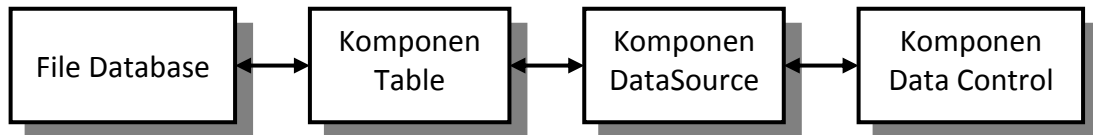




Pengantar Database Dengan Delphi

Database dengan menggunakan Delphi dapat dilihat seperti konsep gambar dibawah ini :




Keterangan :

1. File Database, File database dari sistem database, baik itu dari Paradox(*.db), Microsoft Access (*.mdb), SQL, dan lain-lain.
2. Komponen Table, merupakan komponen di Delphi yang mewakili file database. Setiap melakukan proses dalam komponen tabel tersebut, maka isi file database yang terkoneksi ke komponen tersebut akan berubah juga.
3. Komponen Data Source. Komponen penghubung antara komponen tabel dengan komponen data kontrol. 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, picture, combobox, listbox dan lain – lain.

Program Database e1


Untuk program pertama ini, program database yang akan dibuat adalah untuk melakukan pendataan spesies Ikan dimana databasenya sudah tersedia. Langkah-langkah yang harus dilakukan adalah sebagai berikut :


1. Buat aplikasi baru dengan **File → New Application**
2. Tempatkan sebuah komponen **Table**  dari component palette BDE, kemudian ganti property berikut :

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


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




Property	Nilai
Dataset	T_Ikan
Name	DS_Ikan

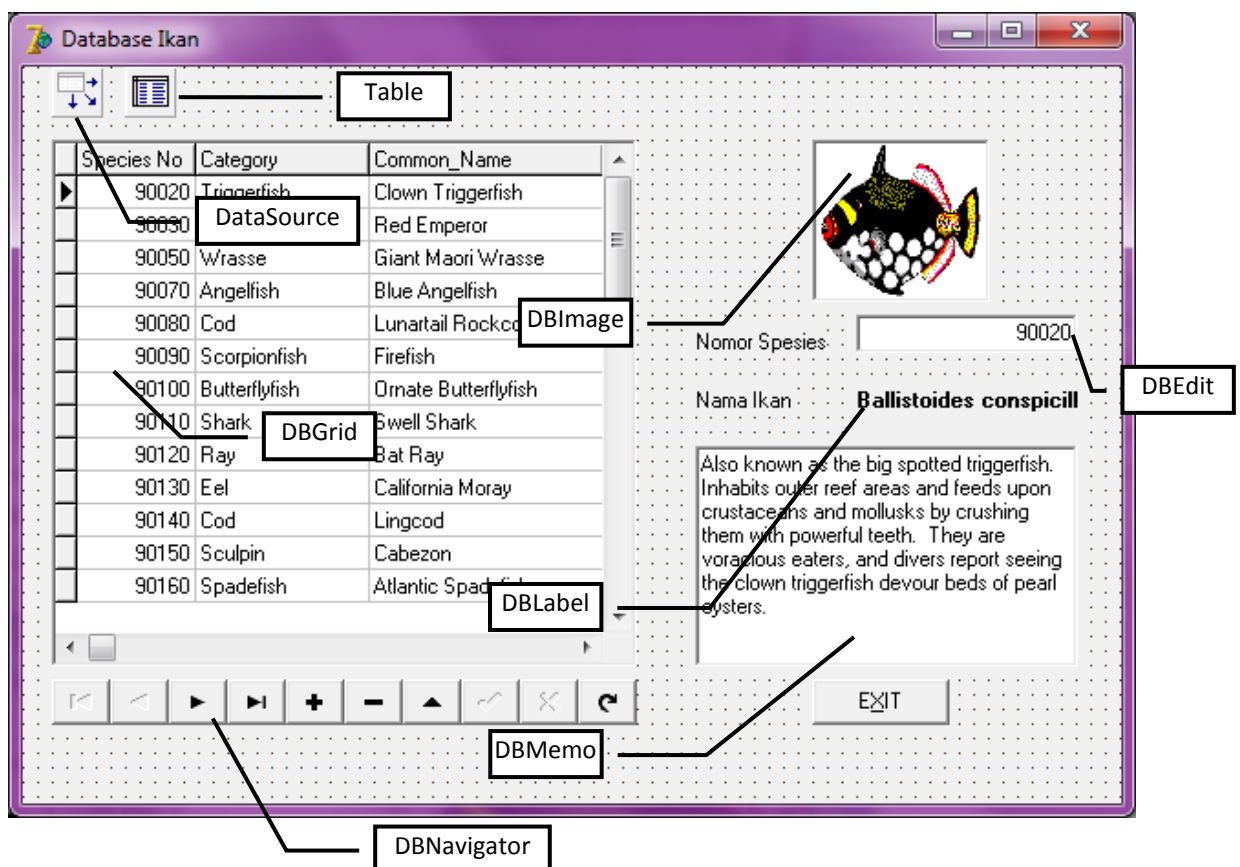
4. Tempatkan sebuah **DBGrid**  dari component Palette Data Controls pada form kemudian isi properti DataSource nya dengan DS_Ikan (karena akan menampilkan isi tabel ikan).

5. Tempatkan sebuah **DBNavigator**  dari Component palette Data Controls pada Form. Komponen ini berguna untuk menjelajahi isi tabel, yaitu :
- Pindah ke record pertama
 - Pindah ke record berikutnya (maju)
 - Pindah ke record sebelumnya (mundur)
 - Terakhir
 - Menambah Data
 - Menyimpan Data
 - Mengedit Data

Isi property **DataSource** dengan DS_Ikan. Kemudian Run Program untuk hasilnya.

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, maka gantilah properti **DataSource** dengan dengan DS_Ikan dan **DataField** dengan field yang diinginkan.

7. Tempatkan sebuah **DBEdit**  dari component Palette Data Controls pada Form. Fungsi dari komponen ini sama dengan komponen Edit tapi perbedaannya isi DBEdit mengacu kepada suatu data field dari suatu tabel. Untuk menghubungkannya dengan suatu field, isi property DataSource dengan DS_Ikan dan FieldName dengan SpesiesNo
8. Untuk menampilkan data bertipe **DBMemo**  dari Component Palette Data Controls, gunakan komponen DBMemo, kemudian ganti Property Datasource dengan DS_Ikan dan FieldName dengan Notes.
9. Untuk menampilkan data bertipe gambar, gunakan komponen **DBImage**  dari component palette Data Controls, kemudian ganti property DataSource dengan DS_Ikan dan FieldName dengan Graphics.
10. Tempatkan suatu tombol kemudian ganti caption dengan **Exit**

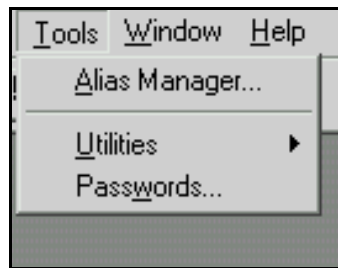


Pembuatan Alias

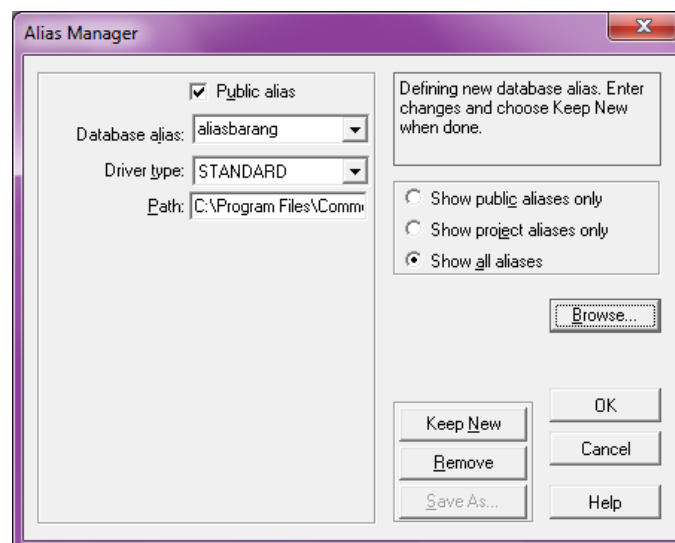
Alias digunakan sebagai langkah cepat dari suatu Sub Directory atau Folder. Sebenarnya dalam tabel kita bisa menggunakan alamat alngsung dari File yang akan kita akses. Contohnya **C:\Data\Mahasiswa.Db**, hal tersebut bisa dilakukan tapi jika file mahasiswa.db dipindahkan maka program pun harus ikut diubah. Maka untuk menghindari hal yang seperti itu, dibuatlah suatu Alias. Contoh kita akan buat suatu alias bernama mahasiswa yang mewakili **C:\Data\Mahasiswa.Db**, sehingga ketika data mahasiswa tersebut dipindah, maka kita tidak perlu lagi mengubah programnya tetapi hanya mengubah alias tesebut ke tempat yang baru.

Untuk membuat Alias, caranya adalah sebagai berikut :

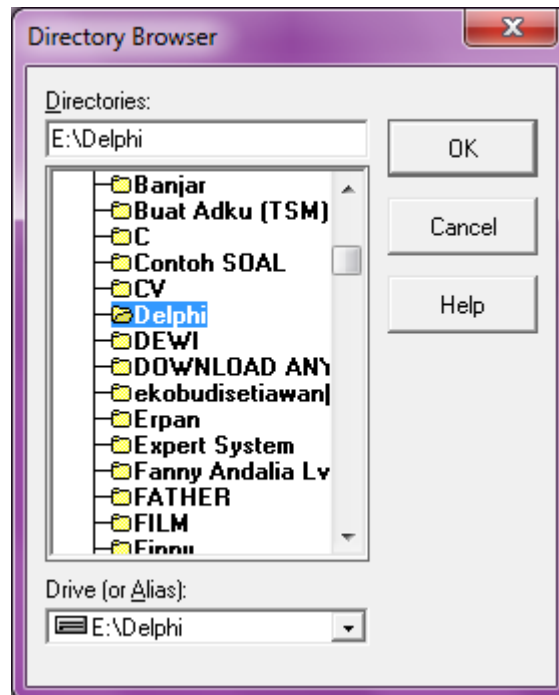
1. Masuk ke **Database Desktop** yang ada pada **Start Menu – Program – Borland Delphi7 – Database Desktop**.
2. Pilih menu **Tools – Alias Manager**



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



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



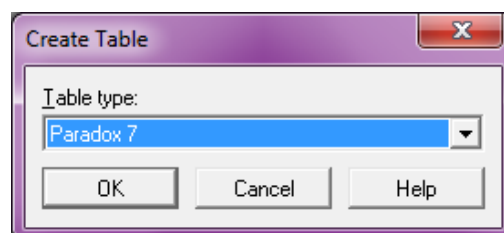
- Jika telah benar tekan tombol **OK** pada Alias Manager. Maka akan muncul dialog apakah alias tersebut akan disimpan dalam file konfigurasi. Tekan tombol **Yes**.

Pembuatan Tabel

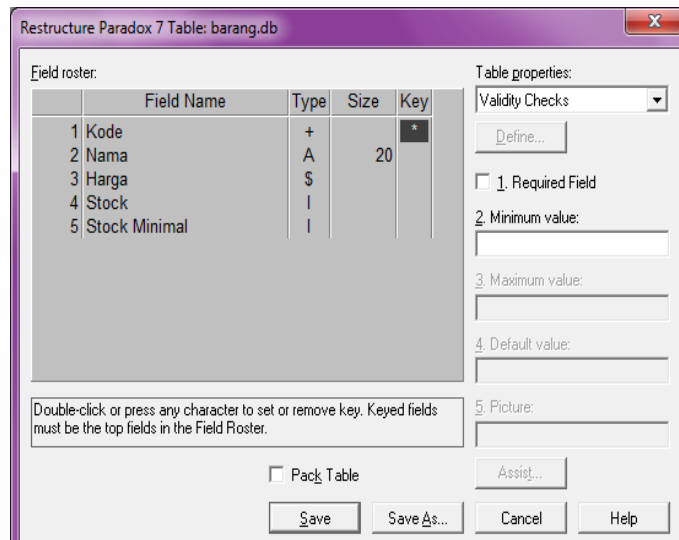
Pembuatan tabel yang akan diakses oleh Delphi bisa dibuat dengan Database Desktop

Cara pembuatan tabel dalam Database Desktop :

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



- Maka akan muncul window untuk membuat tabel seperti berikut ini



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

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

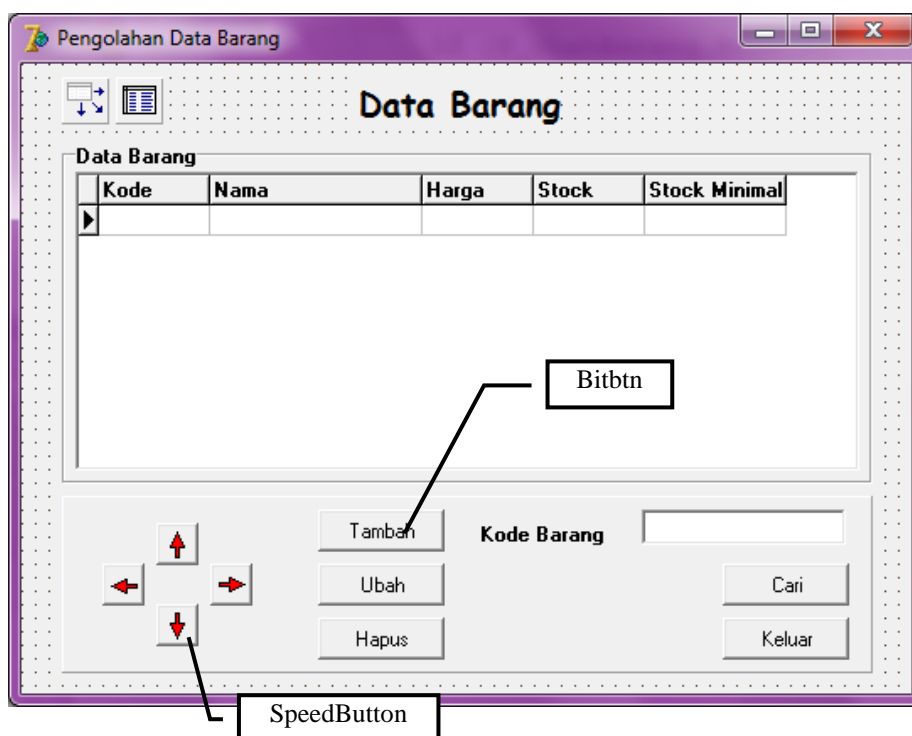
Buka Database Desktop, kemudian pilih **File – Open – Table** kemudian pilih file database mana yang akan diisi.
2. Melalui program yang dibuat

Contoh Aplikasi Database e2

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

Buatlah Form seperti berikut ini dan beri nama **F_DataBarang**:



Nama Tabel adalah T_Barang

1. Program untuk 4 tombol navigator (panah) adalah sebagai berikut :

- Program untuk Pindah ke Data Pertama (**Panah ke atas**)

```
procedure TF_DataBarang.SpeedButton2Click(Sender: TObject);
begin
    T_Barang.First;
end;
```

- Program untuk Pindahke Data Terakhir (**Panah ke bawah**)

```
procedure TF_DataBarang.SpeedButton4Click(Sender: TObject);
begin
    T_Barang.Last;
end;
```

- Program untuk Pindahke Data Sebelumnya (**Panah ke Kiri**)

```

procedure TF_DataBarang.SpeedButton1Click(Sender: TObject);
begin
    T_Barang.Prior;
    if T_Barang.Bof then
        showmessage('Anda sudah berada diawal data');
end;

```

- Program untuk Pindah ke Data Sesudahnya (**Panah ke Kanan**)

```

procedure TF_DataBarang.SpeedButton3Click(Sender: TObject);
begin
    T_Barang.Next;
    if T_Barang.Eof then
        Showmessage('Anda sudah di akhir data');
end;

```

2. Untuk proses penambahan data, diperlukan suatu form baru yang digunakan untuk pengisian data barang tersebut.

- Buat **Form** baru dengan **File → New → Form** dan berikan nama **F_IsiBarang**
- Tempatkan 4 komponen **Edit** untuk tempat pengisian data. Kemudian atur nama sesuai dengan kegunaannya.
- Tempatkan suatu tombol kemudian **caption** diisi dengan Simpan, kemudian ubah properti **modalresult** diisi dengan **mrOK**
- Tempatkan suatu tombol kemudian caption diisi dengan **Batal** dan isi properti **modalresultnya** dengan **mrCancel**



3. Klik 2 kalipada component **table** kemudian klik kanan dan pilih **Add All Field**. Dengan melakukan hal tersebut, maka setiap field mempunyai objek Field khusus.

4. Isilah code berikut pada event onclik tombol **Tambah** di **FormF_DataBarang**

```
procedure TF_DataBarang.Bit_TambahClick(Sender: TObject);
begin
  F_IsiBarang.E_NamaBarang.Text:='';
  F_IsiBarang.E_HargaBarang.Text:='';
  F_IsiBarang.E_Stock.Text:='0';
  F_IsiBarang.E_StockMinimal.Text:='';
  F_IsiBarang.ShowModal;
  if (F_IsiBarang.ModalResult=mrOK) then
    begin
      T_Barang.Append;
      T_BarangNama.Value:=F_IsiBarang.E_NamaBarang.Text;
      T_BarangHarga.Value:=StrToFloat(F_IsiBarang.E_HargaBarang.Text);
      T_BarangStock.Value:=StrToInt(F_IsiBarang.E_Stock.Text);
      T_BarangStockMinimal.Value:=StrToInt(F_IsiBarang.E_StockMinimal.Text);
      T_Barang.Post;
    end;
end;
```

5. Untuk proses perubahan data, diperlukan suatu form baru yang digunakan untuk merubah data barang tersebut. Tampilan form perubahan data sama tampilannya dengan form penambahan data, hanya ditambah component **DBEdit** untuk informasi Kode Barang.

- Buat Form baru dengan **File → New → Form** dan berikan nama **F_UbahBarang**
- Tempatkan sebuah komponen **DBEdit** untuk menampung informasi **Kode Barang**
Ubah property dari **DBEdit** tersebut dengan **Datasource** nya = **F_DataBarang.DS_Barang**, kemudian properti **DataField** nya = **Kode**
- Tempatkan 4 komponen **Edit** untuk tempat pengeditan data. Kemudian atur name sesuai dengan kegunaannya.
- Tempatkan suatu tombol kemudian **caption** diisi dengan **Ubah**, kemudian ubah properti **modalresult** diisi dengan **mrOK**
- Tempatkan suatu tombol kemudian **caption** diisi dengan **Batal** dan isi properti **modalresultnya** dengan **mrCancel**

The screenshot shows a Windows-style dialog box titled "Ubah Data Barang". Inside the dialog, there are five text input fields arranged vertically. The first field is labeled "Kode Barang" and contains the number "1". The second field is labeled "Nama Barang" and contains the text "Buku". The third field is labeled "Harga Barang" and contains "10000". The fourth field is labeled "Stock" and contains "100". The fifth field is labeled "Stock Minimal" and contains "5". At the bottom of the dialog, there are two buttons: "Ubah" on the left and "Batal" on the right.

Kemudian, isikan code pada event **onClick Ubah** di **Form F_DataBarang** seperti berikut :

```
procedure TF_DataBarang.Bit_UbahClick(Sender: TObject);
begin
    F_UbahBarang.E_NamaBarang.Text:=F_DataBarang.T_BarangNama.Value;
    F_UbahBarang.E_HargaBarang.Text:=FloatToStr(F_DataBarang.T_BarangHarga.Value);
    F_UbahBarang.E_Stock.Text:=IntToStr(F_DataBarang.T_BarangStock.Value);
    F_UbahBarang.E_StockMinimal.Text:=IntToStr(F_DataBarang.T_BarangStockMinimal.Value);
    F_UbahBarang.ShowModal;
    if (F_UbahBarang.ModalResult=mrOK) then
    begin
        T_Barang.Edit;
        T_BarangNama.Value:=F_UbahBarang.E_NamaBarang.Text;
        T_BarangHarga.Value:=StrToFloat(F_UbahBarang.E_HargaBarang.Text);
        T_BarangStock.Value:=StrToInt(F_UbahBarang.E_Stock.Text);
        T_BarangStockMinimal.Value:=StrToInt(F_UbahBarang.E_StockMinimal.Text);
        T_Barang.Post;
        end;
    end;
```

5. Untuk event onclick pada Tombol **Hapus**, ketiklah code dibawah ini :

```
procedure TF_DataBarang.Bit_HapusClick(Sender: TObject);
begin
    if Application.MessageBox('Apakah anda yakin akan menghapus data ini?', 'Pertanyaan', MB_YESNO or MB_ICONQUESTION)=mrYes Then
    begin
        if T_Barang.IsEmpty=true then
            showmessage('Data Kosong')
        else
            T_Barang.Delete;
        end;
    end;
```

6. Untuk melakukan pencarian data, gunakan perintah berikut

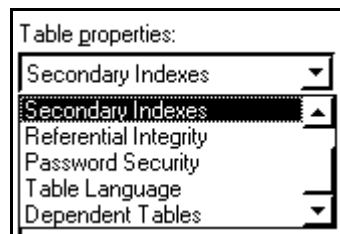
```
procedure TF_DataBarang.BitBtn4Click(Sender: TObject);
begin
    T_Barang.FindNearest([E_Cari.Text]);
end;
```

Pembuatan Index pada Tabel

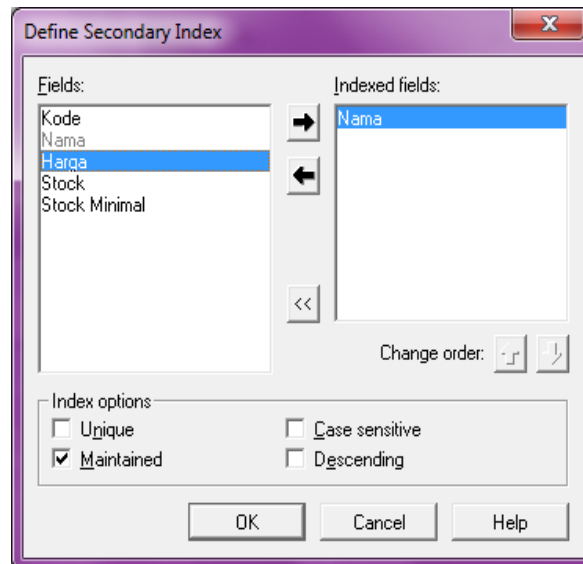
Ketika pembuatan suatu tabel, kita membuat suatu field yang pada fieldnya diberi tanda *. Isi field yang merupakan field kunci tidak boleh ada data yang sama (harus unik). Dengan membuat suatu field sebagai **primary key**, maka setiap pengisian data, maka urutan data akan otomatis terurut. Selain primary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

Cara pembuatan secondary index adalah :

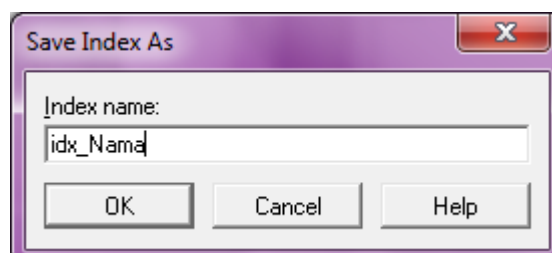
1. Masuk ke **Database Desktop**
2. Pilih menu **Table – Restructure**, kemudian pilih tabel yang akan diubah strukturnya.
3. Pilih **Secondary Indexes** pada field **Table Properties** (lihat gambar).



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 : **idx_Nama**



6. Buat juga index dengan nama **idx_Harga**, yang merupakan dari field harga dengan ketentuan pengurutannya secara menurun (descending)
7. Jika telah selesai, simpan tabel 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 dibawah ini :

Untuk bagian **“Pengurutan Berdasarkan”** digunakan **RadioGroup**, sedangkan event onClick tuliskan perintah berikut :

```

procedure TF_DataBarang.RadioGroup1Click(Sender: TObject);
begin
    if (RadioGroup1.ItemIndex=0) then
        T_Barang.IndexName:=''
    else
        if (RadioGroup1.ItemIndex=1) then
            T_Barang.IndexName:='idx_Nama'
        else
            if (RadioGroup1.ItemIndex=2) then
                T_Barang.IndexName:='idx_Harga';
            end;
end;

```

Untuk tombol **Cari**, tuliskan perintah berikut ini :

```
procedure TF_DataBarang.BitBtn4Click(Sender: TObject);
var
  ketemu:boolean;
begin
  ketemu:=T_Barang.FindKey([E_cari.Text]);
  if (ketemu=false) then
    showmessage('data tidak ditemukan');
end;
```

Untuk tombol **Find Nearest** tuliskan perintah berikut ini :

```
procedure TF_DataBarang.BitBtn1Click(Sender: TObject);
begin
  T_Barang.FindNearest([E_Cari.Text]);
end;
```