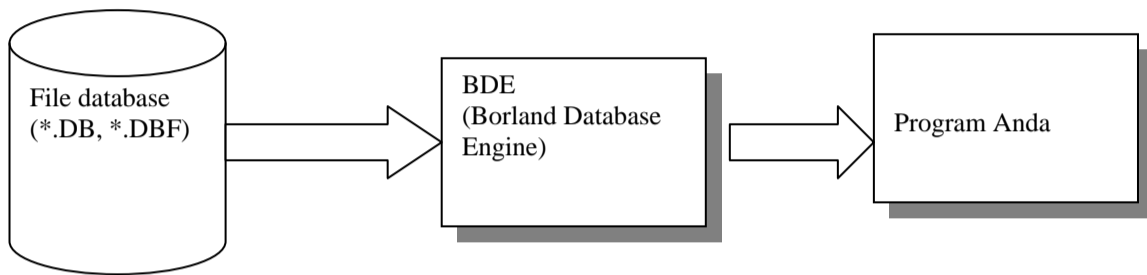


## VIII. PEMRORAMAN DATA BASE (FORNT END)

Delphi menyediakan dua cara untuk membuat program basis data. cara pertama adalah cara konvensional, yaitu dengan menggunakan komponen TTable. cara ini ini biasanya untuk pemrograman basis data pada komputer stand Alone. cara kedua dengan SQL (Structured Query Language) dan biasanya digunakan untuk pemrograman klien/server. SQL akan dibahas pada bab selanjutnya.

Hubungan file data base dengan Delphi dalam pemrograman data abse adalah



Dibutuhkan suatu mesin yang menjadi penghubung antara file database dengan program yang dibuat Delphi. Delphi menyediakan mesin konverter tersebut yang dikenal dengan BDE.

Dalam pemrograman Fornt End ini dibutuhkan dua katagori komponen untuk dapat membuat program database, yaitu:

1. Data Access yang terdiri dari;
  - a. DataSource
  - b. Table
  - c. Query
  - d. StoredProc
  - e. Database
  - f. DLL
2. Data Control yang terdiri dari :
  - a. DBGrid
  - b. DBNavigator
  - c. DBText
  - d. DBEdit
  - e. DBMemo
  - f. DBImage
  - g. DBListBox

- h. DBComboBox
- i. DLL

## Dasar-Dasar Pemrograman Data Base

Untuk membuat aplikasi data base yang sederhana, yang perlu disiapkan adalah komponen Ttable yang akan menghubungkan program Anda dengan file data base yang ada di hard disk. Ada tiga properties yang penting dari Ttable yang perlu diisi, yaitu:

1. DataBaseName, menunjuk tempat dimana file database disimpan, biasanya menunjuk alias dari path database tersebut.
2. TableName, berisi nama-nama table yang ada pada direktori yang ditunjuk.
3. Active, diisi true jika table tersebut ingin dibuka(active) dan diisi false jika table tersebut tetap tertutup (Close).

Setelah Ttable terhubung dengan database, tempelkan TdataSource, untuk menghubungkan Ttable dengan komponen yang menjadi antarmuka database dengan user. Properties yang harus diisi dalam TdataSource adalah DataSet, yaitu diisi dengan nama Ttable yang diinginkan. Setelah itu, tempelkan komponen yang akan dijadikan antarmuka antara database dengan user, seperti DBEdit, DBGrid dan lain-lain.

Properties yang harus diisi dari komponen interfacing itu adalah:

1. DataField, diisi dengan atribut dari suatu table
2. DataSource, diisi dengan nama TdataSource

Table dapat dimanipulasi sesuai dengan kebutuhan user, seperti:

1. Menuju record pertama, dengan prosedur FIRST
2. Menuju record terakhir, dengan prosedur LAST
3. Menuju record berikutnya, dengan prosedur NEXT
4. Menuju record Sebelumnya, dengan prosedur PRIOR
5. Menguji apakah kursor berada pada awal record, dengan prosedur BOF
6. Menguji apakah kursor berada pada akhir record, dengan prosedur EOF
7. Menutup Tabel, dengan prosedur Close atau properties active Ttable diset False
8. Membuka Table, dengan prosedur Open atau properties active Ttable diset True
9. Mengakses Field:

Var s: string;

S:= Table.Fields[0].AsString;, untuk field yang tipe datanya string

Atau

S:= Table.FieldName['Name'].AsString;

## 10. Mengisi Field

```
Table.Fields[0].AsString:='Gurame';
```

Atau

```
Table.FieldByName['Nama'].AsString:='Gurame';
```

## 11. Mengisi Table

Mengisi tabel dengan record baru dapat dilakukan dengan dua cara:

1. dengan perintah Insert, menyisipkan record pada posisi record aktif
2. dengan perintah Append, menambah record pada posisi akhir record

## 12. Menghapus record, dengan prosedur DELETE

## 13. Mengedit Record, dengan prosedur EDIT

## 14. Menyimpan record, dengan prosedur POST

## 15. Mencari data

Mencari data atau record dalam tabel dapat dilakukan dengan perintah:

1. FindKey, mencari dengan record dengan key, yaitu menggunakan index. Jika tabel terindex berdasarkan nomor, anda dapat mencari record dengan nomor '123':

```
Table.FindKey(123);
```

2. FindNearest, hampir sama dengan FindKey, tetapi FindNearest mencari record yang mirip.

```
Table.FindNearest(['Agus']), untuk mencari semua record yang bernama Agus.
```

3. SetKey..GotoKey, hampir sama dengan findKey:

```
Table.SetKey;
```

```
Table.Fields[0].AsInteger:=123;
```

```
Table.GotoKey;
```

4. SetKey..GotoNearest, hampir sama dengan FindNearest

```
Table.SetKey;
```

```
Table.Fields[0].AsString:='Agus';
```

```
Table.GotoNearest;
```

5. ApplyRange

Pencarian dengan ApplyRange akan membatasi record yang ditampilkan sesuai dengan criteria. Dengan perintah ini Anda harus menentukan dahulu batasan record yang akan ditampilkan dengan menggunakan SetRangeStart dan SetRangeEnd.

```
Table.SetRangeStart;
```

```
Table.Fields[0].AsInteger:=100;
```

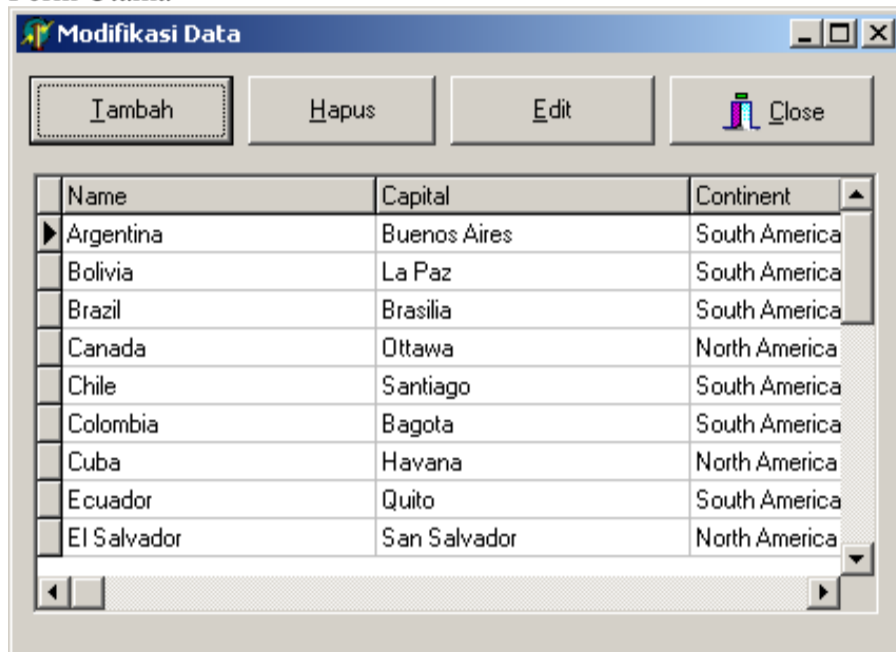
```
Table.SetRangeEnd;  
Table.Fields[0].AsInteger:=2000;  
Table.ApplyRange;
```

6. CancelRange

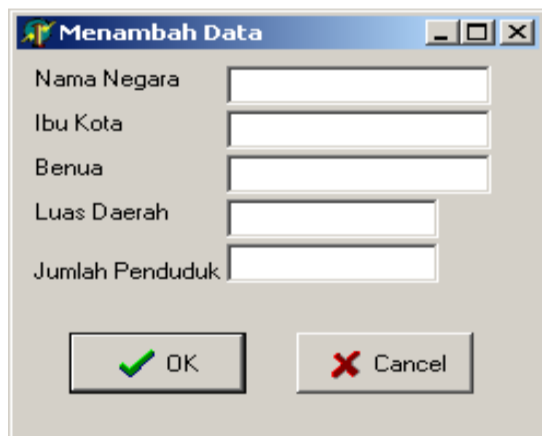
Perintah ini digunakan untuk membatalkan perintah ApplyRange.

### Contoh Program Database Sederhana

Form Utama



Form tambah Data



Keterangan :

Tnegara adalah Nama Ttable dengan :

DatabaseName : DBDEmos

TableName : Country.db

DSNegara adalah Nama DataSource dengan DataSet : Tnegara

### Kode Programnya adalah:

```
unit Unitdbs;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Buttons, StdCtrls, Grids, DBGrids, Db, DBTables;

type
  TFormlatihdatabase = class(TForm)
    DSNegara: TDataSource;
    TNegara: TTable;
    DBGrid1: TDBGrid;
    BtTambah: TButton;
    BtHapus: TButton;
    BtEdit: TButton;
    BitBtn1: TBitBtn;
    procedure BtTambahClick(Sender: TObject);
    procedure BtEditClick(Sender: TObject);
    procedure BtHapusClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Formlatihdatabase: TFormlatihdatabase;

implementation

uses Unittmb;

{$R *.DFM}

procedure TFormlatihdatabase.BtTambahClick(Sender: TObject);
begin
  formtambahdata:=Tformtambahdata.create(self);
  with formtambahdata do
    begin
      ednama.text:="";
      edbukota.text:="";
      edbenua.text:="";
      edluas.text:="";
      edjumlah.text:="";
      caption :='Menambah data';
      activecontrol:=ednama;

      end;
  formtambahdata.showmodal;
  if (formtambahdata.modalresult=mrOK) then
    with TNegara,FormTambahData do
      begin
```

```

        append;
        fieldbyname('Name').asstring:=ednama.text;
        fieldbyname('capital').asstring:=edibukota.text;
        fieldbyname('continent').asstring:=edbenua.text;
        fieldbyname('area').asstring:=edluas.text;
        fieldbyname('population').asstring:=edjumlah.text;
        post;
    end;
    formtambahdata.free;

end;

procedure TFormlatihdatabase.BtEditClick(Sender: TObject);
begin
    formtambahdata:=Tformtambahdata.create(self);
    with TNegara,formtambahdata do
        begin
            ednama.text:=fieldbyname('Name').asstring;
            edibukota.text:=fieldbyname('capital').asstring;
            edbenua.text:=fieldbyname('continent').asstring;
            edluas.text:=fieldbyname('area').asstring;
            edjumlah.text:=fieldbyname('population').asstring;
            caption :='Mengedit data';
            activecontrol:=ednama;
        end;
        formtambahdata.showmodal;
        if (formtambahdata.modalresult=mrOK) then
            with TNegara,FormTambahData do
                begin
                    edit;
                    fieldbyname('Name').asstring:=ednama.text;
                    fieldbyname('capital').asstring:=edibukota.text;
                    fieldbyname('continent').asstring:=edbenua.text;
                    fieldbyname('area').asstring:=edluas.text;
                    fieldbyname('population').asstring:=edjumlah.text;
                    post;
                end;
            formtambahdata.free;
        end;
    end;

procedure TFormlatihdatabase.BtHapusClick(Sender: TObject);
var
    s: array[0..255] of char;
begin
    strPcopy(S,Format('Anda Yakin Negara %s dihapus??',[TNegara.fields[0].asstring]));
    if (application.messageBox(S,'Perhatian',MB_YESNO or
    MB_ICONQUESTION)=IDYES) then
        TNegara.delete;
    end;

end.

```