Rasa Baso	1500	
A0002	Indomie Rebus rasa Soto	2100	
A0003	Mie Rasa Kari Ayam	1600	
A0004	Biore	4500	
A0006	Roti tawar	6000	

b. Desain Form

Kdbarang	Nmbarang	Hrgbeli	Hrgju

c. Desain Properties

Object	Properties	
	Name	Caption/text
Edit1	Ekdbarang	-
Edit2	Enmbarang	-
Edit3	Ehrgbeli	-
Edit4	Ehrgjual	-
Edit5	Estok	-
Label1	Label1	Kode Barang
Label2	Label2	Nama Barang
Label3	Label3	Harga Beli
Label4	Label4	Harga Jual
Label5	Label5	Stok
Button1	Btambah	&Tambah
Button2	Bsimpan	&Simpan
Button3	Bbatal	&Batal
Button4	Bexit	&Exit
Object	Data Base Name	Table Name
Table1	-	-
Object	Data Set	
Data Source1	-	
Object	Data Source	
Dbgrid1	-	
Object	Active	
Query	-	

d. Listing Program

> Pembuatan Prosedur

Ada 4 prosedur yang dibutuhkan : tampil, hidup, kosong, mati.

Langkah pembuatan

Double klik pada form, ketika berada pada jendela unit dari form tempatkan kursor berada declaration, khususnya pada bagian deklarasi public. Setelah itu ketikan deklarasi prosedur seperti berikut:

```
public procedure tampil;
  procedure kosong;
  procedure mati;
  procedure hidup;
  { Public declarations }
end;
```

Penulisan Program Prosedur

Penulisan program prosedur secara umum sama seperti penulisan program pada umumnya, hanya disini penulisan nama prosedurnya ditulis oleh programmer. Penulisannya terdapat pada bagian *implementation*.

Program untuk prosedur tampil

Digunakan untuk menampilkan data pada form

```
procedure tform6.tampil; begin
ekdbarang.Text := table1f kdbarang'; enmbarang.Text := table1f nrnbarang';
ehrgbeli.Text := floattostr(table1['hrgbeli']); ehrgjual.Text :=
floattostr(table1['hrgjual']); estok.Text := floattostr(table1['stok']); end;
```

Program Untuk Prosedur Mati

Digunakan untuk menonaktifkan semua text box yang ada pada form.

```
procedure TForm6.mati;
```

```
begin
```

```
end;
```

- > Program tampil diaktifkan untuk menampilkan data pada edit text box, program digunakan semasa form diaktifkan.

```
procedure TForm6.FormActivate(Sender: TObject);
```

```
begin
```

```
tampil;
```

```
mati;
```

```
bsimpan.Enabled := false; end;
```

- > Program tambah

```
procedure TForm6.btambahClick(Sender: TObject);
```

```
begin
```

```
mati;
```

```
ekdbarang.Enabled := true; bsimpan.Enabled :=
```

```
true; form6.ActiveControl := ekdbarang; kosong;
```

```
end;
```

- > Program Pencarian Data. Control data yang digunakan adalah enter ketika kode barang diinput pada edit text kodebarang.

```
procedure TForm6.ekdbarangKeyPress(Sender: TObject; var Key:
```

```
Char);
```

```
begin
```

```
if key = #13 then begin
```

```
if tabel.FindKey([ekdbarang.Text]) then begin
```

```
showmessage('Data Sudah ada');
```

```
exit; end
```

```
else
```

```
hidup;
```

```
ekdbarang.Enabled := false;
```

```
form6.ActiveControl := enmbarang;
```

```
end;
```

```
end;
```

- > Program Penyimpanan Data ke Dalam tabel

```
procedure TForm6.bsimpanClick(Sender: TObject); begin
```

```
tabel.Append;
```

```
tablel[kdbarang] := ekdbarang.Text; tablel[nmbarang]
```

```
:= enmbarang.Text; tablel[hrgbeli] :=
```

```
strtofloat(ehrgbeli.Text); tablel[hrgjual] :=
```

```
strtofloat(ehrgjual.Text); tablel['stok'] := strtof
```

```
loat(estok.Text); tabel.Post; mati; end;
```

- > Program Membatalkan penyimpanan data

```
procedure TForm6.bbatalClick(Sender: TObject);
```

```
begin
```

```
kosong;
```

```
tabel.First;
```

```
tampil;
```

```
mati;
```

```

end;
> Program Mengaktifkan data sesuai deengan pilihan pada setiap record pada
procedure TForm6.DBGrid1CellClick(Column: TColumn);
begin
tampil;
end;
> Program untuk menutup form
procedure TForm6.bexitClick(Sender: TObject); begin
tabel.Close;
close;
end;

ekdbarang.Enabled := false;
enmbarang.Enabled := false;
ehrgbeli.Enabled := false;
ehrgjual.Enabled := false;
estok.Enabled := false; end;

```

Program Untuk Prosedur Hidup

Fungsi ini digunakan untuk mengembalikan fungsi edit text, seperti fungsi normalnya.

```

procedure
tform6.hidup; begin
ekdbarang.Enabled := true;
enmbarang.Enabled := true;
ehrgbeli.Enabled := true;
ehrgjual.Enabled := true;
estok.Enabled := true; end;

```

Program Prosedur Kosong

```

procedure tform6.kosong;
begin
ekdbarang.Text := "";
enmbarang.Text := "";
ehrgbeli.Text := "";
ehrgjual.Text := "";
estok.Text := ""; end;

```

> Langkah awal menghubungkan semua perangkat tabel untuk mengaktifkan tabel. Perintah ini terdapat *padaformcreate*.

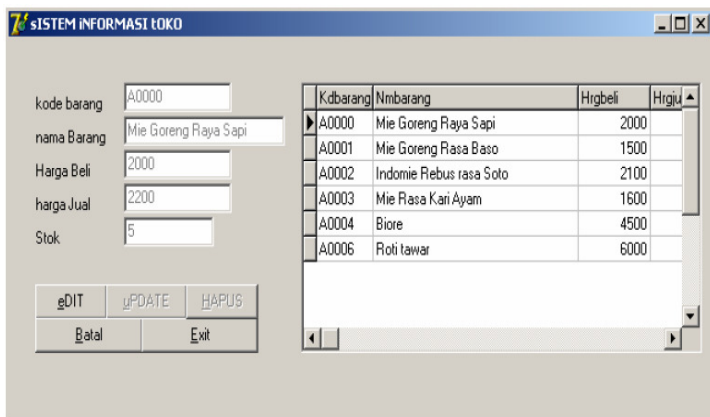
```

procedure TForm6.FormCreate(Sender: TObject); begin
tabel.DatabaseName := 'penjualan' ;
tabel.TableName := 'barang.db';
tabel.Active:= true;
datasourcel.DataSet:= tabel;
dbgrid1.DataSource := datasourcel;

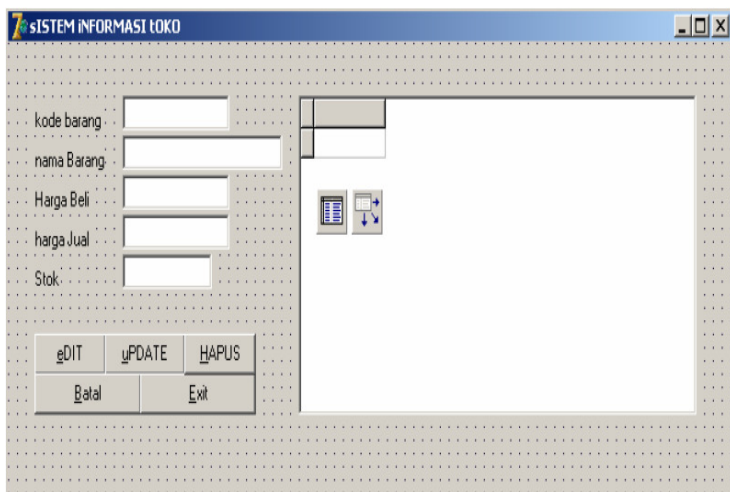
```

11.4. Aplikasi Program Edit Dan Hapus Data

a. Hasil Setelah Form Dijalankan



b Desain Form



c. Desain properties

Label4	Label4	Harga Jual
Label5	Label5	Stok
Button 1	Bedit	&Edit
Button2	B update	&Update
Button3	Bhapus	&Hapus
Button4	Bbatal	&Batal
Button5	Bexit	&Exit
Object	Data Base Name	Table Name
Table1	-	-
Object	Data Set	
Data Source 1	-	
Object	Data Source	
Dbgrid 1	-	
Object	Active	
Query	-	

Object	Properties	
	Name	Caption/text
Edit1	Ekdbarang	-
Edit2	Enmbarang	-
Edit3	Ehrgbeli	-
Edit4	Ehrgjual	-
Edit5	Estok	-
Label 1	Label 1	Kode Barang
Label2	Label2	Nama Barang
Label3	Label3	Harga Beli

d. Listing program

> Program Untuk Prosedur

```
procedure tform7.tampil;  
  
begin  
ekdbarang.Text := tablelf kdbarang']; enmbarang.Text := tablelf nrnbarang'];  
ehrgbeli.Text := floattostr(tablel['hrgbeli']); ehrgjual.Text :=  
floattostr(tablel['hrgjual']); estok.Text := floattostr(tablel['stok']);  
end;
```

```
procedure tform7.kosong;  
  
begin  
ekdbarang.Text := "; enmbarang.Text := ";  
ehrgbeli.Text := "; ehrgjual.Text := ";  
estok.Text := ";  
end;
```

```
procedure tform7.mati;  
  
begin  
ekdbarang.Enabled := false;  
  enmbarang.Enabled := false;  
ehrgbeli.Enabled := false;  
ehrgjual.Enabled := false;  
estok.Enabled := false; end;
```

```
procedure tform7.hidup; begin  
ekdbarang.Enabled := true;  
enmbarang.Enabled := true;  
ehrgbeli.Enabled := true;  
ehrgjual.Enabled := true;  
estok.Enabled := true; end;
```

> Program Untuk Mengaktifkan Database dan tabel pada form

```
procedure TForm7.FormCreate(Sender: TObject); begin  
tablel.DatabaseName := 'penjualan' ; tablel.TableName := 'barang.db';  
tablel.Active:= true; datasourcel.DataSet:= tablel; dbgrid1.DataSource :=  
datasourcel; end;
```

> Program Selama Form Aktif

```
procedure TForm7.FormActivate(Sender: TObject);  
begin  
tampil;  
mati;  
bupdate.Enabled := false; bhapus.Enabled := false; end;
```

> Program untuk button Edit

```
procedure TForm7.beditClick(Sender: TObject);  
begin  
end;
```

> Program hapus Data

```
procedure TForm7.bhapusClick(Sender: TObject); begin  
table1.Delete; mati;  
table1.First; end;
```

> Program Batal

```
procedure TForm6.bbatalClick(Sender: TObject);  
begin  
kosong;  
table1.First;  
tampil;  
mati;  
end;
```

> Program Untuk Dbgrid

```
procedure TForm6.DBGrid1CellClick(Column: TColumn);  
begin  
tampil;  
end;
```

> Program Menutup Form

```
procedure TForm6.bexitClick(Sender: TObject); begin  
table1.Close; close;  
end;
```


Bab XI1

Pencarian Data

Salah satu konsep pemrograman berbasis data base proses pencarian data menjadi satu hal yang sangat penting didalam mendukung kesempurnaan hasil didalam sebuah aplikasi program. Fungsi secara umum adanya pencarian data adalah untuk mendapatkan secara cepat data yang diinginkan dari sebuah tabel didalam sebuah database.

12.1 Konsep Pemrograman Pencarian Data

b Konsep Pencarian dengan Properti

Konsep ini merupakan metode pencarian data dimana memanfaatkan kemampuan property untuk mendapatkan data yang diinginkan

Ada 5 konsep pencarian data dengan property

1. Locate

Locate adalah metode pencarian data untuk record yang sama dengan criteria yang sudah ada ataupun mendekati dengan kriteria yang ada. Pencarian locate dapat digunakan untuk tabel dengan index maupun yang tanpa index

Contoh

```
If not table1.locate ('nama', enama.text, []) then  
    Messagedlg("" + enama.text + " Tdak ditemukan", mterror, [mbok],0);
```

2. Findkey

Metode finkey digunakan mencari record yang sama . Metode ini dapat digunakan untuk tabel dengan index.

Contoh

```
Table1.indexname := '1';  
If not table1.findkey ([ '00099']) then  
    Messagedlg('data Tdak ditemukan', mterror, [mbok],0);
```

Proses pencarian data diatas digunakan untuk table dengan index sebagai primary key. Sedangkan untuk pencarian dengan indes secondary dapat terlihat seperti contoh berikut ini.

```
Table1.indexname := 'nama';  
If not table1.findkey([enama.text]) then  
    Messagedlg("" + enama.text + " " Tdak ditemukan', mterror, [mbok],0); Else  
    Tampil;
```

Contoh

```
Var strsql : string; Begin  
Strsql := 'Select * from barang where nama like  + enama.text +'%';  
Query1.sql.clear;  
Query 1. sql I .add (st r sq I);  
Query.close;  
Query.open;  
End;
```

12.2. Aplikasi Program Pencarian Data

a. Dengan Properti

1. Hasil Form Setelah Dijalankan

The screenshot shows a Windows application window titled "Form11". It contains a table with the following data:

Kdbarang	Nmbarang	Hrgbeli	Hrgjual	Stok	Kdpemasok
A0000	Mie Goreng Raya Sapi	2000	2200	5	01001
A0001	Mie Goreng Rasa Baso	1500	1600	25	01001
A0002	Indomie Rebus rasa Soto	2100	2300	34	03001
A0003	Mie Rasa Kari Ayam	1600	1700	36	02001
A0004	Biore	4500	5000	56	03001
A0006	Roti tawar	6000	6500	56	02002

Below the table, there are search controls:

- A search box labeled "Cari Data Barang" with a "Cari" button.
- A "Kodee Barang" input field containing "A0003".
- A "Nama Barang" input field containing "Mie Rasa Kari Ayam".
- A "Harga Beli" input field containing "1600".
- A "Kode Pemasok" input field containing "02001".
- "Batal" and "Close" buttons.

2. Desain Form

The screenshot shows the design view of the "Form11" application. It features a large empty rectangular area for data display. Below this area, there are search controls:

- A search box labeled "Cari Data Barang" with a "Cari" button.
- A "Kodee Barang" input field.
- A "Nama Barang" input field.
- A "Harga Beli" input field.
- A "Kode Pemasok" input field.
- "Batal" and "Close" buttons.
- Navigation icons (back, forward, search) are visible next to the "Kode Pemasok" field.

3. Desain Properties

Catatan = Koneksi table dan data base akan dilakukan secara programming, jadi untuk setting properties tidak perlu disetup.

Object	Properties	
	Name	Caption/text
Group Box1	Group Box1	Cari Data Barang
Edit1	Ecari	-
Edit2	Enmbarang	-
Edit3	Ehrghbeli	-
Edit4	Kdpemasok	-
Label 1	Label 1	Kode Barang
Label2	Label2	Nama Barang
Label3	Label3	Harga Beli
Label4	Label4	Kode Pemasok
Button 1	Ecari	Cari
Button2	Ebatal	Batal
Button3	Eclose	Close
Object	Data Base Name	Table Name
Table1	-	-
Object	Data Set	
Data Source 1	-	
Object	Data Source	
Dbgrid 1	-	
Object	Active	
Table1	-	

4. Listing Program

Ketika pada saat program dijalankan maka proses koneksi program akan diaktifkan selama form tersebut diaktifkan.

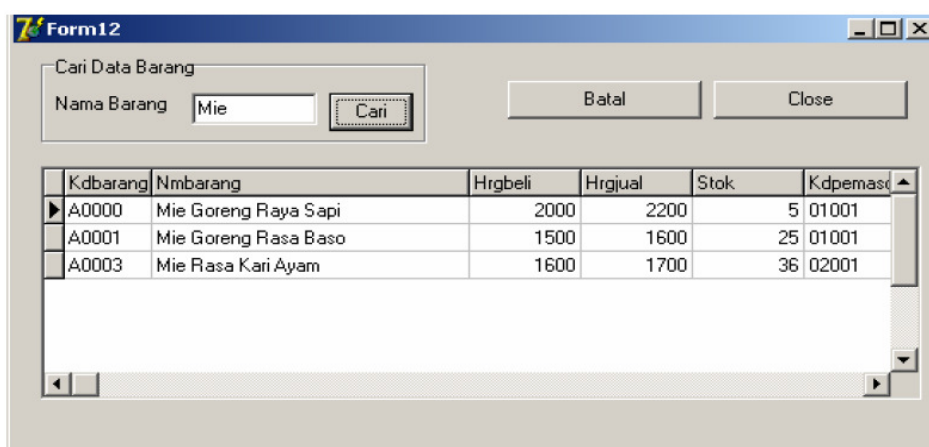
```

procedure TForm11.FormCreate(Sender: TObject); begin
tabel.DatabaseName := 'penjualan' ;
tabel.TableName := 'barang.db'; tabel.Active:=
true; datasourcel.DataSet:= tabel;
dbgrid1.DataSource := datasourcel; end;

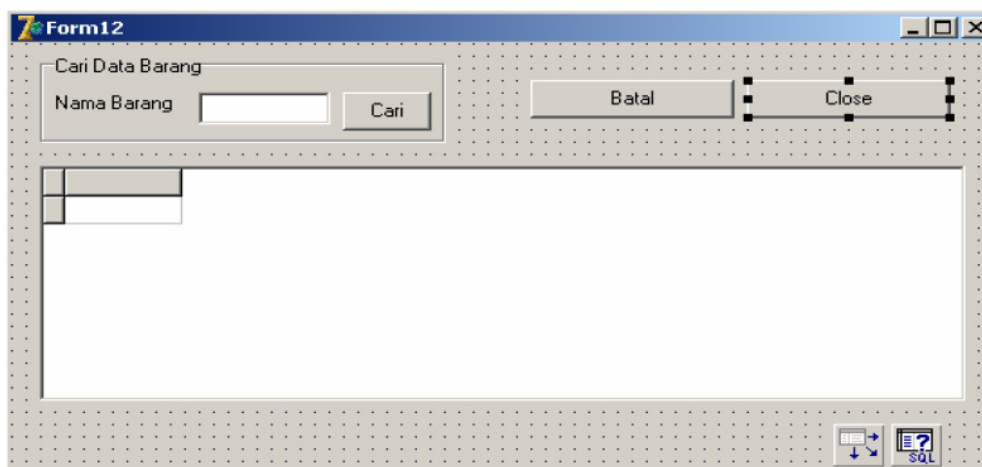
```

b. Dengan Metode SQL

1. Hasil Setelah Form Dijalankan




2. Design Form



3. Desain Properties

Catatan = untuk koneksi dengan perintah SQL maka icon table tidak dibutuhkan tetapi icon query yang dibutuhkan. Icon Query terdapat dalam tab BDE.

Object	Properties	
	Name	Caption/text
Group Box 1	Group Box 1	Cari Data Barang
Edit1	Ecari	-
Label1	Label1	Nama Barang
Button1	Ecari	Cari
Button2	Ebatal	Batal
Button3	Eclose	Close
Object	Data Base Name	Table Name
Query1	-	-
Object	Data Set	
Data Source 1	-	
Object	Data Source	
Dbgrid1	-	
Object	Active	
Query	-	

4. Listing Program

Program koneksi dengan perintah SQL

```

procedure TForm12.FormCreate(Sender: TObject);
//var strsql : string;
begin
query1.DatabaseName := 'penjualan'; query1.SQL.Add
('Select * from barang'); query1.Active := true;
datasource1.DataSet := query1; dbgrid1.DataSource :=
datasource1; end;

```

Penjelasan

Secara umum koneksi database dan tabel hampir sama dengan koneksi dengan icon table, hanya terdapat perbedaan ketika harus mengaktifkan tabel. Dengan query harus dituliskan sintaks perintah SQL. Kondisi ini lebih luwes karena bisa menampilkan data lebih dari satu tabel.

Program pencarian nama

```

procedure TForm12.bcariClick(Sender: TObject);
var strsql : string;
begin
strsql := 'Select * from barang where nmbarang like "' + ecari.Text +
/o ,
query1. SQL.Clear;
query1.SQL.Add(strsql);
query1.Close;
query1.Open;
end;

```

Program Untuk mengembalikan data ke semua record

```

procedure TForm12.bbatalClick(Sender: TObject);
var sql : string;
begin
sql := 'Select * from barang' ;
query1. SQL.Clear;

```

```

queryl.SQL.Add(sql);
queryl.Close;
queryl.Open;
end;

```

Program menutup Form

```

procedure TForm12.bcloseClick(Sender: TObject);
begin
close;
end;

```

Program pencarian dimulai dengan input kode barang pada txtcari dan diakhiri dengan button cari. Jika data barang tidak ditemukan maka akan ditampilkan pesan, tetapi jika ditemukan maka data barang akan ditampilkan.

```

procedure TForm11.bcariClick(Sender: TObject); begin
table1.IndexName := '';
if not table1.FindKey([ecari.Text]) then
begin
showmessage('Data Belum ada'); exit;
form11.ActiveControl := ecari; end else
enmbarang.Text := table1f'nmbarang']; ehrgbeli.Text :=
table1f'hrgbeli']; ekdpemasok.Text :=
table1f'kdpemasok']; end;

```

Penjelasan

Table1.indexname := '' menunjukkan bahwa proses pencarian dengan memanfaatkan kunci utama (*primary key*), sehingga koneksi program dianjurkan dengan menggunakan metode *findkey*. *If Not table1.findkey([ecari.text])* menunjukkan bahwa jika data tidak ada, maka akan ditampilkan pesan bahwa tidak ditemukan dan kursor dikembalikan ke txtcari, tetapi jika ada maka data akan ditampilkan.

Program untuk membatalkan proses pencarian dengan mengaktifkan txtcari kembali.

```

procedure TForm11.bbatalClick(Sender: TObject); begin
ecari.Text := ''; form11.ActiveControl :=
ecari; end;

```

Program untuk menutup form

```

procedure TForm11.bbatalClick(Sender: TObject); begin
ecari.Text := ''; form11.ActiveControl :=
ecari; end;

```

BAB XIII

FILTER DAN RANGE DATA

Didalam Suatu konsep pemograman data base memberikan sebuah kemudahan didalam penanganan suatu tabel terhadap setiap user adalah sesuatu hal yang mutlak untuk diadakan. Salah satu fasilitas yang dapat memberikan kemudahan didalam pengelolaan data base adalah adanya fungsi range maupun filter. Fungsi ini hanya maksimal digunakan untuk field dengan type data numeric.

13.1. Filter

merupakan fungsi yang digunakan untuk membatasi tampilan data pada setiap tabel sesuai dengan data yang diinginkan. Method yang sering digunakan didalam penulisan fungsi filter adalah :

- a. *Tablefilterrecord* : fungsi ini digunakan untuk kontrol data terhadap field mana yang digunakan sebagai kunci.
- b. *Filtered* : fungsi ini digunakan untuk mengaktifkan fungsi dari event filter.
- c. *Fieldbyname* : berfungsi untuk menentukan field yang digunakan sebagai kunci pengurutan.
- d. *Indexname* : digunakan untuk mengaktifkan nama index dari suatu tabel.

13.2. Range

merupakan fungsi yang dapat digunakan untuk menampilkan data berdasarkan cakupan data atau kelompok data. Dalam hal ini fungsi range membutuhkan nilai yang dapat digunakan sebagai batasan awal kelompok maupun nilai untuk menentukan batasan akhir terhadap kelompok tersebut.

- a. *serangestart*: digunakan untuk menentukan nilai awal terhadap suatu range.
- b. *Setrangeend*: digunakan untuk menentukan nilai akhir terhadap suatu range.
- c. *Fieldbyname*: digunakan sebagai kunci field mana yang digunakan sebagai kunci.
- d. *Applyrange* : digunakan untuk mengeksekusi terhadap batasan range yang diinginkan.
- e. *Canceclrange* : digunakan untuk membatalkan perintah range yang sudah diberikan dan mengembalikan data ke fungsi normal.
- f. *Indexname* : digunakan untuk memanggil index primary key.

d. Listing Program

Untuk mengaktifkan fungsi filter record klik *table* dan pada **event** pilih dan doubleklik *onfilterrecord*. Ketikkan kode program berikut:

```
procedure TForm3.TableFilterRecord(DataSet: TDataSet;  
  var Accept: Boolean); begin  
  accept := table1.FieldByName('stok').AsFloat = strtolofloat(efilter.Text); end;
```

Program untuk mengeksekusi hasil filter record, ketikkan program pada button Filter.

```
procedure TForm3.bfilterClick(Sender: TObject); begin  
  table1.IndexName := 'indstok';  
  table1.Filtered:=true;
```

13.3. Merancang Aplikasi Program Dengan Fungsi Range dan Filter

a. Hasil setelah form dijalankan

Menampilkan Data (berdasarkan Stok)

Batas Bawah: 5
Batas Akhir: 36
Tampilkan

Filter Data (berdasarkan Stok)

Stok Barang:
Tampilkan

Normal

Kdbarang	Nmbarang	Hrgbeli	Hrgjual	Stok	Kdpemasok
A0000	Mie Goreng Raya Sapi	2000	2200	5	01001
A0001	Mie Goreng Rasa Baso	1500	1600	25	01001
A0002	Indomie Rebus rasa Soto	2100	2300	34	03001
A0003	Mie Rasa Kari Ayam	1600	1700	36	02001

b. Desain Form

Menampilkan Data (berdasarkan Stok)

Batas Bawah:
Batas Akhir:
Tampilkan

Filter Data (berdasarkan Stok)

Stok Barang:
Tampilkan

Normal

Kdbarang	Nmbarang	Hrgbeli	Hrgjual	Stok	Kdpemasok
A0000	Mie Goreng Raya Sapi	2000	2200	5	01001
A0001	Mie Goreng Rasa Baso	1500	1600	25	01001
A0002	Indomie Rebus rasa Soto	2100	2300	34	03001
A0003	Mie Rasa Kari Ayam	1600	1700	36	02001
A0004	Biore	4500	5000	56	03001
A0006	Roti tawar	6000	6500	56	02002

bfilter: TButton
Origin: 8, 48; Size: 153 x 25
Tab Stop: True; Order: 1

c.Desain Properties

Object	Properties	
	Name	Caption/text
Group Box 1	Group Box 1	Menampilkan Data (berdasarkan kode)
Group Box 2	Group box 2	Filter Data (berdasarkan kode)
Edit1	Eawal	-
Edit2	Eakhir	-
Edit3	Efilter	-
Label1	Label1	Batas Bawah
Label2	Label2	Batas Akhir
Label3	Label3	Stok Barang
Button1	Brange	Tampilkan
Button2	bfilter	Tampilkan
Button3	Bnormal	Normal
Object	Data Base Name	Table Name
Table1	Penjualan	Barang
Object	Data Set	
Data Source1	Table1	
Object	Data Source	
Dbgrid1	Data Source1	
Object	Active	
Query	True	

D.Listing Program

Untuk mengaktifkan fungsi filter record klik *table* dan pada *event* pilih dan *doubleklik onfilterrecord*. Ketikkan kode program berikut:

Program mengeksekusi range tabel, ketikkan program berikut pada button range

```

procedure TForm3.brangleClick(Sender: TObject); begin
table1.IndexName := 'indstok';
table1.SetRange([strtof loat(eawal.Text)],[strtof loat(eakhir.Text)]);
table1.ApplyRange;
end;

```

Program untuk mengembalikan data ke posisi default

```

procedure TForm3.bnormalClick(Sender: TObject); begin
table1.IndexName := "";
table1.CancelRange;
table1.Filtered := false;
end;

```


BAB XIV

Koneksi Database dengan menggunakan Ms.Access

14.1 Langkah - Langkah pembuatan Database Pada Access

Buat Database : Penggajian

Table : T_Pegawai

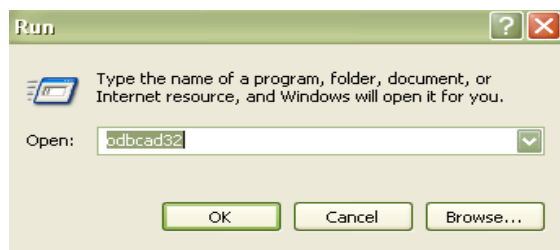
Field Name	Data Type	Size
Nip	Text	9
Nm_pegawai	Text	30
Bagian	Text	20
Gol	Text	5
Gaji_pokok	Number	Longint
Tmpt_lahir	Text	25
Tgl_lahir	Date /Time	
J_kelamin	Text	15
Agama	Text	15
Sts_nikah	Text	10
Jml_anak	Number	Byte
Alamat	Text	100
Tlp	Text	15

Table : T_Pegawai

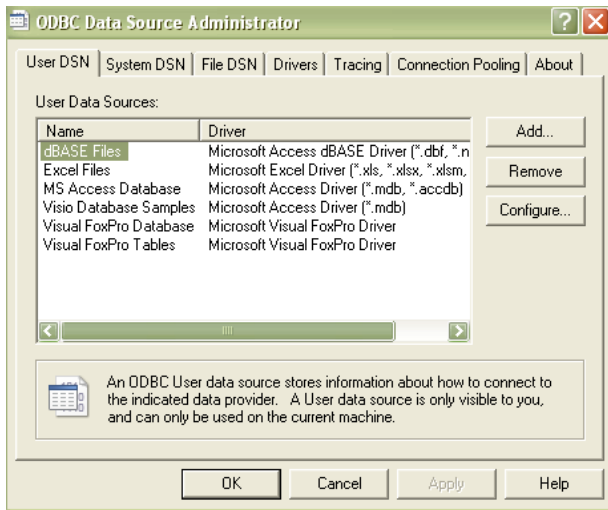
Field Name	Data Type	Size
No_Slip	Text	9
Tanggal	Text	30
Jam	Text	20
Pendapatan	Text	5
Potongan	Number	Longint
Gaji_bersih	Text	25
Nip	Date /Time	
Kode_User	Text	15

14.2 Langkah-langkah Koneksi database

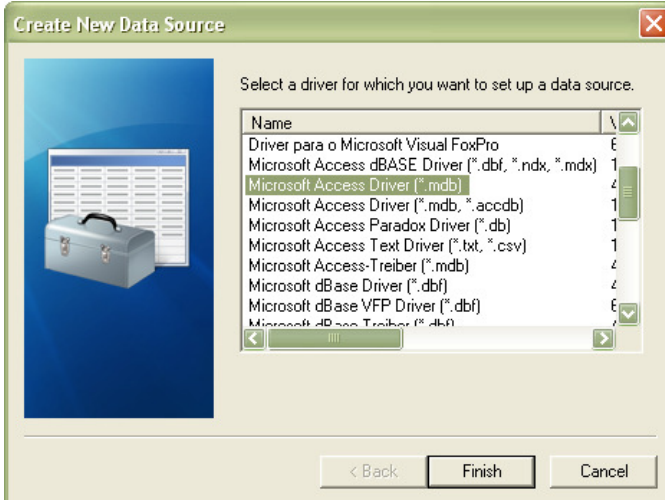
Klik Start → Run



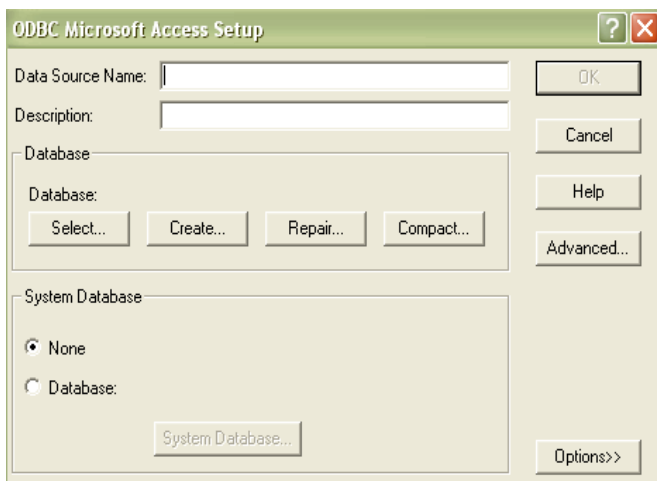
Ketikkan *odbcad32* → **OK**



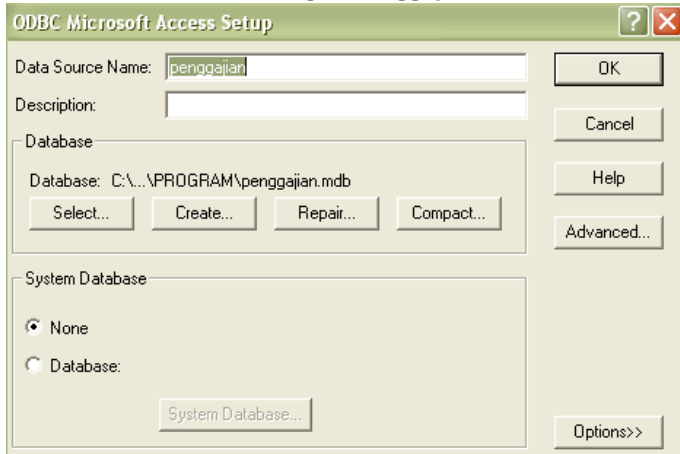
Pilih *MS Access Database* → **Add**



Pilih *Microsoft Access Driver* → **Finish**



Isi Data Source Name dengan Penggajian



Pilih tombol Select... (sesuai tempat penyimpanan alamat database yang sudah dibuat)

Pilih tombol OK → **OK**

BAB XV

Komponen Query Atau SQL (Strutured Query Language)

15.1. Sekilas Mengenai SQL

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

15.2. Dasar - Dasar mengenai Metode SQL

15.2.1. 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_field1
    tippedatal, Nm_field2 tippedata2,
    nm_f ieldN tippedataN
);
```

Contoh

```
Create table barang (Kdbarang varchar(5) not null primary key,
    Nmbarang varchar(15)
);
```

15.2.2. DML (*Data Manipulation Language*) a. Metode

Select

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

Ketentuan Lain =

1. Buat Secondary Indeks untuk field nmpemasok dengan nama indek = nama dan ktpemasok dengan nama indeks = kota
2. Isi Data pemasok dengan data sebagai berikut

Kdpemasok	nmpemasok	Almpemasok	Ktpemasok	Tlppemasok
01001	PT Wahana Kencana	Jalan B angka no 14	Jakarta	021-736666
01002	Pt Kencana Abadi	Jalan Bekasi Raya No 45	Bandung	021-450000
02001	PT Makmur Sejahtera	Jalan Raya Cibubur no 56	Bandung	021-888888
02002	PT Abadi Raya	Jalan Bandung No 35	Jakarta	021-488222
03001	PT Cinta Abadi	Jalan Lenteng Agung NO 62	Jakarta	021-888883
03002	PT Mutiara Makmur	Jalan Jaksa NO 53	Jakarta	021-000211

1. 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.

2. 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.

d. Fungsi Agregasi

Fungsi Agregasi adalah fungsi matematika yang digunakan bersamaan dengan perintah **Select**. Berbagai macam agregasi yang digunakan bersamaan dengan perintah **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 (namajield) From namajabel Contoh

Select Sum(stok) From barang e.

Query Untuk banyak Tabel

Adalah fungsi Query yang digunakan untuk menampilkan lebih dari satu tabel **Bentuk Umum**

Select *index1.daftarjieldjabell, index1.daftarJieldJabel2* **From** *tabell index1, tabel2 index1* **Where** *index2.tabell = index1.tabell*

Atau

Select Distinct *tabell.daftarjield, tabell.Daftar Jield* **From** *tabell, tabell* **Where** *tab ell.namajield = tabell.nama Jield*

Contoh

Select Distinct *barang.nmbarang, barang.hargajel, 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.

Select * From *barang* **where** *Harga_beli* **Between** *5000 And 10000*

Adalah semua data pada tabel barang dengan harga beli di antara 5000 dan 10000 **Select * From**

barang **where** *nmbarang like 'mei%'*

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

Select * From *barang* **where** *nmbarang Like 7omei'*

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

b. 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

c. Mengelompokkan Data (Group By)

Fungsi ini digunakan untuk mengelompokkan data berdasarkan field terpilih. **Bentuk Umum**

Select * From *barang* **Group By** *kd barang* **Contoh**

Select * From *barang* **group by** *kd barang*

Menampilkan seluruh data barang dengan dikelompokkan berdasarkan kode barang.

f. 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

1. Insert

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

Insert Into *namajabel* (*field1, field2, field3,..... fieldn*)
Values (*nilai1, nilai2, nilai3,..... nilain*)

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

Contoh

Insert Into *barang*(*kdbarang, nmbarang, hargabeli, hargajual, stok*) **Values**
(*'A001', 'Indomei rasa Soto',20000,30000,23*)

2. Update

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

Update *namajabel*
Set *field1 = nilai1, field2=nilai2, field3 = nilai3,fieldn = nilain*)
Where *syarat*

Contoh

Update *barang*
Set *kdbarang = 'A001', nmbarang = 'Tdomie 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.

3. Delete

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

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

Atau

Delete From *namajabel* (digunakan untuk menghapus 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

15.2.3. Penerapan Konsep SQL pada Aplikasi

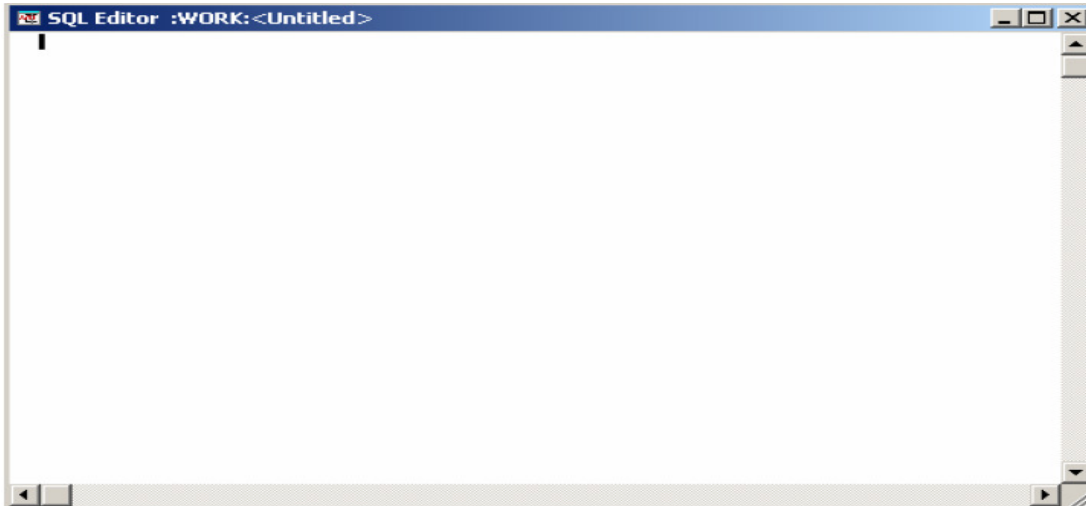
Penerapan konsep pemrograman SQL pada Delphi dapat kita implementasikan dengan dua konsep yaitu dengan konsep Data Base Dekstop dan Konsep pemrograman

15.2.3.1 Konsep Database Dekstop

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

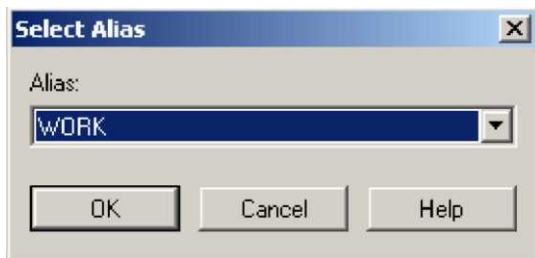
Langkah - langkahnya

- a. Aktifkan Data Base Dekstop
- b. Dari Menu **File I New I SQL File**. Jendela SQL Editor akan ditampilkan



Gambar 15.1 Jendela SQL Editor

- a. Dari Menu Sql pilih Select Alias (untuk table yang ada dalam folder penyimpanan), Sehingga pada Layar akan ditampilkan jendela select alias.



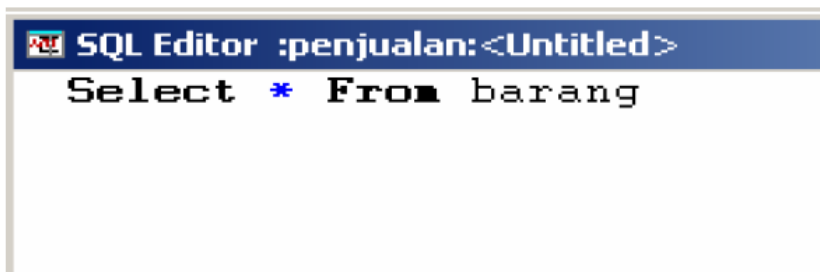
Gambar 9.2. Jendela Select Alias

- e. Dari jendela Select Alias pilih dan aktifkan nama alias yang sudah dibuat sebelumnya (**TOKO**). Setelah selesai klik **OK**.
- f. Pada jendela editor tulisakn perintah SQL dan jalankan dengan memilih icon **Run SQL**.

Untuk mencoba kita gunakan beberapa kasus berikut ini

- a. Menampilkan semua data barang **Sintaks**

Penulisan



ANSWER	Kdbarang	Nmbarang	Hrgbeli	Hrgjual	Stok	Kdpemasok
1	A0000	Mie Goreng Raya Sapi	2.000,00	2.200,00	5,00	01001
2	A0001	Mie Goreng Rasa Baso	1.500,00	1.600,00	25,00	01001
3	A0002	Indomie Rebus rasa Soto	2.100,00	2.300,00	34,00	03001
4	A0003	Mie Rasa Kari Ayam	1.600,00	1.700,00	36,00	02001
5	A0004	Biore	4.500,00	5.000,00	56,00	03001
6	A0006	Roti tawar	6.000,00	6.500,00	56,00	02002

b. Menampilkan semua Data Barang dengan kode Pemasok=01001 Sintaks Penulisan

```
SQL Editor :penjualan:<Untitled>
Select * From barang where kdpemasok = '01001'
```

Hasil

ANSWER	Kdbarang	Nmbarang	Hrgbeli	Hrgjual	Stok	Kdpemasok
1	A0000	Mie Goreng Raya Sapi	2.000,00	2.200,00	5,00	01001
2	A0001	Mie Goreng Rasa Baso	1.500,00	1.600,00	25,00	01001

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

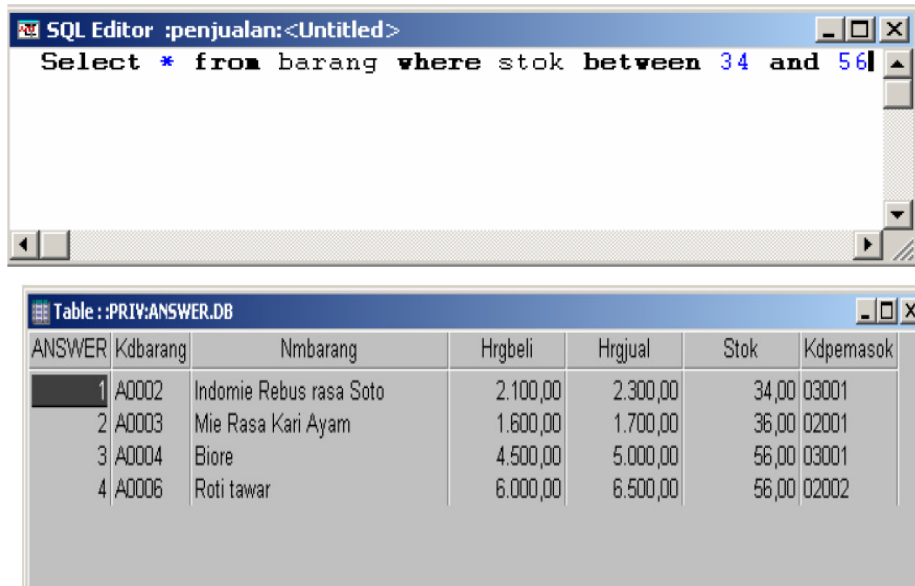
Sintaks Penulisan

```
SQL Editor :penjualan:<Untitled>
Select kdbarang as KodeBarang, nmbarang From barang
order by nmbarang desc
```

Hasil

ANSWER	KodeBarang	nmbarang
1	A0006	Roti tawar
2	A0003	Mie Rasa Kari Ayam
3	A0000	Mie Goreng Raya Sapi
4	A0001	Mie Goreng Rasa Baso
5	A0002	Indomie Rebus rasa Soto
6	A0004	Biore

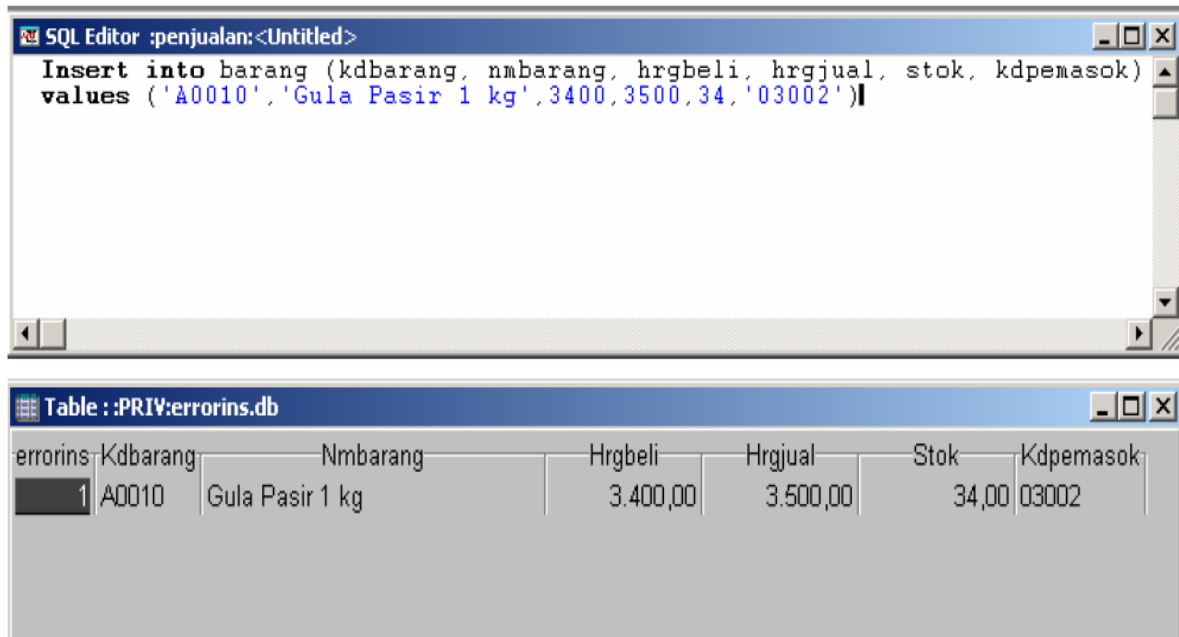
- c. Menampilkan Semua Data barang untuk stok diantara 34 sampai dengan 56
Sintak Penulisan



- e. Menambahkan data barang dengan Data sebagai berikut:

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

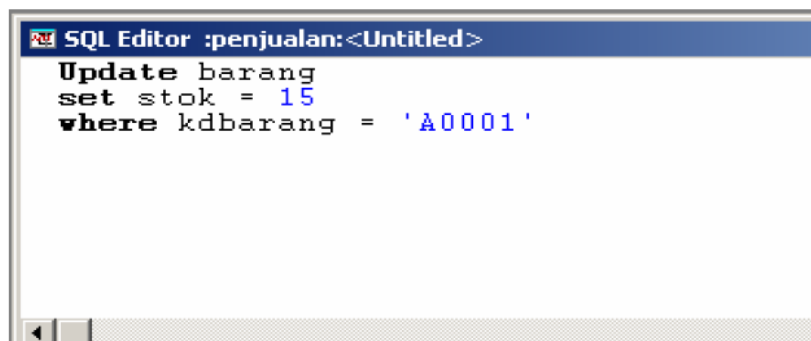
Sintaks Penulisan



Untuk melihat semua data gunakan perintah menampilkan data seluruhnya

- f. Mengganti Stok barang untuk kode A001 menjadi 15

Sintaks Penulisan



Hasil

updated	kdbarang	Nmbarang	Hrgbeli	Hrgjual	stok	Kdpemasok
1	A0001	Mie Goreng Rasa Baso	1.500,00	1.600,00	15,00	01001

G. Tugas

1. Hapus Data Barang untuk field stok dibawah 20
2. Tampilkan Nama Pemasok, Kode Pemasok dari Tabel Pemasok dan nama barang,harga beli dari table barang. Sesuai dengan pemasok masing-masing seperti yang ada dalam table barang khusus nama barang yang berwalan dengan kata mie. **Dengan hasil sebagai berikut ini**

ANSWER	kdpemasok	nmpemasok	nmbarang	hrghbeli
1	01001	PT Wahana Kencana	Mie Goreng Rasa Baso	1.500,00
2	01001	PT Wahana Kencana	Mie Goreng Raya Sapi	2.000,00
3	02001	PT Makmur Sejahtera	Mie Rasa Kari Ayam	1.600,00
4	02002	PT Abadi Raya	Roti tawar	6.000,00
5	03001	PT Cinta Abadi	Biore	4.500,00
6	03001	PT Cinta Abadi	Indomie Rebus rasa Soto	2.100,00
7	03002	PT Mutiara Makmur	Gula Pasir 1 kg	3.400,00

3. Tampilkan Nama Pemasok, Kode Pemasok dari tabel pemasok dan nama barang, harga beli dari tabel barang sesuai dengan pemasok masing - masing seperti yang ada dalam tabel barang khusus nama barang yang berawaln dengan kata **Mie**. Dengan hasil seperti berikut ini

ANSWER	kdpemasok	nmpemasok	nmbarang	hrghbeli
1	01001	PT Wahana Kencana	Mie Goreng Rasa Baso	1.500,00
2	01001	PT Wahana Kencana	Mie Goreng Raya Sapi	2.000,00
3	02001	PT Makmur Sejahtera	Mie Rasa Kari Ayam	1.600,00

4. Cari rata-rata dari stok barang

BAB XVI

APLIKASI FORM FILE TRANSAKSI (PENJUALAN)

Simpanlah Form dibawah ini didalam Database Alias yang sudah dibuat yaitu :**Toko_nimmasing2**

Buatlah Table dibawah ini:

1. Table Barang

Field	Tipe Data	Size	Key
Idbrg	A	5	*
Nmbrg	A	30	
Ukuran	A	15	
Harga	N		

2. Table Customer

Field	Tipe Data	Size	Key
No_cust	Alpha	5	*
Nama	Alpha	30	
No_hp	Alpha	15	

3. Table Trans

Field	Tipe Data	Size	Key
Nofak	A	3	*
No_cust	A	5	
Total	N		
Ubay	N		
Ukem	N		

4. Table DetailTrans

Field	Tipe Data	Size	Key
Nofak	Alpha	5	
Idbrg	Alpha	3	
Qty	N		
Subtotal	N		

5. Table Sementara

Field	Tipe Data	Size	Key
Idbrg	A	5	*
Nmbrg	A	30	
Ukuran	A	15	
Harga	N		
Qty	N		
Subtotal	N		

Form Transaksi



Simpanlah Form dibawah ini didalam Dabase Alias yang sudah dibuat yaitu : **Toko_nimmasing2**
Dan simpanlah Form dengan nama : **form_trans**

The screenshot shows a Windows-style application window titled "Form Penjualan <<". The main content area has a dotted background and is titled "FORM TRANSAKSI PENJUALAN PT. POHAN JAYA tbk".

Fields include:

- NO FAKTUR (text box)
- NO CUSTOMER (text box)
- NAMA CUSTOMER (text box)
- NO TELP (text box)
- KODE BARANG (dropdown menu with "++Pilih Kode Barang++")
- NAMA BARANG (text box)
- UKURAN (text box)
- HARGA (text box)
- JUMLAH BELI (text box)
- SUBTOTAL (text box)
- TOTAL (text box)
- UANG BAYAR (text box)
- UANG KEMBALI (text box)

Below the fields is a data grid with columns: Idbrg, Nnbrg, Ukuran, Harga, Qty, Subtotal. To the right of the grid are buttons: "Input Baru", "ADD", "SAVE", and "EXIT".

Object	Keterangan	Name
Edit Text	No Faktur	Nofak
Edit Text	No Customer	Nocust
Edit Text	Nama Customer	Nmcust
Edit Text	No Telp	Notelp
Combo Box	Kode Barang	Kdbrg
Edit text	Nama Barang	Nnbrg
Edit text	Ukuran	Ukuran
Edit text	Harga	Harga
Edit text	Jumlah Beli	jumlah
Edit Text	Total	Total
Edit Text	Uang Bayar	Ubay
Edit text	Uang Kembali	Ukem

Object	Keterangan	Name
Table 1	Table Barang	Tblbrg
Table 2	Table Customer	Tblcust
Table 3	Table Trans	Tbltrans
Table 4	Table Detail Trans	Tbldetail
Table 5	Table Sementara	Tblsem
Data Source1	Dataset :Tblbrg	
Data Source2	Dataset :Tblcust	
Data Source3	Dataset :Tbltrans	
Data Source4	Dataset :Tbldetail	
Data Source5	Dataset :Tblsem	
Edit Text	Subtotal	txtsubtot
DB Grid	Db Grid	Dbgrid1

Logika Program

1. Buatlah Procedure **Aktif, Pasif, Bersih dan No Otomatis** (Untuk membuat No Otomatis dengan format cth: **001,002,003**).
2. Ketika Form pertama kali dijalankan seluruh objek dalam keadaan aktif.
3. Add di klik maka no otomatis pada faktur langsung terisi.
4. Pada saat no customer di enter maka terjadi **Validasi pencarian berdasarkan Table customer**.
5. Isi **Kode barang diambil dari Data barang**, ketika dipilih maka nama barang, ukuran dan harga terisi.
6. Save berfungsi sebagai Menyimpan serta mengupdate Data.
7. Pada saat Jumlah beli di enter maka total keluar.
8. Pada saat Uang bayar < dari total maka akan ada pesan jika tidak maka Uang kembali terisi.
9. Exit untuk tutup form.

Listing :

```
unit form_penjualan;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, DB, DBTables, StdCtrls, Grids, DBGrids;
```

```
type
```

```
TForm4 = class(TForm)  
Label4: TLabel;  
Label5: TLabel;  
Label11: TLabel;  
Label2: TLabel;  
Label3: TLabel;  
Label6: TLabel;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;
```

```

Label13: TLabel;
Label14: TLabel;
nmcust: TEdit;
notelp: TEdit;
harga: TEdit;
ukuran: TEdit;
nmbrg: TEdit;
total: TEdit;
ubay: TEdit;
ukem: TEdit;
jumlah: TEdit;
nofak: TEdit;
nocust: TEdit;
kdbrg: TComboBox;
tblcust: TTable;
tblbrg: TTable;
tbltrans: TTable;
tbldetail: TTable;
DBGrid1: TDBGrid;
tblsem: TTable;
DataSource1: TDataSource;
DataSource2: TDataSource;
DataSource3: TDataSource;
DataSource4: TDataSource;
DataSource5: TDataSource;
badd: TButton;
bsave: TButton;
bexit: TButton;
Button1: TButton;
txtsubtot: TEdit;
Label15: TLabel;
procedure baddClick(Sender:
TObject);
    procedure nocustKeyPress(Sender:
TObject; var Key: Char);

```

```

    procedure kdbrgClick(Sender:
TObject);
    procedure Button1Click(Sender:
TObject);
    procedure FormCreate(Sender:
TObject);
    procedure FormActivate(Sender:
TObject);
    procedure bexitClick(Sender:
TObject);
    procedure bsaveClick(Sender:
TObject);
    procedure ubayKeyPress(Sender:
TObject; var Key: Char);
    procedure jumlahKeyPress(Sender:
TObject; var Key: Char);
private
    { Private declarations }

```

```

public
procedure bersih;
procedure aktif;
procedure pasif;
procedure no_otomatis;
procedure hitung;
procedure gridbersih;
    { Public declarations }
end;

```

```

var
mtotal:real;
    Form4: TForm4;

```

implementation

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

```
procedure tform4.gridbersih ;
```

```
begin
```

```
if tblsem.RecordCount<>0 then
```

```
begin
```

```
tblsem.First;
```

```
while not tblsem.eof do
```

```
begin
```

```
tblsem.Delete ;
```

```
tblsem.Next;
```

```
tblsem.Refresh
```

```
end;
```

```
end;
```

```
end;
```

```
procedure tform4.bersih;
```

```
begin
```

```
nofak.Text:="";
```

```
nocust.Text:="";
```

```
nmcust.Text:="";
```

```
notelp.Text:="";
```

```
kdbrg.Text:="";
```

```
nmbrg.Text:="";
```

```
ukuran.Text:="";
```

```
harga.Text:="";
```

```
jumlah.Text:="";
```

```
end;
```

```
procedure tform4.aktif;
```

```
begin
```

```
nofak.Enabled:=True ;
```

```
nocust.Enabled:=True ;
```

```
nmcust.Enabled:=True ;
```

```
notelp.Enabled:=True ;
```

```
kdbrg.Enabled:=True ;
```

```
nmbrg.Enabled:=True ;
```

```
ukuran.Enabled:=True ;
```

```
harga.Enabled:=True ;
```

```
jumlah.Enabled:=True ;
```

```
total.Enabled:=True ;
```

```
ubay.Enabled:=True ;
```

```
ukem.Enabled:=True ;
```

```
end;
```

```
procedure tform4.pasif;
```

```
begin
```

```
nofak.Enabled:=false ;
```

```
nocust.Enabled:=false;
```

```
nmcust.Enabled:=false ;
```

```
notelp.Enabled:=false;
```

```
kdbrg.Enabled:=false ;
```

```
nmbrg.Enabled:=false;
```

```
ukuran.Enabled:=false ;
```

```
harga.Enabled:=false;
```

```
jumlah.Enabled:=false;
```

```
total.Enabled:=false;
```

```
ubay.Enabled:=false;
```

```
ukem.Enabled:=false;
```

```
end;
```

```
procedure tform4.no_otomatis;
```

```
var
```

```
c:string;
```

```
a:integer;
```

```
begin
```

```
tbltrans.Last;
```

```
if tbltrans.RecordCount=0 then
```

```
begin
```

```
nofak.Text:='001';
```

```
exit;
```

```
end
```

```
else
```

```
begin
```

```
c:=tbltrans['nofak'];
```

```
a:=strtoint(c)+1;
```

```
if a<10 then
```

```
nofak.Text:='00'+ inttostr(a)
```

```
else if a<100 then
```

```
nofak.Text:='0' + inttostr(a)
```

```
else
```

```
nofak.Text:=inttostr(a)
```

```
end;
```

```
end;
```



```

procedure tform4.hitung;
begin
  mtotal:=0;
  if tblsem.RecordCount<>0 then
  begin
    tblsem.First;
    while not tblsem.eof do
    begin
      mtotal:=mtotal+tblsem['subtotal'];
      tblsem.Next;
    end;
  end;
  end;
  else
  begin
    mtotal:=0;
  end;
  total.Text:=floattostr(mtotal);

end;

procedure
TForm4.baddClick(Sender: TObject);
begin
  aktif;
  bersih;
  no_otomatis;
end;

procedure
TForm4.nocustKeyPress(Sender:

```

```

TObject; var Key: Char);
begin
  if key=#13 then

  if tblcust.FindKey([nocust.Text]) then
  begin
    showmessage('Data Customer ada!');
    nmcust.Text:=tblcust['nama'];
    notelp.Text:=tblcust['no_hp'];
  end
  else
  begin
    showmessage('Data Customer Tidak
    ada!');
    exit;
  end;
  end;

procedure
TForm4.kdbrgClick(Sender:
TObject);
begin
  if tblbrg.FindKey([kdbrg.Text]) then
  begin
    nmbrg.Text:=tblbrg['nmbrg'];
    ukuran.Text:=tblbrg['ukuran'];
    harga.Text:=tblbrg['harga'];
    jumlah.SetFocus;
  end;
  end;

```

```

procedure
TForm4.Button1Click(Sender:
TObject);
begin
if tblsem.FindKey([kdbrg.Text]) then
begin
tblsem.Edit;
tblsem['qty']
:=tblsem['qty']+strtoint(jumlah.Text);
tblsem['subtotal']:=tblsem['subtotal']+strtoint(txtsubtot.Text);
tblsem.Post;
end
else
begin
tblsem.Append;
tblsem['idbrg']:=kdbrg.Text;
tblsem['nmbrg']:=nmbrg.Text;
tblsem['ukuran']:=ukuran.Text;
tblsem['harga']:=strtoint(harga.Text);
tblsem['qty']:=strtoint(jumlah.Text);
tblsem['subtotal']:=strtoint(txtsubtot.Text);
tblsem.Post;

end;
hitung;
kdbrg.Text:="";
nmbrg.Text:="";
ukuran.Text:="";
harga.Text:="";
jumlah.Text:="";
txtsubtot.Text:="";

```

end;

```

procedure
TForm4.FormCreate(Sender:
TObject);
begin
tblbrg.First;

```

```

while not tblbrg.eof do
begin
kdbrg.Items.add(tblbrg['idbrg']);

```

```

tblbrg.Next;
end;
end;

```

```

procedure
TForm4.FormActivate(Sender:
TObject);
begin
gridbersih;
end;

```

```

procedure TForm4.bexitClick(Sender:
TObject);
begin
close;
end;

```

```

procedure
TForm4.bsaveClick(Sender:
TObject);
begin
tblsem.First;
while not tblsem.eof do
begin
tbldetail.Append;
tbldetail['nofak']:=nofak.Text;
tbldetail['idbrg']:=tblsem['idbrg'];
tbldetail['qty']:=tblsem['qty'];
tbldetail['subtotal']:=tblsem['subtotal'];
tbldetail.Post;
tblsem.Next;
end;

```

```

tbltrans.Append;
tbltrans['nofak']:=nofak.Text;
tbltrans['nocust']:=nocust.Text;
tbltrans['total']:=strtofloat(total.Text);
tbltrans['ubay']:=strtofloat(ubay.Text);
tbltrans['ukem']:=strtofloat(ukem.Text);
tbltrans.Post;

```

```

pasif;
gridbersih;
end;

```

```
procedure TForm4.ubayKeyPress(Sender: TObject; var Key: Char);  
begin  
if key=#13 then  
if strtoint(ubay.Text)< strtofloat(total.Text) then  
begin  
showmessage('Uang Bayar tidak valid');  
exit;  
end  
else  
begin  
showmessage('Uang Bayar valid');  
ukem.Text:=floattostr(strtoint(ubay.Text)- strtofloat(total.Text));  
end;  
end;
```

```
procedure TForm4.jumlahKeyPress(Sender: TObject; var Key: Char);  
begin  
if key=#13 then  
txtsubtot.Text:=inttostr(strtoint(jumlah.Text)*strtoint(harga.Text));  
  
end;  
  
end.
```

Latihan !!

- Buat Project dengan nama : Pjtkbuku
- Buat Database Alias dengan nama : toko_buku
- Buat tabel sbb:

1. Tabel buku

Field Name	Type	Size	Key
Kdbuku	A	5	*
Judul	A	30	
Harga	N		
Stok	N		

Isi sebanyak 3 record

2. Tabel Transaksi

Field Name	Type	Size	Key
Nofak	A	5	*
Kdbuku	A	5	
Jmlbeli	N		
subtotal	N		
Totbayar	N		
ubayar	N		
Ukem	N		

3. Tabel Sementara

Field Name	Type	Size	Key
Kdbuku	A	5	
Jmlbeli	N		
subtotal	N		
Harga	N		
Judul	A	30	

Form1 Amanah Book Store

No. Faktur:

Kode Buku: Judul: Harga: Stok: Jumlah Beli:

Kdbuku	Jmlbeli	Subtotal	Harga	Judul
▶ BK002	3	150000	50000	Microsoft Access 2003

Total Bayar:

Uang Bayar:

Uang Kembali:

Ketentuan soal!!!

1. Pada saat form dijalankan semua object tidak aktif kecuali tombol input dan keluar.
2. No faktur muncul secara otomatis.
3. Klik tombol input, maka setfocus di kode buku, tombol simpan dan batal aktif.
4. Masukan kode buku kemudian di enter judul,harga,stok muncul dan setfocus di jumlah beli. Namun jika kode tidak ada muncul pesan "data tidak ditemukan"
5. Jumlah beli diinput dan dienter, maka data masuk ke dbgrid dan total bayar muncul otomatis, setfocus di uang bayar.
6. Masukan uang bayar, lalu enter maka uang kembali muncul.
7. Klik tombol simpan, maka data masuk ke tabel transaksi dan muncul pesan " data sudah disimpan". Tombol input, simpan dan batal tidak aktif.
8. Klik tombol batal maka form menjadi bersih.
9. Klik tombol keluar maka muncul pesan.

BAB XVII QUERY DAN SQL LANJUTAN

Perintah Sql merupakan perintah dengan tingkat akurasi dan fleksibilitas yang tinggi, sehingga mampu diterapkan untuk semua aplikasi program baik itu berupa penggunaan perintah standart maupun perintah dengan tingkat kesulitan yang lebih tinggi. Kemudahan itu adalah salah satu kelebihan dengan menggunakan perintah SQL.

17.1. Aplikasi Program Hapus Dan Retur Stok Barang

a. Hasil Setelah Form Dijalankan

The screenshot shows a window titled "Sistem Informasi Penjualan Toko". It contains the following elements:

- No Transaksi:** Input field with value "00003".
- Kode Barang:** Dropdown menu with value "A0002".
- Rincian Data Barang:** A table with columns: Kode Barang, Nama Barang, Stok Barang, Harga Jual, Jumlah Beli. The row contains: A0002, Indomie Rebus ras, 30, 2300, 1.
- Buttons:** "Retur Data" and "Close".
- Main Table:** A table with columns: notrans, jmbeli, kdbarang, nbarang, hrgjual, subtotal. The row contains: 00003, 1, A0002, Indomie Rebus rasa Soto, 2300, 2300.

b. Desain Form

The screenshot shows a wireframe version of the "Sistem Informasi Penjualan Toko" window. It includes the same layout as the previous screenshot but with dotted borders and empty input fields:

- No Transaksi:** Empty input field.
- Kode Barang:** Dropdown menu with placeholder "[Kode barang]".
- Rincian Data Barang:** Table with empty input fields for each column.
- Buttons:** "Retur Data" and "Close".
- Main Table:** Table with empty input fields for each column.
- Footer:** Four icons labeled "SQL" with question marks.

C.Properties

Object	Properties	
	Name	Caption/text
Edit1	Enotransaksi	-
Edit2	Ekdbarang	-
Edit3	Enmbarang	-
Edit4	Estok	-
Edit5	Ehrjual	-
Edit6	Ejmlbeli	-
Combo box 1	ckdbarang	[Kode Barang]
Label1	Label1	No Transaksi
Label2	Label2	Kode Barang
Label3	Label3	Nama Barang
Label4	Label4	Harga Jual
Label5	Label5	Stok
Label6	Label6	Jumlah beli
Button1	Bretur	&Retur Penjualan
Button2	BClose	&Close
Object	Data Base Name	Table Name
Table1	-	-
Table2	-	-
Table3	-	-
Object	Data Set	
Data Source 1	-	
Object	Data Source	
Dbgrid1	-	
Object	Active	
Table1	-	
Table2	-	
Table3	-	

d. Listing Program > Program Menghubungkan Form dengan Database

```

procedure TForm14.FormCreate(Sender: TObject); begin
query1.DatabaseName := 'penjualan';
query1.SQL.Add ('Select p.notrans, p.jmlbeli, p.kdbarang, b.nmbarang, b.hrgjual,
p.subtotal from transaksi p, barang b where b.kdbarang = p.kdbarang');
query2.DatabaseName := 'penjualan';
query2.SQL.Add('Select * from barang');
query3.DatabaseName := 'penjualan'; query1.Active :=
true; datasourcel.DataSet := query1;
dbgrid1.DataSource := datasourcel;
end;

```

Penjelasan

Query.databasename : 'penjualan'. penggalan program untuk menghubungkan database dengan form. Pada grid kalau diperhatikan ada field - field dari dua tabel yaitu tabel transaksi dengan tabel barang dengan hanya data barang yang ada

ditabel transaksi yang ditampilkan. Hal itu bisa dilihat dari perintah *'Select p. no trans, p.jmlbeli, p.kdbarang, b.nmbarang, b.hrgjual, p. subtotal from transaksip, barang b where b.kdbarang = p.kdbarang'*. **Select p.notrans, p.jmlbeli, p.kdbarang, b.nmbarang, b.hrgjual, p.subtotal from transaksi p, barang b** adalah menampilkan data dari tabel barang dan transaksi. **where b.kdbarang = p.kdbarang'** adalah kondisi dimana hanya data barang yang ada ditabel transaksi yang ditampilkan, tanpa diberikan kondisi seperti ini maka semua data barang akan ditampilkan walaupun ditabel transaksi tidak ada.

- > Program Untuk menampilkan data transaksi dan data barang sesuai dengan pilihan nomor transaksi yang ada di text nomor transaksi. Dan menampilkan kode barang pada combo box sesuai dengan jumlah barang yang dibeli

```

procedure TForm14.enotransaksiKeyPress(Sender: TObject; var Key: Char);
var strsql : string;
begin
if key = #13 then begin
strsql := 'Select p.notrans, p.jmlbeli, p.kdbarang, b.nmbarang, b.hrgjual,
p.subtotal from transaksi p, barang b where notrans like "' +
enotransaksi.Text + '%" and p.kdbarang = b.kdbarang';
query1.SQL.Clear;
query1.SQL.Add(strsql);
query1.Close;
query1.Open;
while not query1.Eof do
begin
ckdbarang.Items.Add (query1[kdbarang' ]);
query1.Next;
end;
end;
end;

```

- > Program untuk menghapus data transaksi dan mengupdate data stok yang ada ditabel barang.

```

procedure TForm14.hapusitemClick(Sender: TObject);
var sql : string;
strsql : string;
strsqll : string;
a : integer;
b : integer;
c : integer ;
begin
a := strtoint(ejmlbeli.Text);
b := strtoint(estok.Text); c := a + b;

```



```

sql := 'update barang set stok = "%d" where kdbarang = ' +
ckdbarang.Text + ' ' ';
query2.SQL.Clear;
query2.SQL.Add(format(sql,[c]));
query2.ExecSQL;
strsql := 'Delete from transaksi where kdbarang = ' + ckdbarang.Text
+ ' ' and notrans = ' + enotransaksi.Text + ' ' ';
query3.SQL.Clear;
query3.SQL.Add(strsql);
query3.ExecSQL;
strsql1 := 'Select p.notrans, p.jmlbeli, p.kdbarang, b.nmbarang, b.hrgjual,
p.subtotal from transaksi p, barang b where p.kdbarang = b.kdbarang';
query1.SQL.Clear;
query1.SQL.Add(strsql1);
query1.Close;
query1.Open;
form14.ActiveControl := enotransaksi; end;

```

Penjelasan

```

sql := 'update barang set stok = "%d" where kdbarang = ' + ckdbarang.Text + ' ' ';
query2.SQL.Clear;

end;

end;

```

Penjelasan

Secara umum perintah ini adalah sama seperti pada perintah untuk menampilkan data barang sesuai dengan data barang yang ada ditabel transaksi. Perbedaan disini adalah kondisi dimana data yang ditampilkan tidak semuanya tetapi hanya disesuaikan dengan pilihan kondisi sesuai dengan nomor transaksi yang sudah diinput (*where notrans like ' ' + enotransaksi.Text + '% ' and p.kdbarang = b.kdbarang'*). Kondisi yang selanjutnya adalah menampilkan data barang sesuai dengan data barang yang ada ditabel transaksi. Hal itu bisa dilihat dari penggalan program berikut:

```

while not query1.Eof do begin
ckdbarang.Items.Add (query1[kdbarang' ]);
query1.Next;
end

```

Program untuk menampilkan data pada grid diperoleh dari penggalan program berikut:

```

query1.Close;
query1.Open;

```

> Program menampilkan rincian data barang ketikan kode barang terpilih

```
procedure TForm14.ckdbarangClick(Sender: TObject);
var sql : string;
begin
  sql := 'Select p.jmlbeli, b.kdbarang, b.nmbarang, b.hrgjual, b.stok from
  transaksi p, barang b where kdbarang like "' + ckdbarang.Text + '%"
  and p.kdbarang = b.kdbarang';
  query2.SQL.Clear;
  query2.SQL.Add(sql);
  query2.Close;
  query2.Open;
  ekdbarang.Text := query2['kdbarang'];
  enmbarang.Text := query2['nmbarang']; estok.Text
  := query2['stok']; ehrgjual.Text := query2['hrjual'];
  ejmlbeli.Text := query2['jmlbeli'];
  query2.SQL.Add(format(sql,[c]));
  query2.ExecSQL;
```

Penggalan program tersebut untuk mengupdate data barang sesuai dengan kondisi kode barang sama dengan kode barang yang dipilih. *%d* adalah format masukan yang diijinkan didalam penggunaan perintah SQL. (*%s* untuk data dengan tipe string, *%d* untuk field dengan tipe data integer, *%n* untuk field data dengan tipe numeric). *query2.SQL.Add(format(sql,[c]));* adalah penggalan program untuk menambahkan program kedalam Query. Eksekusi program dilaksanakan dengan perintah *query2.ExecSQL;*.

```
strsql := 'Delete from transaksi where kdbarang = "' + ckdbarang.Text
+ '" and notrans = "' + enotransaksi.Text + '"';
query3.SQL.Clear;
query3.SQL.Add(strsql);
query3.ExecSQL;
```

Adalah penggalan program untuk menghapus data barang yang ada ditabel transaksi. Hal itu di dikarenakan pada query diberikan perintah *'Delete from transaksi where kdbarang = "' + ekdbarang.Text + '" and notrans = "' + enotransaksi.Text + '"';*.

```
strsqll := 'Select p.notrans, p.jmlbeli, p.kdbarang, b.nmbarang, b.hrgjual,
p.subtotal from transaksi p, barang b where p.kdbarang = b.kdbarang';
query1.SQL.Clear;
query1.SQL.Add(strsqll);
query1.Close;
query1.Open;
```

Adalah penggalan program untuk menampilkan data transaksi setelah adanya pengurangan data pengembalian (retur).

BAB XVIII **PEMBUATAN LAPORAN.**

Delphi 7 menyediakan dua library untuk membuat Report, yaitu Quick Report dan Rave Report. Quick report sudah menjadi bagian dari Delphi 3. Didalam Delphi 7, Quick Report akan digantikan oleh Rave Report. Ketika kita menginstall Delphi Quick Report tidak secara otomatis diinstall, sehingga kita perlu menginstallnya sendiri.

Didalam Modul matakuliah Praktek Algoritma dan Pemrograman II (Delphi) ini akan menggunakan Rave Report dan diharapkan mahasiswa dapat mengembangkan penggunaan Rave Report secara lebih kompleksitas lagi untuk menghasilkan Dokumen Output yang lebih baik lagi.

A. Pembuatan Report List

Dalam setiap pembuatan program aplikasi, kita tentu perlu untuk membuat Report List yang hanya menampilkan sebuah daftar dengan judul , Header dan Footer.

Langkah-langkah Pembuatan Delphi

1. Masukkan Komponen RVProject dan RVDatasetConnectio kedalam Form Penjualan didalam Kasus I yang sudah dikerjakan sebelumnya.

The screenshot shows a Delphi application window titled "Form Penjualan". The form is titled "FORM TRANSAKSI PENJUALAN PT. POHAN JAYA tbk". It contains several input fields and a table. The fields are arranged in a grid-like layout. At the top, there are four small icons representing different report components: RV RAVE, RV QRY, RV QRY, and RV QRY. Arrows point from the text "RVProject" and "RVDatasetConnectio" to the first and second icons respectively. The form fields include:

- NO FAKTUR
- NO CUSTOMER
- NAMA CUSTOMER
- NO. TELP
- KODE BARANG
- NAMA BARANG
- UKURAN
- HARGA
- JUMLAH BELI
- SUBTOTAL
- TOTAL
- UANG BAYAR
- UANG KEMBALI

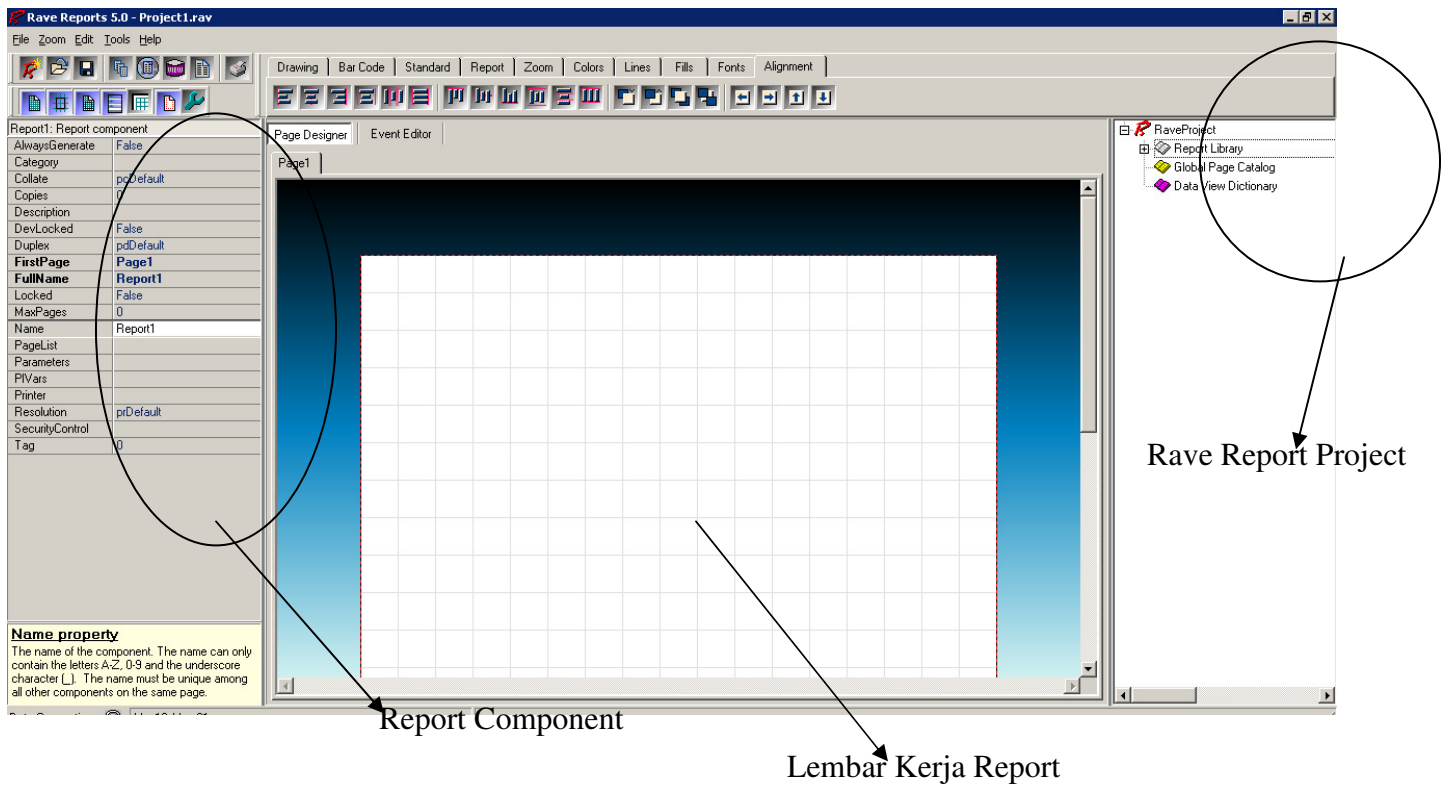
At the bottom, there is a table with columns: Idbrg, Nmbrg, Ukuran, Harga, Qty, Subtotal. Below the table are buttons for "Input Baru", "ADD", "SAVE", and "EXIT".

Gambar 2.1 Form Transaksi

- Koneksikan RVDatasetConnection ke masing-masing Tabel.

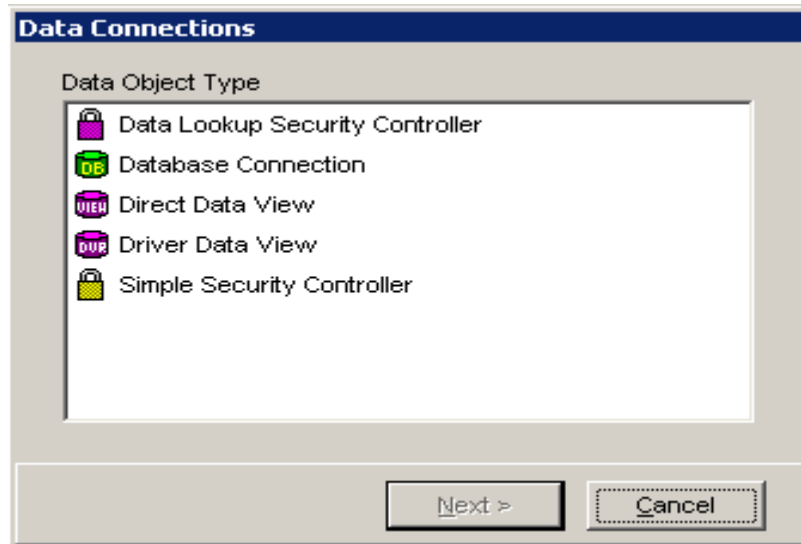
No	Komponen	Properties
1	RVDatasetConnection1	Dataset : tbltrans
2	RVDatasetConnection2	Dataset : tbldetail
3	RVDatasetConnection3	Dataset : tblbrg
4	RVDatasetConnection4	Dataset : tblcust

- Setelah dikoneksikan, Doble klik RV Project untuk membuka Layar Rave Report seperti tampilan dibawah ini:



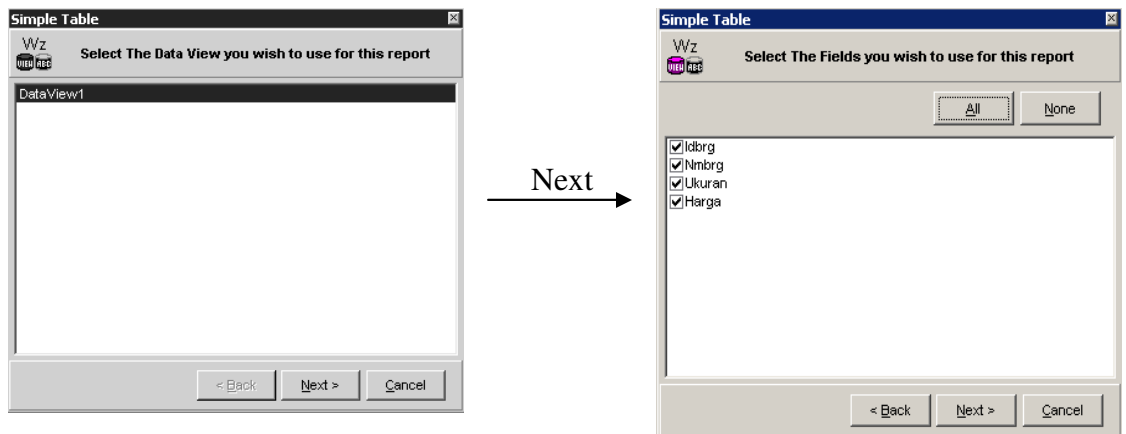
Gambar 2.2 Tampilan Rave Report

- Selanjutnya kita pilih tabel yang akan kita buat Laporan List dengan cara pilih File-New data Object

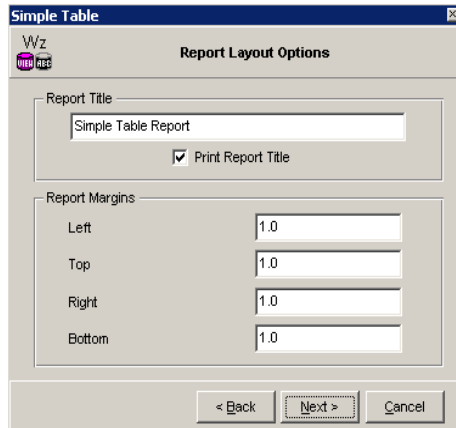


Gambar 2.3 Tampilan Data Connection

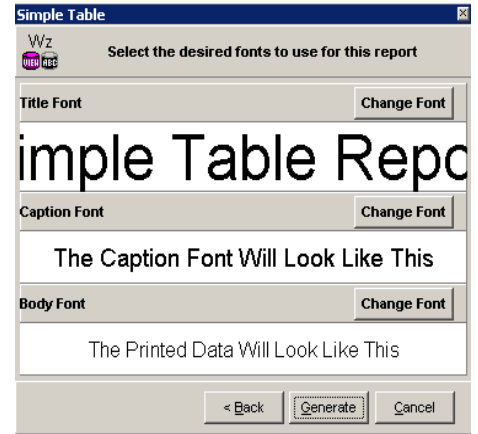
- Pilih Menu Direct Data View, lalu pilih RVDatasetConnection3 untuk mengkoneksikan ke Tabel Barang setelah itu Finish.
- Setelah memilih koneksinya, selanjutnya kita membuat Laporan menggunakan Wizard dengan cara :
 - Pilih Menu bar Tool-Report wizard-Simple Table.
 - Maka akan tampil layar seperti dibawah ini :



- Pilih Field yang akan ditampilkan-pilih field yang akan dijadikan Unik yaitu "Idbrg"



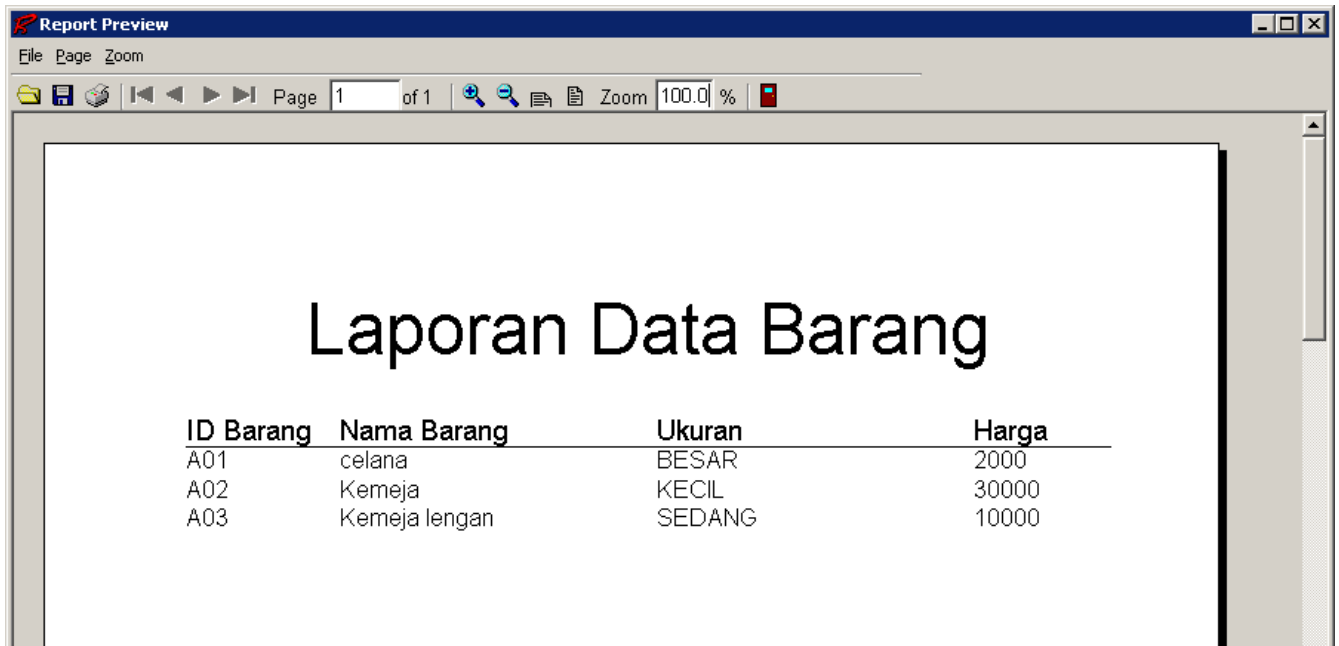
Next
→



- ⇒ Report Layout berfungsi untuk menentukan Bentuk tampilan Report
- ⇒ Selanjutnya disetiap Band akan dapat diganti Font nya masing-masing.
- ⇒ Tekan tombol Generate untuk membuat laporannya. Maka tampilannya akan seperti dibawah ini :

▼ DataView1Region DataView1TitleBand (BGRDrgb 1PC)			
Laporan Data Barang			
▼ DataView1Region DataView1Band (BGRDrgb 1PC)			
idbrg \Nmbrg		Ukuran	Harga
◆ DataView1Region DataView1DataBand (Master 1PC)			
{idbrg} \Nmbrg		{Ukuran}	{Harga}

7. Untuk Melihat Hasilnya dengan cara : File-Execute Report.

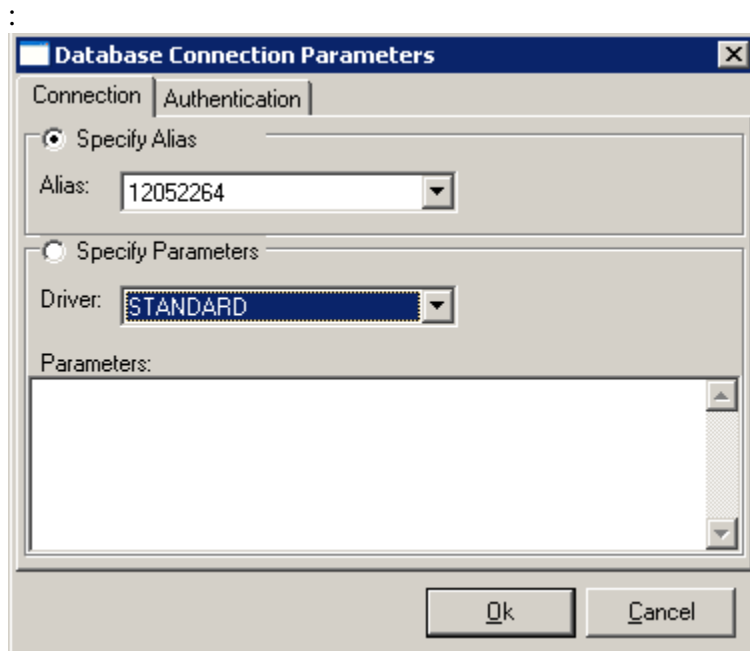


Gambar 2.4 Tampilan Laporan Barang

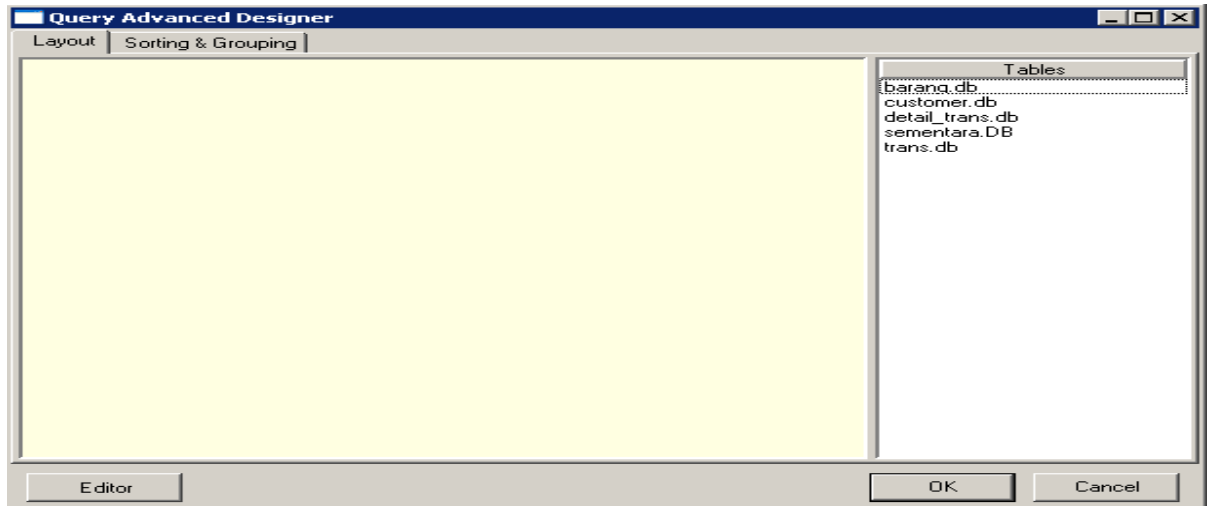
8. Setelah itu simpan Project Reportnya.

B. Pembuatan Report Relational

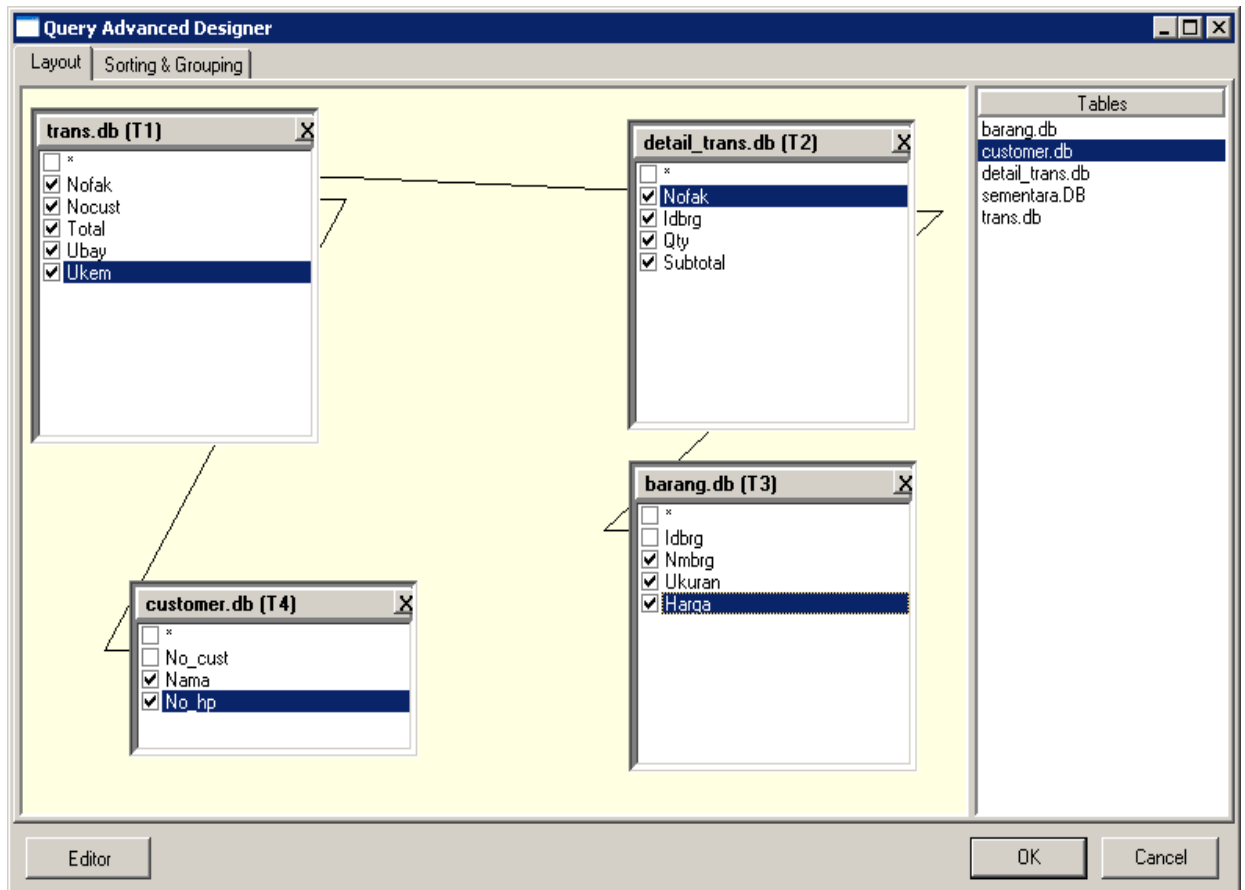
1. Pilih Menu File-New Data Object
2. Pilih Database Connection-BDE. Selanjutnya akan tampil gambar dibawah ini



3. Tentukan Alias yang sudah dibuat dan Driver nya “Standard”. Lalu OK
4. Selanjutnya kita buat Relasi antar tabel dengan cara : File-New data Object-Driver data View-Database1(Sesuai koneksi yang sudah dilakukan di Perintah ke-3).



5. Selanjutnya Drag Tabel yang akan direlasikan kedalam Query Advanced Designer. Selanjutnya Ceklist Semua Field kecuali Idbrg pada Tabel Barang dan Nocust pada Tabel Customer. Lalu OK



6. Tools – Report Wizard - Simple Tabel – DriverDataView1.
7. Dengan langkah yang sama dengan pembuatan Report List akan menghasilkan Tampilan Report seperti dibawah ini :

Laporan Transaksi penjualan

Nofak	Idbrg	Nmbrg	Ukuran	Harga	Qty	Subtotal	Total	Nocust	Nama
001	A01	celana	BESAR	2000	2	4000	9400	00001	ABQ
001	A02	Kemeja	KECIL	3000	3	9000	9400	00001	ABQ
002	A01	celana	BESAR	2000	2	4000	1400	00002	Jejen
002	A03	Kemeja lengan	SEDANG	1000	1	1000	1400	00002	Jejen
003	A01	celana	BESAR	2000	1	2000	6200	00003	Udit
003	A02	Kemeja	KECIL	3000	2	6000	6200	00003	Udit
004	A02	Kemeja	KECIL	3000	1	3000	5000	00001	ABQ
004	A03	Kemeja lengan	SEDANG	1000	2	2000	5000	00001	ABQ
005	A03	Kemeja lengan	SEDANG	1000	2	2000	2000	00002	Jejen

8. Silahkan di Design laporan seperti diatas.

BAB XIX

PEMBUATAN MENU UTAMA.

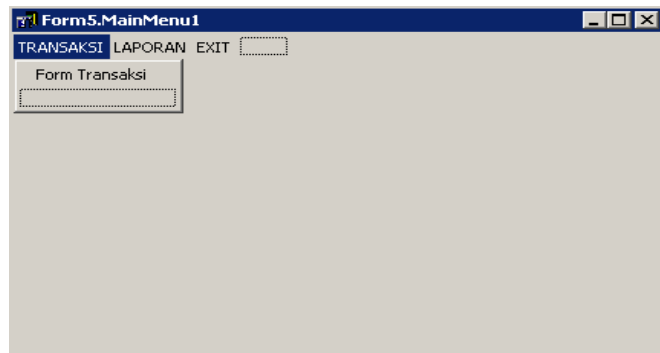
Sekarang tiba saatnya kita membuat Menu utama dari program yang sudah kita buat di Kasus 1. Menu utama yang akan kita buat berbentuk menu pull-down yaitu menu yang mempunyai pilihan (menu) utama berjajar secara mendatar dibagian atas jendela program. Setiap menu utama, dapat mempunyai submenu yang berada pada kotak dibawah tiap menu utama dan submenu tersebut diatur bertumpuk dalam kotak submenu. Submenu hanya akan ditampilkan jika menu utama sedang terpilih atau aktif. Tiap submenu, nantinya dapat mempunyai sub-submenu yang ditampilkan disebelah kanan submenu yang bersangkutan dalam kotak sub-submenu. Demikian seterusnya.

Adapun cara pembuatan Menu utama sebagai berikut :

1. Bukalah Project Penjualan pada kasus 1 yang sudah dibuat.
2. File-New Form untuk membuat Form Menu Utama
3. Pada Menu Standard, pilih Component Main Menu dan diletakkan pada Form.

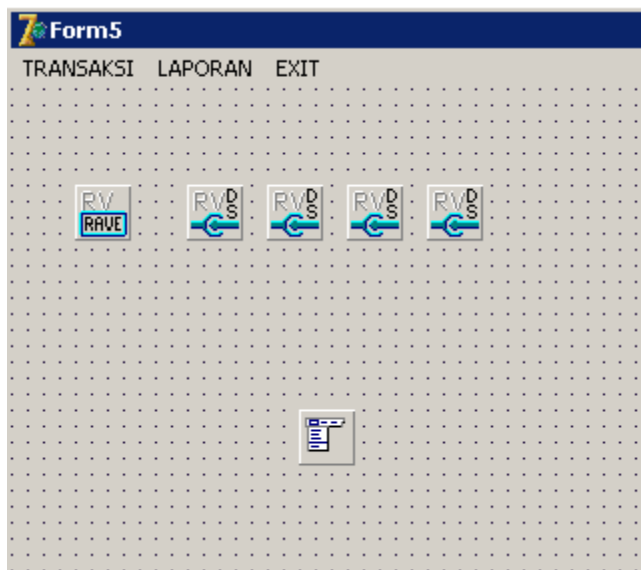


4. Ubah Property Name menjadi :Menuutama. Selanjutnya double klik pada Objek Menuutama maka akan tampil **Menu Designer**. Silahkan Ganti Caption dari Submenu dan Sub-subMenu seperti contoh dibawah ini :



SubMenu	Sub-subMenu
Transaksi	Transaksi Penjualan
Laporan	Laporan Barang
	Laporan Transaksi
Exit	-

5. Setelah itu tutup Jendela Mneu Desaigner. Masukkan *RaveProject* dan *RaveDatasetConnection* dan dengan langkah yang sama dengan Pembuatan Koneksi pada saat pembuatan Laporan menggunakan Rave Report.



6. Lalu Double Klik pada setiap Sub-submenu untuk meletakkan Listing Programnya.

unit terutama;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, Menus, RpDefine, RpRave, RpCon, RpConDS, RpConBDE, DB, DBTables;

type

```
TForm5 = class(TForm)
  MainMenu1: TMainMenu;
  TRANSAKSI1: TMenuItem;
  FormTransaksi1: TMenuItem;
  Laporan1: TMenuItem;
  LaporanBarang1: TMenuItem;
  LaporanTransaksi1: TMenuItem;
  Exit1: TMenuItem;
  RvProject1: TRvProject;
  RvDataSetConnection1: TRvDataSetConnection;
  RvDataSetConnection2: TRvDataSetConnection;
  RvDataSetConnection3: TRvDataSetConnection;
  RvDataSetConnection4: TRvDataSetConnection;
  procedure FormTransaksi1Click(Sender: TObject);
  procedure LaporanBarang1Click(Sender: TObject);
  procedure LaporanTransaksi1Click(Sender: TObject);
  procedure Exit1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form5: TForm5;
```

implementation

```
uses form_penjualan;
```

*{R *.dfm}*

procedure TForm5.FormTransaksi1Click(Sender: TObject);

begin

form4.Show;

end;

procedure TForm5.LaporanBarang1Click(Sender: TObject);

begin

rvproject1.Execute;

rvproject1.ExecuteReport('Report2.rav');

end;

procedure TForm5.LaporanTransaksi1Click(Sender: TObject);

begin

rvproject1.Execute;

rvproject1.ExecuteReport('Report3.rav');

end;

procedure TForm5.Exit1Click(Sender: TObject);

begin

close;

end;

end.

7. Setelah Itu kita dapat menjalankan Program yang sudah di EXE yang secara otomatis terbentuk :

