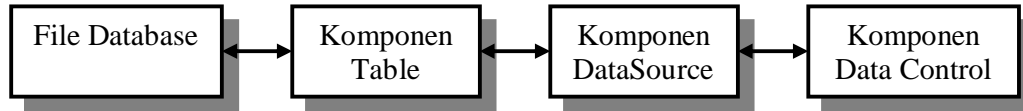


# DASAR-DASAR MEMBUAT PROGRAM DATABASE DENGAN DELPHI

## PENGANTAR DATABASE DENGAN DELPHI

Database dengan menggunakan Delphi menggunakan konsep seperti gambar dibawah ini :



Keterangan :

1. File Database : File database dari sistem database lain seperti Dbase (\*.dbf), Paradox (\*.db), Microsoft Access (\*.mdb), dan lain-lain.
2. Komponen **Table** : Komponen yang mewakili file database. Setiap melakukan proses dalam komponen tabel tersebut, maka isi file database yang terkoneksi ke komponen tersebut berubah juga.
3. Komponen **DataSource** : Komponen penghubung antara komponen tabel dengan komponen data control. Dalam datasource harus diisi tabel yang berelasi ke datasource tersebut.
4. Komponen-Komponen **Data Control** : Komponen yang digunakan untuk menampilkan data-data yang berasal dari datasource (tabel). Data control ada yang berbentuk tabel, label, edit box, gambar, combobox, listbox dan lain-lain.

## PROGRAM DATABASE 1 :

Untuk program pertama ini, program database yang akan dibuat adalah untuk mendata spesies ikan pada suatu toko. (data sudah ada)


1. Buat suatu aplikasi baru dengan **Alt+F - New Application**

2. Tempatkan sebuah komponen **Table**  dari component palette **BDE**.  
Ganti property berikut :


Property	Nilai
DatabaseName	DBDEMOS
TableName	BIOLIFE.DB
Name	Tikan
Active	True

Keterangan :

**DatabaseName** adalah Alias directory atau koneksi ODBC. Pembuatan alias akan diterangkan berikutnya.

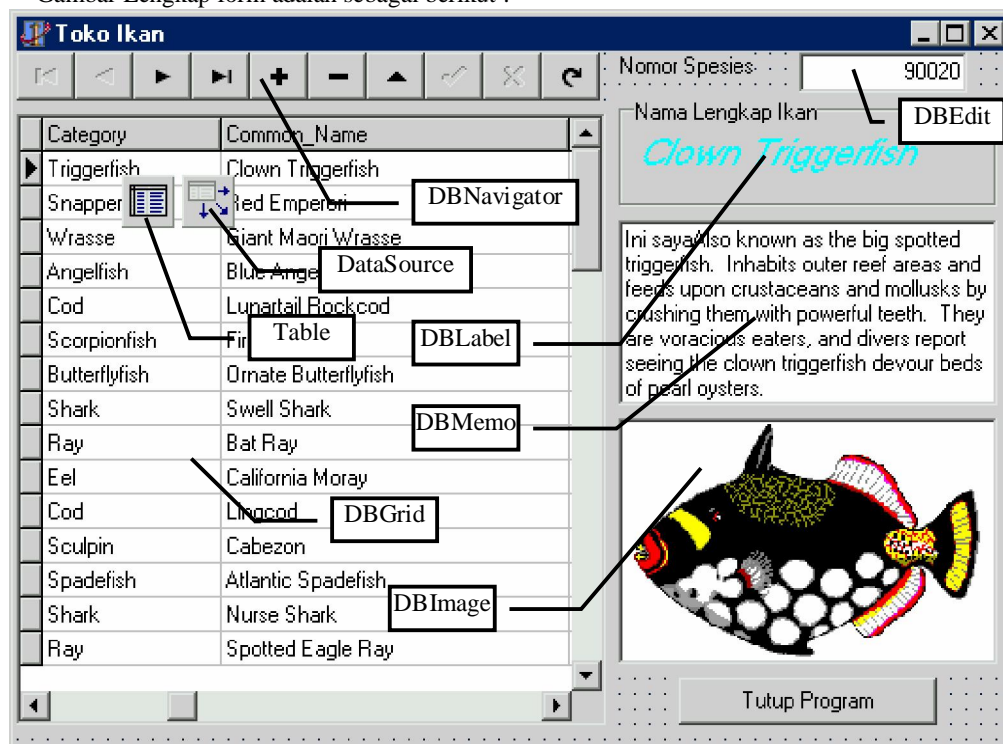
3. Tempatkan sebuah komponen **DataSource**  dari component palette **Data Access**  
Ganti property berikut :

Property	Nilai
DataSet	Tikan
Name	DSIkan

4. Tempatkan sebuah **DBGrid**  dari component palette **Data Controls** pada form kemudian isi properti datasourcenya dengan DSIkan (karena akan menampilkan isi tabel ikan). Coba run program dan lihat hasilnya. Dengan cara yang tadi dijalankan, maka kita sudah dapat melihat, menambah, mengedit data.

5. Tempatkan sebuah **DBNavigator**  dari component palette **Data Controls** pada form. Komponen ini berguna untuk menjelajahi isi tabel yaitu : Pindah ke record pertama, berikutnya (maju), sebelumnya (mundur), terakhir, menambah data, menghapus data, menyimpan data, mengedit data, membatalkan perubahan data dan refresh data. Isi property DataSource dengan DSikan. Kemudian run program untuk melihat kegunaan dari DBNavigator tersebut.
6. Tempatkan sebuah **DBText**  dari component palette **Data Controls** pada form. Fungsi dari komponen ini sama dengan komponen Label pada component palette Standar. Perbedaannya isi dari DBText berasal dari suatu data field dari suatu tabel. Untuk menghubungkan DBText dengan field, gantilah properti DataSource dengan DSikan dan DataField dengan field yang diinginkan contoh CommonName. Kemudian run program sehingga kita dapat melihat setiap record aktif pindah maka isi DBText berisi field CommonName record yang aktif.
7. Tempatkan sebuah **DBEdit**  dari component palette **Data Controls** pada form. Fungsi dari komponen ini sama dengan komponen Edit perbedaannya isi DBEdit mengacu kepada suatu data field dari suatu tabel. Untuk menghubungkannya dengan suatu field, isi property DataSource dengan DSikan dan FieldName dengan Spesies No.
8. Untuk menampilkan data bertipe memo  dari component palette **Data Controls**, gunakan komponen **DBMemo**, kemudian ganti property Datasource dengan DSikan dan FieldName dengan Notes.
9. Untuk menampilkan data bertipe gambar, gunakan komponen **DBImage**  dari component palette **Data Controls**, kemudian ganti property Datasource dengan DSikan dan FieldName dengan Graphics.
10. Tempatkan suatu tombol kemudian ganti caption dengan Tutup Program dan kemudian isi even Onclick dengan perintah Close; atau Application.Terminate.

Gambar Lengkap form adalah sebagai berikut :



The screenshot shows a Delphi form titled "Toko Ikan". It features a table with two columns: "Category" and "Common\_Name". The table lists various fish species. To the right of the table, there are several data controls: a "DBNavigator" for navigating through the records, a "DBText" (labeled "DBLabel") for displaying the "Common\_Name", a "DBEdit" for editing the "Spesies No.", a "DBMemo" for displaying the "Notes", and a "DBImage" for displaying the "Graphics". A "Tutup Program" button is located at the bottom right. The form is annotated with labels pointing to these components: "DBNavigator", "DataSource", "Table", "DBLabel", "DBMemo", "DBGrid", "DBImage", and "DBEdit".

Category	Common_Name
Triggerfish	Clown Triggerfish
Snapper	Red Emperor
Wrasse	Giant Maori Wrasse
Angelfish	Blue Angel
Cod	Lunartail Rockcod
Scorpionfish	Fir
Butterflyfish	Ornate Butterflyfish
Shark	Swell Shark
Ray	Bat Ray
Eel	California Moray
Cod	Linecod
Sculpin	Cabezon
Spadefish	Atlantic Spadefish
Shark	Nurse Shark
Ray	Spotted Eagle Ray

Form Details:

- Nomor Spesies: 90020
- Nama Lengkap Ikan: Clown Triggerfish
- Notes: Ini saya also known as the big spotted triggerfish. Inhabits outer reef areas and feeds upon crustaceans and mollusks by crushing them with powerful teeth. They are voracious eaters, and divers report seeing the clown triggerfish devour beds of pearl oysters.
- Graphics: Image of a Clown Triggerfish.
- Tutup Program

**PEMBUATAN ALIAS**

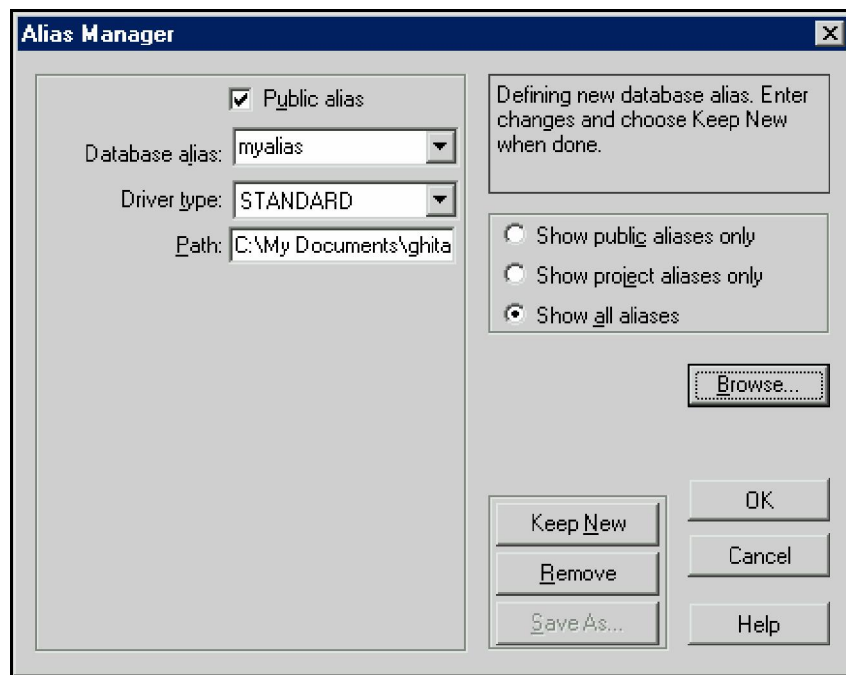
Alias digunakan sebagai short cut dari suatu sub directori/folder. Sebenarnya dalam tabel kita bisa menggunakan alamat langsung dari file yang akan diakses contohnya : C:\MY DOCUMENTS\DATA\KARYAWAN.DB, hal tersebut bisa dilakukan tapi kalau file karyawan.db dipindah maka program harus dirubah. Untuk menghindari hal seperti itu, maka dibuatlah suatu Alias. Contoh kita buat suatu alias bernama Karyawan yang mewakili C:\My Documents\Data. Kalau data karyawan tersebut dipindah maka kita tidak lagi mengubah programnya tapi hanya membelokan alias tersebut ke tempat yang baru.

Cara pembuatan alias adalah sebagai berikut :

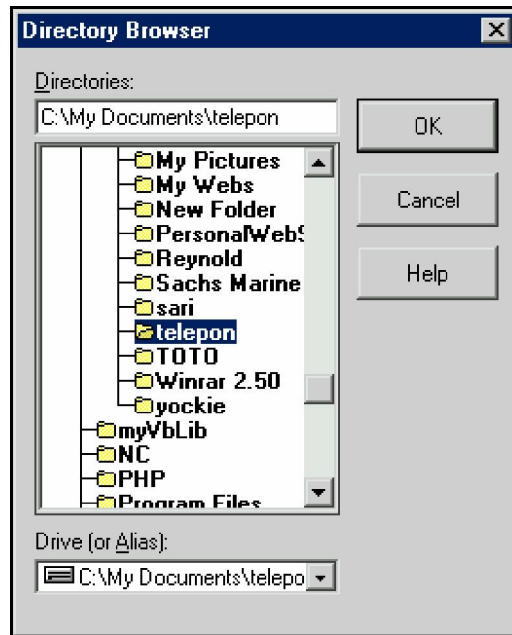
1. Masuk ke **Database Desktop** yang ada pada **Start Menu – Program – Borland Delphi – Database Desktop** atau pilih menu **Tool – DataBase Desktop**.
2. Pilih menu **Tools – Alias Manager**



3. Klik tombol **New**, kemudian isi edit box Database alias dengan alias yang diinginkan, kemudian klik tombol **Browse**.



4. Pilih drive yang diinginkan pada combo (drive or alias), kemudian pilih direktori/folder yang dituju pada list **Directories**. Kemudian tekan tombol OK.



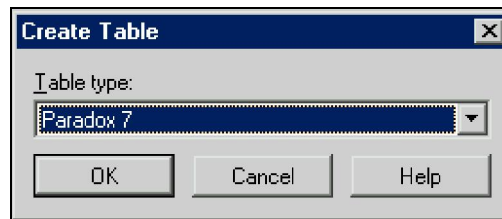
5. Jika telah benar tekan tombol **OK** pada Alias Manager. Maka akan muncul dialog apakah alias tersebut akan disimpan dalam file konfigurasi. Tekan tombol **Yes**.

**PEMBUATAN TABEL**

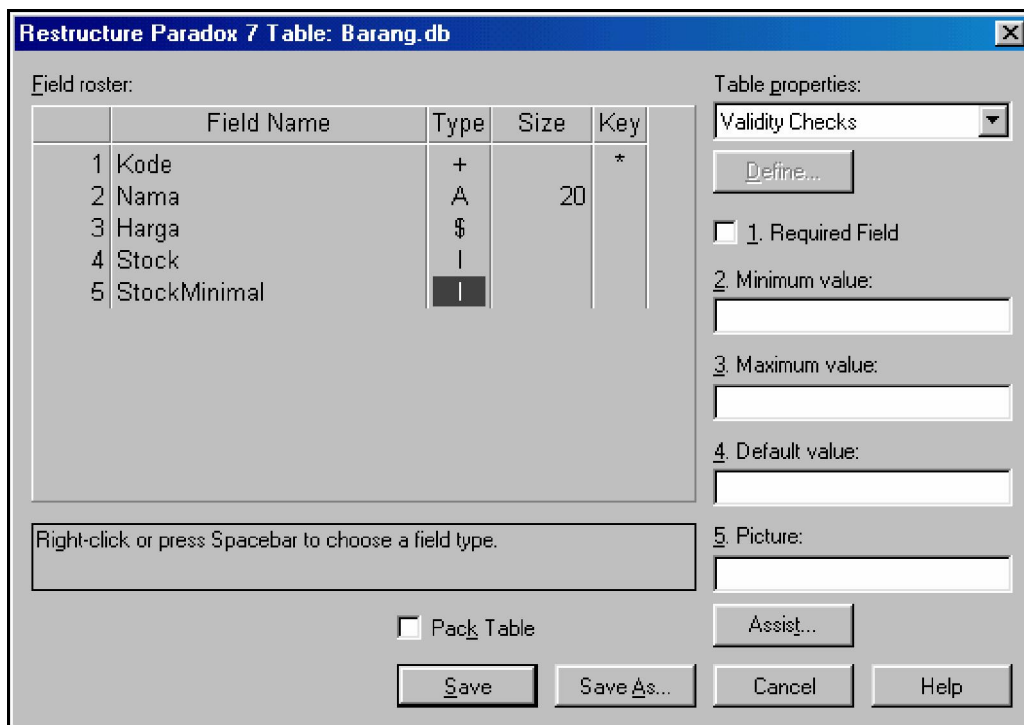
Pembuatan tabel yang akan diakses oleh Delphi bisa dibuat dengan Database Desktop atau MsAccess.

Cara pembuatan tabel dalam Database Desktop :

1. Masuk ke program **Database Desktop**
2. Pilih menu **File – New – Table** sehingga akan muncul dialog berikut yang akan menanyakan tabel jenis apa yang akan dibuat apakah paradox atau dbase atau yang lain, kemudian tekan OK.



3. Maka akan muncul window pembuatan tabel seperti berikut :



4. Jika sudah lengkap tekan tombol **Save As** untuk menyimpan tabel tersebut.
5. Tulis nama file tabel yang sudah dibuat dan simpan pada alias yang tadi dibuat.
6. Tabel sudah bisa diakses oleh Delphi.

*Jenis-jenis field dapat dilihat di halaman berikutnya.*

Jenis field-field dalam Paradox yang sering dipakai :

Simbol Jenis Field	Jenis Field	Keterangan
A	AlphaNumeric	String, Alpabet, Numerik
I	Integer	Integer (Bilangan Bulat)
N	Number	Pecahan
D	Date	Tanggal
T	Time	Jam
M	Memo	Memo / Catatan panjang
G	Graphic	Gambar
+	AutoIncrement	Otomatis naik tiap ada penambahan data
\$	Money	Mata Uang
S	Short Int	Integer kecil (-32768 s/d 32767)

#### PENGISIAN DATA

Pengisian data pada tabel bisa dilakukan dengan dua cara, yaitu :

1. Melalui Database Desktop  
Buka Database Desktop, kemudian pilih **File – Open – Table** kemudian pilih file database mana yang akan diisi.
2. Melalui program yang dibuat

## Contoh Aplikasi Database 1

Untuk kasus aplikasi sekarang, kita akan membuat suatu program untuk pengolahan data barang menggunakan tabel yang telah dibuat dan memiliki fasilitas berikut :

- Navigator Data
- Penambahan Data
- Pengeditan Data
- Penghapusan Data
- Pencarian Data

Untuk membuat hal tersebut lakukan langkah berikut :

1. Buatlah form seperti berikut

The screenshot shows a Delphi form titled "Pengolahan Data Barang". Inside the form, there is a table titled "Data Barang" with the following columns: Kode, Nama, Harga, Stock, and StockMinimal. Below the table, there are four speed buttons (represented by hand icons) for navigation. To the right of these buttons is a label "Bitbtn" with a line pointing to the "Tambah" (Add) button. Below the navigation buttons is a label "SpeedButton" with a line pointing to the first speed button. To the right of the speed buttons is a text field labeled "Kode Barang". To the right of the text field are three buttons: "Tambah", "Edit", and "Hapus". Below these buttons are two more buttons: "Pencarian" and "Tutup".

Catatan : Nama tabel adalah **TBarang**

2. Program untuk tombol navigator (4 buah speedbutton bergambar tangan) adalah sebagai berikut :

- Program untuk **Pindah Ke Data Pertama** (gambar tangan menunjuk ke atas)
 

```
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
    TBarang.First;
end;
```
- Program untuk **Pindah Ke Data Sebelumnya / Mundur** (gambar tangan menunjuk ke kiri)
 

```
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
    TBarang.Prior;
    if TBarang.BOF then
        ShowMessage('Anda sudah di awal data');
end;
```
- Program untuk **Pindah Ke Data Berikutnya / Maju** (gambar tangan menunjuk ke kanan)
 

```
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
    TBarang.Next;
    if TBarang.EOF then
        ShowMessage('Anda sudah di akhir data');
```

```
end;
```

- Program untuk **Pindah Ke Data Terakhir** (gambar tangan menunjuk ke bawah)

```
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
    TBarang.Last;
end;
```

- Untuk proses penambahan dan pengeditan data, diperlukan suatu form baru yang digunakan untuk pengisian data barang tersebut. Buat suatu form seperti berikut :
  - Buat form baru dengan **File – New – Form** dan namailah dengan **FIsiBarang**.
  - Tempatkan 4 buah Edit untuk tempat pengisian data. Kemudian atur properti Name-nya sesuai dengan kegunaannya.
  - Tempatkan tombol suatu tombol kemudian **caption** diisi dengan **Save** kemudian properti **modalresult** diisi dengan mrOk.
  - Tempatkan tombol suatu tombol kemudian **caption** diisi dengan **Cancel** kemudian properti **modalresult** diisi dengan mrCancel.

- Buatlah objek/component field untuk mempermudah dalam pengisian program dengan cara mendouble klik di componen Table kemudian klik kanan dan pilih **Add All field**. Dengan melakukan hal tersebut, maka setiap field mempunyai objek field khusus.
- Isilah program berikut pada event onclick tombol Tambah.

```
Procedure TForm1.TblTambahClick(Sender : TObject);
begin
    FIsiBarang.ENama.Text:='';
    FIsiBarang.EHarga.Text:='0';
    FIsiBarang.EStock.Text:='0';
    FIsiBarang.EStockMinimal.Text:='0';
    FIsiBarang.ShowModal; //tampilkan Form Fisibarang
    if FIsiBarang.ModalResult=mrOk then
    begin
        TBarang.Append;//meminta tempat kosong untuk data baru
        TBarangNama.Value:=FIsiBarang.ENama.Text;
        TBarangHarga.Value:=StrToFloat(FIsiBarang.EHarga.Text);
        TBarangStock.Value:=StrToInt(FIsiBarang.EStock.Text);
        TBarangStockMinimal.Value:=StrToInt(FIsiBarang.
                                                EStockMinimal.Text);

        TBarang.Post;//simpan data
    end;
end;
```

6. Isilah program berikut untuk event onclick dari Tombol Edit

```
Procedure TForm1.TblEditClick(Sender : TObject);
begin
  FisiBarang.ENama.Text:=TBarangNama.Value;
  FisiBarang.EHarga.Text:=FloatToStr(TBarangHarga.Value);
  FisiBarang.EStock.Text:=IntToStr(TBarangStock.Value);
  FisiBarang.EStockMinimal.Text:=
      IntToStr(TBarangStockMinimal.Value);
  FisiBarang.ShowModal; //tampilkan Form Fisibarang
  if FisiBarang.ModalResult=mrOk then
  begin
    TBarang.Edit; //menyiapkan data untuk diedit
    TBarangNama.Value:=FisiBarang.ENama.Text;
    TBarangHarga.Value:=StrToInt(FisiBarang.EHarga.Text);
    TBarangStock.Value:=StrToInt(FisiBarang.EStock.Text);
    TBarangStockMinimal.Value:=StrToInt(FisiBarang.
      EStockMinimal.Text);

    TBarang.Post;
  end;
end;
```

7. Untuk event onclick pada Tombol Hapus tuliskan perintah sebagai berikut :

```
Procedure TForm1.TblHapusClick(Sender : TObject);
begin
  if MessageDlg('Data Mau Dihapus ?',mtconfirmation,
    [mbyes,mbno],0)=mryes then
  begin
    TBarang.Delete;
  end;
end;
```

Keterangan :

Perintah MessageDlg digunakan untuk melakukan suatu dialog dengan user.

Cara penulisan MessageDlg adalah :

**MessageDlg(Pesan, TipeDialog, [DaftarTombol],NoHelp).**

Parameter pertama yaitu Pesan harus bertipe string.

Parameter kedua bisa berisi tipe dari dialog. Parameter ini bisa berisi:

- **mtconfirmation** : Ada lambang tanda tanya dalam dialognya
- **mtwarning** : Ada lambang peringatan / tanda seru (!)
- **mtinformation** : Ada lambang huruf i
- **mterror** : Ada lambang silang berwarna merah
- **mtcustom** : Tidak ada lambang dalam dialognya.

Parameter ketiga merupakan parameter yang mengatur tombol mana yang akan digunakan.

Parameter ini bisa berisi :

**MrNone, mrAbort, mrYes, mrOk, mrRetry, mrNo, mrCancel, mrIgnore, mrAll**

Fungsi Message dialog ini akan menghasilkan nilai sesuai dengan tombol yang ditekannya.

Jika tombol OK yang ditekan, maka MessageDlg ini berisi MrOK,

Jika tombol yes yang ditekan, maka MessageDlg ini berisi MrYes dst

8. Untuk tombol pencarian, maka gunakan perintah berikut :

Untuk pencarian pada field key :

```
Procedure TForm1.TblCariClick(Sender : TObject);
begin
  if Tbarang.Findkey([Ecari.Text])=false then
    ShowMessage('Data yang dicari tidak ditemukan');
end;
```

Fungsi Findkey akan memberikan nilai true jika data ditemukan dan akan mengembalikan nilai false jika data tidak ditemukan.

9. Cara lain untuk penambahan dan pengeditan data adalah dengan menggunakan komponen data control. Lebih jelasnya ikutilah langkah-langkah berikut :
- Buat form baru dengan File – New – Form beri nama FIsiBarang2
  - Tempatkan 5 buah DBEdit untuk tempat pengisian data. Kemudian atur properti DataSource dan DataField dari tiap DBEditnya. (lihat form)
  - Tempatkan tombol suatu tombol kemudian **caption** diisi dengan **Save** kemudian properti **modalresult** diisi dengan mrOk.
  - Tempatkan tombol suatu tombol kemudian **caption** diisi dengan **Cancel** kemudian properti **modalresult** diisi dengan mrCancel.

10. Untuk onlick pada Tombol Tambah, ketikan perintah berikut :

```

Procedure TForm1.TblTambahClick(Sender : TObject);
begin
    TBarang.Append;
    FIsiBarang2.ShowModal;
    if FIsiBarang2.ModalResult=mrOk then
    begin
        TBarang.Post;
    end
    else
    begin
        TBarang.Cancel;
    end;
end;

```

11. Untuk onlick pada Tombol Edit, ketikan perintah berikut :

```

Procedure TForm1.TblEditClick(Sender : TObject);
begin
    TBarang.Edit;
    FIsiBarang2.ShowModal;
    if (FIsiBarang2.ModalResult=mrOk) then
    begin
        TBarang.Post;
    end
    else
    begin
        TBarang.Cancel;
    end;
end;

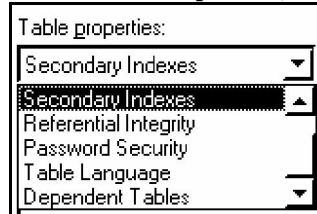
```

## Pembuatan Index Pada Table

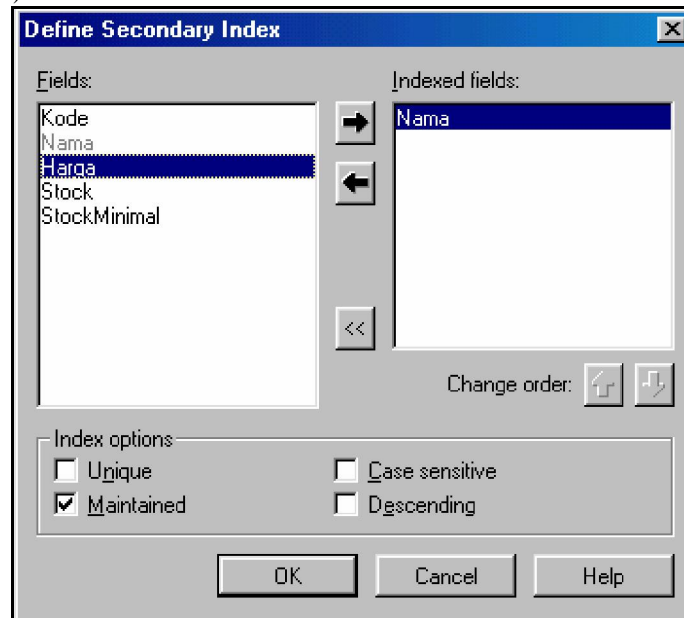
Ketika pembuatan suatu tabel, kita membuat suatu field yang pada fieldnya diberi tanda \*. Isi field yang merupakan field kunci tidak boleh ada data yang sama (harus unik). Dengan membuat suatu field sebagai primary key, maka setiap pengisian data, maka urutan data akan otomatis terurut. Selain primary key ada juga yang namanya secondary index yang fungsinya mirip dengan primary key.

Cara pembuatan secondary index adalah :

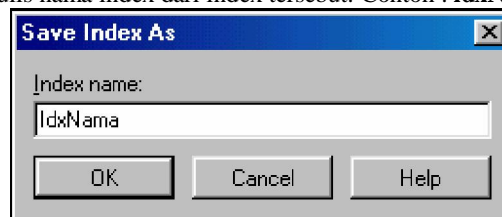
1. Masuk ke **Database Desktop**
2. Pilih menu **Table – Utilities – Restructure**, kemudian pilih tabel yang akan diubah strukturnya.
3. Pilih **Secondary Indexes** pada field **Table Properties** (lihat gambar).



4. Pilih field yang akan dijadikan index. Pilih field **Nama** kemudian tekan tombol panah kanan. (lihat gambar).



5. Jika telah dipilih, tulis nama index dari index tersebut. Contoh : **idxNama**.



6. Buat juga index dengan nama **IdxHarga** yang merupakan index dari field harga dengan ketentuan pengurutannya secara menurun (descending).
7. Jika telah selesai, simpan table tersebut.

## Penggunaan Index dalam Program

Untuk mengaktifkan suatu index yang telah dibuat, gunakan field IndexName yang merupakan properti milik Table. Untuk lebih jelasnya silahkan buat form seperti di bawah ini.

Kode	Nama	Harga	Stock	StockMinimal
1	Buku	\$2,500.00	100	25
2	Mouse	\$25,000.00	20	5
3	Disket	\$2,500.00	20	10
4	Mouse Pad	\$5,000.00	10	15
5	Pulpen	\$3,000.00	25	
6	Sil	\$1,500.00	25	
7	Hapus	\$1,500.00	15	
8	Calculator	\$100,000.00	10	
9	Tinta Printer	\$20,000.00	10	

Untuk bagian “**Pengurutan Berdasarkan**” gunakan RadioGroup.

Untuk OnClicknya tuliskan perintah berikutkan :

```

Procedure TForm1.RadioGroup1Click(Sender:TObject);
begin
    if (RadioGroup1.ItemIndex=0) then
        Tbarang.IndexName:=''
    else
        if (RadioGroup1.ItemIndex=1) then
            Tbarang.IndexName:='idxBarang'
        else
            if (RadioGroup1.ItemIndex=2) then
                Tbarang.IndexName:='IdxHarga';
End;

```

Untuk tombol “**Cari**” tuliskan perintah berikut :

```

Procedure TForm1.CariClick(Sender:TObject);
Var
    Ditemukan:Boolean;
Begin
    Ditemukan:=Tbarang.FindKey([Ecari.Text]);
    if (ditemukan=false) then // atau if not ditemukan then
        ShowMessage('Data tidak ditemukan');
End;

```

Untuk tombol “**Find Nearest**” tuliskan perintah berikut :

```

Procedure TForm1.TfindNearestClick(Sender:TObject);
Begin
    TBarang.FindNearest([Ecari.Text]);
End;

```