LAMPIRAN

1. *FormLogin*.Pas

```
unit FormLogin;
interface
uses
 Winapi.Windows, Winapi.Messages,
System.SysUtils, System.Variants,
System.Classes, Vcl.Graphics,
 Vcl.Controls, Vcl.Forms, Vcl.Dialogs,
Vcl.StdCtrls, FireDAC.Stan.Intf,
 FireDAC.Stan.Option,
FireDAC.Stan.Error, FireDAC.UI.Intf,
FireDAC.Phys.Intf,
 FireDAC.Stan.Def, FireDAC.Stan.Pool,
FireDAC.Stan.Async, FireDAC.Phys,
 FireDAC.Phys.SQLite,
FireDAC.Phys.SQLiteDef,
FireDAC.Stan.ExprFuncs,
 FireDAC.Stan.Param, FireDAC.DatS,
FireDAC.DApt.Intf, FireDAC.DApt,
Data.DB,
 FireDAC.Comp.DataSet,
FireDAC.Comp.Client,
FireDAC.VCLUI.Wait;
type
 TFrmLogin = class(TForm)
  EditUsername: TEdit;
  EditPassword: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  btnLogin: TButton;
  FDConnection1: TFDConnection;
  FDQuery1: TFDQuery;
  procedure btnLoginClick(Sender:
TObject);
  procedure FormCreate(Sender: TObject);

private
  { Private declarations }
 public
  { Public declarations }
  LoginStatus : boolean;
 end;
var
 FrmLogin: TFrmLogin;

implementation

{$R *.dfm}
procedure
TFrmLogin.btnLoginClick(Sender:
TObject);
var
 Username, Password: string;
begin
 Username := Trim(EditUsername.Text);
 Password := Trim(EditPassword.Text);
 // Pastikan input tidak kosong
 if (Username = '') or (Password = '') then
 begin
  ShowMessage('Username dan Password
harus diisi!');
  Exit;
 end;
 // Query ke database untuk mengecek
Username dan Password
 FDQuery1.Close;


 FDQuery1.SQL.Text := 'SELECT
COUNT(*) FROM Users WHERE
Username = :uname AND Password =
:pass';

FDQuery1.ParamByName('uname').AsStrin
g := Username;
 FDQuery1.ParamByName('pass').AsString
:= Password;
 FDQuery1.Open;
 if FDQuery1.Fields[0].AsInteger > 0 then
 begin
//  ShowMessage('Login berhasil!');
  LoginStatus := True;
  Close;
 end
 else
 begin
  ShowMessage('Username atau Password
salah!');
  LoginStatus := False;
 end;
end;
```

```
procedure TFrmLogin.FormCreate(Sender: TObject);
begin
end;

end.
```

2. *FormUsers*.pas

```
unit FormUsers;

interface

uses

  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ComCtrls, FireDAC.Stan.Intf,

  FireDAC.Stan.Option, FireDAC.Stan.Error, FireDAC.UI.Intf, FireDAC.Phys.Intf,

  FireDAC.Stan.Def, FireDAC.Stan.Pool, FireDAC.Stan.Async, FireDAC.Phys,

  FireDAC.Phys.SQLite, FireDAC.Phys.SQLiteDef, FireDAC.Stan.ExprFuncs,

  FireDAC.Stan.Param, FireDAC.DatS, FireDAC.DApt.Intf, FireDAC.DApt,

  Vcl.Bind.Grid, System.Rtti, System.Bindings.Outputs, Vcl.Bind.Editors,

  Data.Bind.EngExt, Vcl.Bind.DBEngExt, Vcl.ExtCtrls, Data.Bind.Components,

  Data.Bind.Grid, Vcl.Grids, Data.DB, FireDAC.Comp.DataSet, FireDAC.Comp.Client,

  Data.Bind.DBScope, FireDAC.VCLUI.Wait, FireDAC.Comp.UI, Vcl.StdCtrls,

  Vcl.DBGrids;

type

  TFrmUsers = class(TForm)

    PageControl1: TPageControl;

    TabSheet1: TTabSheet;

    Panel1: TPanel;

    editSearch: TEdit;

    btnSearch: TButton;

    FDConnection1: TFDConnection;

    FDQuery1: TFDQuery;

    DataSource1: TDataSource;

    DBGrid1: TDBGrid;

    btnSimpan: TButton;

    Label1: TLabel;

    Label2: TLabel;

    btnEdit: TButton;

    btnDelete: TButton;

    btnClear: TButton;

    EditUsername: TEdit;

    EditPassword: TEdit;

    Label3: TLabel;

    EditFullname: TEdit;

    Panel2: TPanel;

    procedure btnSearchClick(Sender: TObject);

    procedure btnSimpanClick(Sender: TObject);

    procedure DBGrid1CellClick(Column: TColumn);

    procedure btnEditClick(Sender: TObject);

    procedure btnDeleteClick(Sender: TObject);

    procedure btnClearClick(Sender: TObject);

  private

    { Private declarations }

  public
```

```
  { Public declarations }
 end;


var
  FrmUsers: TFrmUsers;
implementation
{$R *.dfm}
procedure TFrmUsers.btnSearchClick(Sender: TObject);
begin
  FDQuery1.SQL.Clear;
  FDQuery1.Open('SELECT * FROM users WHERE username LIKE :username', ['%'+ editSearch.Text +'%']);
  FDQuery1.Active := True;
end;
procedure TFrmUsers.btnSimpanClick(Sender: TObject);
begin
  FDQuery1.Append;

FDQuery1.FieldByName('fullname').AsString := EditFullname.Text;

FDQuery1.FieldByName('username').AsString := EditUsername.Text;

FDQuery1.FieldByName('Password').AsString := EditPassword.Text;
  FDQuery1.Post;
end;
procedure  TFrmUsers.btnEditClick(Sender: TObject);
begin
  FDQuery1.Edit;
FDQuery1.FieldByName('fullname').AsString := EditFullname.Text;
FDQuery1.FieldByName('username').AsString := EditUsername.Text;
FDQuery1.FieldByName('Password').AsString := EditPassword.Text;
  FDQuery1.Post;
end;
procedure TFrmUsers.btnClearClick(Sender: TObject);
begin
  EditFullname.Text := '';
  EditUsername.Text := '';
  EditPassword.Text := '';
end;
procedure TFrmUsers.btnDeleteClick(Sender: TObject);
begin
  if MessageDlg('Yakin akan menghapus data ini?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
  FDQuery1.Delete;
end;
procedure TFrmUsers.DBGrid1CellClick(Column: TColumn);
begin
  EditFullname.Text := FDQuery1.FieldByName('fullname').AsString;
  EditUsername.Text := FDQuery1.FieldByName('username').AsString;
  EditPassword.Text := FDQuery1.FieldByName('Password').AsString;
end;
```

```
end.
```

3. *FormHome*.pas

```
unit FormHome;

interface

uses

  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls,

  Vcl.Imaging.jpeg, Vcl.ExtCtrls;

type

  TFrmHome = class(TForm)

    PageControl1: TPageControl;

    HOME: TTabSheet;

    Image1: TImage;

  private

    { Private declarations }

  public

    { Public declarations }

  end;

var

  FrmHome: TFrmHome;

implementation

{$R *.dfm}

end.
```

4. *Form*Utama.pas

```
unit FormUtama;

interface

uses

  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, FormLogin, Vcl.StdCtrls, Vcl.ExtCtrls,

  Vcl.Imaging.pngimage;

type

  TFrmUtama = class(TForm)

    Panel1: TPanel;

    Panel2: TPanel;

    imgHome: TImage;

    imgRAB: TImage;

    imgUsers: TImage;

    imgLogout: TImage;

    Label1: TLabel;

    Label2: TLabel;

    Label3: TLabel;

    Label4: TLabel;

    imgBarang: TImage;

    Label5: TLabel;

    procedure FormShow(Sender: TObject);

    procedure imgLogoutClick(Sender: TObject);

    procedure imgRABClick(Sender: TObject);

    procedure imgHomeClick(Sender: TObject);

    procedure imgUsersClick(Sender: TObject);

    procedure imgBarangClick(Sender: TObject);

  private

    { Private declarations }

  public

    { Public declarations }

  end;

var
```

```
  FrmUtama: TFrmUtama;
  Login Status: boolean;
implementation
{$R *.dfm}
uses FormHome, FormRAB, FormUsers, FormBarang;
procedure TFrmUtama.FormShow(Sender: TObject);
begin
  frmLogin := TFrmLogin.Create(self);
  frmLogin.ShowModal;
  LoginStatus := frmLogin.LoginStatus;
  if LoginStatus = False then
  begin
    Application.Terminate;
  end;
end;
procedure TFrmUtama.imgHomeClick(Sender: TObject);
begin
  TFrmHome.Create(NIL);
end;
procedure TFrmUtama.imgBarangClick(Sender: TObject);
begin
  TFrmBarang.Create(NIL);
end;
procedure TFrmUtama.imgRABClick(Sender: TObject);
begin
  TFrmRAB.Create(NIL);
end;
procedure TFrmUtama.imgUsersClick(Sender: TObject);
begin
  TFrmUsers.Create(NIL);
end;
procedure TFrmUtama.imgLogoutClick(Sender: TObject);
begin
  if MessageDlg('Yakin ingin logout?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
  Application.Terminate;
end;
end.
```

5. *Form*Barang.pas

```
unit FormBarang;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.ComCtrls, Data.DB, Vcl.StdCtrls,
  Vcl.Grids, Vcl.DBGrids, Vcl.ExtCtrls, FireDAC.Stan.Intf, FireDAC.Stan.Option,
  FireDAC.Stan.Param, FireDAC.Stan.Error, FireDAC.DatS, FireDAC.Phys.Intf,
  FireDAC.DApt.Intf, FireDAC.Stan.Async, FireDAC.DApt, FireDAC.UI.Intf,
  FireDAC.Stan.Def, FireDAC.Stan.Pool, FireDAC.Phys, FireDAC.Phys.SQLite,
  FireDAC.Phys.SQLiteDef, FireDAC.Stan.ExprFuncs, FireDAC.Comp.Client,
  FireDAC.Comp.DataSet, FireDAC.VCLUI.Wait;
```

```
type
  TFrmBarang = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Panel2: TPanel;
    EditNamaBarang: TEdit;
    EditSatuan: TEdit;
    EditHarga: TEdit;
    EditPajak: TEdit;
    btnSimpan: TButton;
    btnHapus: TButton;
    FDQuery1: TFDQuery;
    FDConnection1: TFDConnection;
    DataSource1: TDataSource;
    EditIDBarang: TEdit;
    DBGrid1: TDBGrid;
    procedure btnSimpanClick(Sender: TObject);
    procedure btnHapusClick(Sender: TObject);
    procedure LoadData;
    procedure DBGrid1CellClick(Column: TColumn);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FrmBarang: TFrmBarang;

implementation

{$R *.dfm}

procedure TFrmBarang.btnSimpanClick(Sender: TObject);
begin
  if Trim(EditNamaBarang.Text) = '' then
  begin
    ShowMessage('Nama barang harus diisi!');
    Exit;
  end;
  if Trim(EditSatuan.Text) = '' then
  begin
    ShowMessage('Satuan harus diisi!');
    Exit;
  end;
  if Trim(EditHarga.Text) = '' then
  begin
    ShowMessage('Harga harus diisi!');
    Exit;
  end;
  if Trim(EditPajak.Text) = '' then
  begin
    ShowMessage('Pajak harus diisi!');
    Exit;
  end;
  FDQuery1.Close;
  FDQuery1.SQL.Text :=
    'INSERT INTO tbl_barang (nama_barang, satuan, harga, pajak) ' +
```

```
  'VALUES (:nama_barang, :satuan, :harga, :pajak)';

FDQuery1.ParamByName('nama_barang').AsString := EditNamaBarang.Text;
FDQuery1.ParamByName('satuan').AsString := EditSatuan.Text;
FDQuery1.ParamByName('harga').AsInteger := StrToIntDef(EditHarga.Text, 0);
FDQuery1.ParamByName('pajak').AsInteger := StrToIntDef(EditPajak.Text, 0);
 try
  FDQuery1.ExecSQL;
  ShowMessage('Barang berhasil disimpan.');
  EditNamaBarang.Clear;
  EditSatuan.Clear;
  EditHarga.Clear;
  EditPajak.Clear;
  LoadData;
 except
  on E: Exception do
   ShowMessage('Gagal menyimpan barang: ' + E.Message);
 end;
end;
procedure TFrmBarang.DBGrid1CellClick(Column: TColumn);
begin
 if not FDQuery1.IsEmpty then
 begin
  EditIDBarang.Text := FDQuery1.FieldByName('id_barang').AsString;
  EditNamaBarang.Text := FDQuery1.FieldByName('nama_barang').AsString;
  EditSatuan.Text := FDQuery1.FieldByName('satuan').AsString;
  EditHarga.Text := FDQuery1.FieldByName('harga').AsString;
  EditPajak.Text := FDQuery1.FieldByName('pajak').AsString;
 end;
end;
procedure TFrmBarang.btnHapusClick(Sender: TObject);
begin
 if Trim(EditIDBarang.Text) = '' then
 begin
  ShowMessage('Pilih data terlebih dahulu!');
  Exit;
 end;
 if MessageDlg('Apakah Anda yakin ingin menghapus barang ini?', mtConfirmation, [mbYes, mbNo], 0) = mrNo then
  Exit;
 FDQuery1.Close;
 FDQuery1.SQL.Text := 'DELETE FROM tbl_barang WHERE id_barang = :id_barang';
 FDQuery1.ParamByName('id_barang').AsInteger := StrToIntDef(EditIDBarang.Text, 0);
 try
  FDQuery1.ExecSQL;
  ShowMessage('Barang berhasil dihapus.');
  EditIDBarang.Clear;
  EditNamaBarang.Clear;
  EditSatuan.Clear;
```

```
    EditHarga.Clear;
    EditPajak.Clear;
    LoadData;
  except
    on E: Exception do
      ShowMessage('Gagal menghapus barang: ' + E.Message);
  end;
end;

procedure TFrmBarang.LoadData;
begin
  FDQuery1.Close;
  FDQuery1.SQL.Text := 'SELECT id_barang, nama_barang, satuan, harga, pajak FROM tbl_barang';
  FDQuery1.Open;
end;

end.
```

6. *Form*Rab.pas

```
TFrmRAB = class(TForm)
    PageControl1: TPageControl;
    DATA_RAB: TTabSheet;
    FDConnection1: TFDConnection;
    FDQueryUraian: TFDQuery;
    DataSourceUraian: TDataSource;
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    FDQueryRab: TFDQuery;
    DataSourceRab: TDataSource;
    DATA_URAIAN: TTabSheet;
    Panel2: TPanel;
    DBGrid2: TDBGrid;
    EditNamaRab: TEdit;
    EditKode: TEdit;
    btnSimpan: TButton;
    btnEdit: TButton;
    btnHapus: TButton;
    btnClear: TButton;
    Label1: TLabel;
    Label2: TLabel;
    ComboBox1: TComboBox;
    EditRABUraianKode: TEdit;
    GroupBox1: TGroupBox;
    Label3: TLabel;
    Label4: TLabel;
    btnKelolaUraian: TButton;
    btnRefresh: TButton;
    btnCetak: TButton;
    PrintDialog1: TPrintDialog;
    frxReport1: TfrxReport;
    frxDBDataset1: TfrxDBDataset;
    EditTotal: TEdit;
    FDQueryPengaturan: TFDQuery;
    GroupBox2: TGroupBox;
    Label5: TLabel;
    EditNamaDirektur: TEdit;
    ButtonSimpanPengaturanUraian: TButton;
    procedure btnSimpanClick(Sender: TObject);
    procedure btnEditClick(Sender: TObject);
    procedure btnHapusClick(Sender: TObject);
    procedure btnClearClick(Sender: TObject);
    procedure DBGrid2CellClick(Column: TColumn);
    procedure DATA_URAIANShow(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure DATA_RABShow(Sender: TObject);
    procedure btnKelolaUraianClick(Sender: TObject);
    procedure btnRefreshClick(Sender: TObject);
    procedure DBGrid1DblClick(Sender: TObject);
    procedure btnCetakClick(Sender: TObject);
    procedure frxUserDataSet1CheckEOF(Sender: TObject; var Eof: Boolean);
    procedure ButtonSimpanPengaturanUraianClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure HitungTotal;
```

```
  private
    { Private declarations }
   SelectedID: Integer;
  public
    { Public declarations }
  end;

var
  FrmRAB: TFrmRAB;

implementation

{$R *.dfm}

uses FormRAB_Kelola_Uraian,
FormRAB_Kelola_Barang;

procedure
TFrmRAB.FormShow(Sender:
TObject);
begin
  FDQueryPengaturan.Close;
  FDQueryPengaturan.SQL.Text :=
'SELECT nama_direktur FROM
tbl_pengaturan WHERE id_pengaturan
= 1';
  FDQueryPengaturan.Open;

  if not FDQueryPengaturan.IsEmpty
then
     EditNamaDirektur.Text :=
FDQueryPengaturan.FieldByName('na
ma_direktur').AsString;
end;

// TAB DATA RAB //
procedure
TFrmRAB.DATA_RABShow(Sender:
TObject);
begin
  FDQueryRab.Close;
  FDQueryRab.SQL.Text := 'SELECT *
FROM tbl_rab';
  FDQueryRab.Open;
end;
procedure
TFrmRAB.btnSimpanClick(Sender:
TObject);
begin
  FDQueryRab.Append;
FDQueryRab.FieldByName('nama_rab')
.AsString := EditNamaRab.Text;

FDQueryRab.FieldByName('kode').AsS
tring := EditKode.Text;
  FDQueryRab.Post;
end;

procedure
TFrmRAB.ButtonSimpanPengaturanUr
aianClick(Sender: TObject);
begin
  if Trim(EditNamaDirektur.Text) = ''
then
  begin
    ShowMessage('Nama direktur tidak
boleh kosong.');
    Exit;
  end;

  FDQueryPengaturan.Close;
  FDQueryPengaturan.SQL.Text :=
    'UPDATE tbl_pengaturan ' +
    'SET nama_direktur = :nama ' +
    'WHERE id_pengaturan = 1'; // ganti
sesuai primary key yang kamu pakai

FDQueryPengaturan.ParamByName('na
ma').AsString :=
EditNamaDirektur.Text;
  FDQueryPengaturan.ExecSQL;

  ShowMessage('Nama direktur berhasil
diperbarui.');
end;

procedure
TFrmRAB.btnEditClick(Sender:
TObject);
begin
  FDQueryRab.Edit;

FDQueryRab.FieldByName('nama_rab')
.AsString := EditNamaRab.Text;

FDQueryRab.FieldByName('kode').AsS
tring := EditKode.Text;
  FDQueryRab.Post;
end;
```

```
procedure
TFrmRAB.btnHapusClick(Sender:
TObject);
begin
  if MessageDlg('Yakin akan menghapus
data ini?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes then
  FDQueryRab.Delete;
end;

procedure
TFrmRAB.btnClearClick(Sender:
TObject);
begin
  EditNamaRab.Text := '';
  EditKode.Text := '';
end;

procedure
TFrmRAB.DBGrid2CellClick(Column:
TColumn);
begin
  EditNamaRab.Text :=
FDQueryRab.FieldByName('nama_rab')
.AsString;
  EditKode.Text :=
FDQueryRab.FieldByName('kode').AsS
tring;
end;

procedure
TFrmRAB.frxUserDataSet1CheckEOF(
Sender: TObject; var Eof: Boolean);
begin

end;

// TAB DATA URAIAN //
procedure TFrmRAB.HitungTotal;
var
  total, nilai: Double;
  i: Integer;
begin
  total := 0;

  for i := 0 to
DBGrid1.DataSource.DataSet.RecordC
ount - 1 do
  begin
    DBGrid1.DataSource.DataSet.RecNo
:= i + 1;
    nilai :=
StrToFloatDef(DBGrid1.Fields[7].AsSt
ring, 0); // kolom ke-8 (index 7)
    total := total + nilai;
  end;
  EditTotal.Text := FloatToStr(total);
end;

procedure
TFrmRAB.DATA_URAIANShow(Sen
der: TObject);
begin
  ComboBox1.Items.Clear;

  FDQueryRab.Open;
  FDQueryRab.First;
  while not FDQueryRab.Eof do
  begin

ComboBox1.Items.AddObject(FDQuer
yRab.FieldByName('nama_rab').AsStrin
g,
TObject(FDQueryRab.FieldByName('id
_rab').AsInteger));
    FDQueryRab.Next;
  end;
end;

procedure
TFrmRAB.ComboBox1Change(Sender:
TObject);
begin
  if ComboBox1.ItemIndex = -1 then
Exit;

  SelectedID :=
Integer(ComboBox1.Items.Objects[Co
mboBox1.ItemIndex]);

  FDQueryRab.Close;
  FDQueryRab.SQL.Text := 'SELECT *
FROM tbl_rab WHERE id_rab = :id';

FDQueryRab.ParamByName('id').AsInt
eger := SelectedID;
  FDQueryRab.Open;

  FDQueryUraian.Close;
  FDQueryUraian.SQL.Text :=
   'WITH DataUraian AS ( ' +
   '    SELECT ' +
```

```
    '       NULL AS id_detail, ' +
    '       u.id_uraian, ' +
    '       u.uraian, ' +
    '       NULL AS nama_barang, ' +
    '       NULL AS volume, ' +
    '       NULL AS satuan, ' +
    '       NULL AS harga, ' +
    '       NULL AS total_harga, ' +
    '       NULL AS pajak, ' +
    '       NULL AS total_sebelum_pajak,
' +
    '       NULL AS total_setelah_pajak, '
+
    '       "Tambah Barang" AS aksi, ' +
    '       0 AS sort_order ' +
    '   FROM tbl_uraian u ' +
    '   WHERE u.id_rab = :id_rab ' +
    '   UNION ALL ' +
    '   SELECT ' +
    '       d.id_detail, ' +
    '       u.id_uraian, ' +
    '       NULL AS uraian, ' +
    '       b.nama_barang, ' +
    '       d.volume, ' +
    '       b.satuan, ' +
    '       b.harga, ' +
    '       d.total_harga, ' +
    '       b.pajak, ' +
    '                       (d.total_harga)  AS
total_sebelum_pajak, ' +
    '       (d.total_harga + (d.total_harga *
b.pajak / 100)) AS total_setelah_pajak, '
+
    '       "Hapus Barang" AS aksi, ' +
    '       1 AS sort_order ' +
    '   FROM tbl_uraian u ' +
    '    LEFT JOIN tbl_rab_detail d ON
u.id_uraian = d.id_uraian ' +
    '       LEFT JOIN tbl_barang b ON
d.id_barang = b.id_barang ' +
    '   WHERE u.id_rab = :id_rab ' +
    ') ' +
    'SELECT * FROM DataUraian ' +
    'ORDER BY id_uraian, sort_order;';

  // **Tentukan tipe data parameter
sebelum diisi**

FDQueryUraian.ParamByName('id_rab'
).DataType := ftInteger;

FDQueryUraian.ParamByName('id_rab'
).AsInteger := SelectedID;  // Set nilai
parameter
  FDQueryUraian.Open;

  if not FDQueryRab.IsEmpty then
    begin
      EditRABUraianKode.Text :=
FDQueryRab.FieldByName('kode').AsS
tring;
    end
  else
    begin
      EditRABUraianKode.Clear;
    end;

  HitungTotal;
end;

procedure
TFrmRAB.btnKelolaUraianClick(Sende
r: TObject);
begin
  if SelectedID = 0 then
  begin
    ShowMessage('Silakan pilih RAB
terlebih dahulu.');
    Exit;
  end;

  FormKelolaUraian :=
TFormKelolaUraian.Create(Self);
  try

FormKelolaUraian.LoadData(SelectedI
D);
    FormKelolaUraian.ShowModal;
  finally
    FormKelolaUraian.Free;
  end;
end;

procedure
TFrmRAB.btnRefreshClick(Sender:
TObject);
begin
  FDQueryUraian.Close;
  FDQueryUraian.SQL.Text :=
    'WITH DataUraian AS ( ' +
    '   SELECT ' +
```

```
    '       NULL AS id_detail, ' +
    '       u.id_uraian, ' +
    '       u.uraian, ' +
    '       NULL AS nama_barang, ' +
    '       NULL AS volume, ' +
    '       NULL AS satuan, ' +
    '       NULL AS harga, ' +
    '       NULL AS total_harga, ' +
    '       NULL AS pajak, ' +
    '       NULL AS total_sebelum_pajak,
' +
    '       NULL AS total_setelah_pajak, '
+
    '       "Tambah Barang" AS aksi, ' +
    '       0 AS sort_order ' +
    '    FROM tbl_uraian u ' +
    '    WHERE u.id_rab = :id_rab ' +
    '    UNION ALL ' +
    '    SELECT ' +
    '       d.id_detail, ' +
    '       u.id_uraian, ' +
    '       NULL AS uraian, ' +
    '       b.nama_barang, ' +
    '       d.volume, ' +
    '       b.satuan, ' +
    '       b.harga, ' +
    '       d.total_harga, ' +
    '       b.pajak, ' +
    '                       (d.total_harga)  AS
total_sebelum_pajak, ' +
    '       (d.total_harga + (d.total_harga *
b.pajak / 100)) AS total_setelah_pajak, '
+
    '       "Hapus Barang" AS aksi, ' +
    '       1 AS sort_order ' +
    '    FROM tbl_uraian u ' +
    '     LEFT JOIN tbl_rab_detail d ON
u.id_uraian = d.id_uraian ' +
    '       LEFT JOIN tbl_barang b ON
d.id_barang = b.id_barang ' +
    '    WHERE u.id_rab = :id_rab ' +
    ') ' +
    'SELECT * FROM DataUraian ' +
    'ORDER BY id_uraian, sort_order;';

  // **Tentukan tipe data parameter
sebelum diisi**

FDQueryUraian.ParamByName('id_rab'
).DataType := ftInteger;

FDQueryUraian.ParamByName('id_rab'
).AsInteger := SelectedID;  // Set nilai
parameter
  FDQueryUraian.Open;

  HitungTotal;
end;

procedure
TFrmRAB.DBGrid1DblClick(Sender:
TObject);
var
  idDetail: Integer;
  idUraian: Integer;
  aksi: string;
begin
  // Pastikan ada data yang dipilih
  if FDQueryUraian.IsEmpty then Exit;

  // Ambil ID uraian dan aksi dari kolom
yang diklik
  idUraian                          :=
FDQueryUraian.FieldByName('id_uraia
n').AsInteger;
  aksi                              :=
FDQueryUraian.FieldByName('aksi').A
sString;

  if aksi = 'Tambah Barang' then
  begin
    FormKelolaBarang                :=
TFormKelolaBarang.Create(Self);
    try

FormKelolaBarang.LoadData(idUraian)
;
      FormKelolaBarang.ShowModal;
    finally
      FormKelolaUraian.Free;
    end;
  end
  else if aksi = 'Hapus Barang' then
  begin
    if      MessageDlg('Yakin      ingin
menghapus      barang      ini?',
mtConfirmation, [mbYes, mbNo], 0) =
mrYes then
    begin
      // Ambil ID detail dari dataset
```

```
    idDetail := FDQueryUraian.FieldByName('id_detail').AsInteger;

    if idDetail > 0 then
    begin
      // Jalankan query DELETE pada tbl_rab_detail berdasarkan id_detail
      FDQueryRab.Close;
      FDQueryRab.SQL.Text := 'DELETE FROM tbl_rab_detail WHERE id_detail = :id_detail';

FDQueryRab.ParamByName('id_detail').AsInteger := idDetail;
      FDQueryRab.ExecSQL;

      // Refresh data setelah hapus
      btnRefresh.Click;

      ShowMessage('Barang berhasil dihapus.');
    end
    else
      ShowMessage('ID Detail tidak ditemukan, data tidak dapat dihapus.');
   end;
  end;
end;

procedure TFrmRAB.btnCetakClick(Sender: TObject);
var
  KodeRab, NamaRab, TotalRab, NamaDirektur: string;
begin
  // Ambil data dari komponen di form
  KodeRab := EditRABUraianKode.Text;
  NamaRab := ComboBox1.Text;
  TotalRab := EditTotal.Text;
  NamaDirektur := EditNamaDirektur.Text;

  // Set nilai variabel report di FastReport
  frxReport1.Variables['KodeRab'] := QuotedStr(KodeRab); // Mengirim KodeRab
  frxReport1.Variables['NamaRab'] := QuotedStr(NamaRab); // Mengirim NamaRab
  frxReport1.Variables['TotalRab'] := QuotedStr(TotalRab); // Mengirim TotalRab
  frxReport1.Variables['NamaDirektur'] := QuotedStr(NamaDirektur); // Mengirim NamaDirektur

  // Tampilkan laporan
  frxReport1.ShowReport();
end;

end.
```

7. *Form*Rab_Kelola_Uraian.pas

```
unit FormRAB_Kelola_Uraian;

interface

uses

  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, FireDAC.Stan.Intf, FireDAC.Stan.Option,

  FireDAC.Stan.Param, FireDAC.Stan.Error, FireDAC.DatS, FireDAC.Phys.Intf,

  FireDAC.DApt.Intf, FireDAC.Stan.Async, FireDAC.DApt, FireDAC.UI.Intf,

  FireDAC.Stan.Def, FireDAC.Stan.Pool, FireDAC.Phys, FireDAC.Phys.SQLite,

  FireDAC.Phys.SQLiteDef, FireDAC.Stan.ExprFuncs, Data.DB, Vcl.StdCtrls,

  Vcl.Grids, Vcl.DBGrids, Vcl.ExtCtrls, FireDAC.Comp.Client,

  FireDAC.Comp.DataSet, FireDAC.VCLUI.Wait;

type

  TFormKelolaUraian = class(TForm)

    DataSource1: TDataSource;

    FDQuery1: TFDQuery;
```

```
    FDConnection1: TFDConnection;
    Panel1: TPanel;
    DBGrid1: TDBGrid;
    EditUraian: TEdit;
    Label1: TLabel;
    btnSimpan: TButton;
    btnHapus: TButton;
    procedure btnSimpanClick(Sender: TObject);
    procedure btnHapusClick(Sender: TObject);
  private
    { Private declarations }
    FSelectedID: Integer;
  public
    { Public declarations }
    procedure LoadData(ASelectedID: Integer);
  end;
var
  FormKelolaUraian: TFormKelolaUraian;
  SelectedID: Integer;
implementation

{$R *.dfm}

procedure TFormKelolaUraian.btnSimpanClick(Sender: TObject);
begin
  if Trim(EditUraian.Text) = '' then
  begin
    ShowMessage('Uraian tidak boleh kosong!');
    Exit;
  end;
  try
    FDQuery1.Close;
    FDQuery1.SQL.Text := 'INSERT INTO tbl_uraian (id_rab, uraian) VALUES (:id_rab, :uraian)';
FDQuery1.ParamByName('id_rab').AsInteger := FSelectedID;
FDQuery1.ParamByName('uraian').AsString := Trim(EditUraian.Text);
    FDQuery1.ExecSQL;
    ShowMessage('Data berhasil disimpan!');
    FDQuery1.Close;
    FDQuery1.SQL.Text := 'SELECT * FROM tbl_uraian WHERE id_rab = :id';
FDQuery1.ParamByName('id').AsInteger := FSelectedID;
    FDQuery1.Open; // Refresh data setelah simpan
  except
    on E: Exception do
      ShowMessage('Error: ' + E.Message);
  end;
end;
procedure TFormKelolaUraian.btnHapusClick(Sender: TObject);
begin
  if FDQuery1.IsEmpty then
  begin
    ShowMessage('Tidak ada data yang dipilih untuk dihapus!');
    Exit;
  end;
  if MessageDlg('Apakah Anda yakin ingin menghapus uraian ini?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
```

```
 begin
  try
   FDQuery1.Delete; // Menghapus baris yang dipilih dalam dataset
   ShowMessage('Data berhasil dihapus!');
  except
   on E: Exception do
    ShowMessage('Error saat menghapus data: ' + E.Message);
  end;
 end;
end;


procedure TFormKelolaUraian.LoadData(ASelectedID : Integer);
begin
 FSelectedID := ASelectedID;
 if FSelectedID = 0 then
 begin
  ShowMessage('ID RAB tidak valid!');
  Exit;
 end;
 if not FDConnection1.Connected then
  FDConnection1.Connected := True;
 FDQuery1.Close;
 FDQuery1.SQL.Text := 'SELECT * FROM tbl_uraian WHERE id_rab = :id';
FDQuery1.ParamByName('id').AsInteger := FSelectedID;
 FDQuery1.Open;
end;
end.
```

8. *Form*Rab_Kelola_Barang.pas

```
unit FormRAB_Kelola_Barang;
interface
uses
 Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,
 Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Vcl.StdCtrls, Vcl.Grids,
 Vcl.DBGrids, Vcl.ExtCtrls, FireDAC.Stan.Intf, FireDAC.Stan.Option,
 FireDAC.Stan.Error, FireDAC.UI.Intf, FireDAC.Phys.Intf, FireDAC.Stan.Def,
 FireDAC.Stan.Pool, FireDAC.Stan.Async, FireDAC.Phys, FireDAC.Phys.SQLite,
 FireDAC.Phys.SQLiteDef, FireDAC.Stan.ExprFuncs, FireDAC.Stan.Param,
 FireDAC.DatS, FireDAC.DApt.Intf, FireDAC.DApt, FireDAC.Comp.DataSet,
 FireDAC.Comp.Client, FireDAC.VCLUI.Wait;
type
 TFormKelolaBarang = class(TForm)
  Panel1: TPanel;
  DBGrid1: TDBGrid;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  FDConnection1: TFDConnection;
  FDQuery1: TFDQuery;
  DataSource1: TDataSource;
  ComboBoxBarang: TComboBox;
  EditVolume: TEdit;
  EditSatuan: TEdit;
```

```
    Label4: TLabel;
    EditHarga: TEdit;
    Label5: TLabel;
    EditTotalHarga: TEdit;
    EditIDUraian: TEdit;
    btnSimpan: TButton;
    btnHapus: TButton;
    FDQueryBarang: TFDQuery;
    procedure FormShow(Sender: TObject);
    procedure
ComboBoxBarangChange(Sender: TObject);
    procedure HitungTotal;
    procedure EditVolumeKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure btnSimpanClick(Sender:
TObject);
    procedure btnHapusClick(Sender:
TObject);
  private
    { Private declarations }
    FSelectedID: Integer;
  public
    { Public declarations }
    procedure LoadData(ASelectedID:
Integer);
  end;
var
  FormKelolaBarang: TFormKelolaBarang;
  SelectedID: Integer;
implementation
{$R *.dfm}
procedure
TFormKelolaBarang.btnSimpanClick(Sende
r: TObject);
var
  BarangID, IDUraian: Integer;
  Volume, Harga, Total: Double;
  FS: TFormatSettings;
begin
  // Pastikan Delphi menggunakan titik (.)
sebagai pemisah desimal
  FS := TFormatSettings.Create;
  FS.DecimalSeparator := '.';
  // Validasi pemilihan barang
  if ComboBoxBarang.ItemIndex = -1 then
  begin
    ShowMessage('Silakan pilih barang
terlebih dahulu.');
    Exit;
  end;
  // Ambil nilai barang dari ComboBox
  BarangID :=
Integer(ComboBoxBarang.Items.Objects[Co
mboBoxBarang.ItemIndex]);
  // Validasi input angka
  if not TryStrToFloat(EditVolume.Text,
Volume, FS) or (Volume <= 0) then
  begin
    ShowMessage('Masukkan volume yang
valid.');
    Exit;
  end;
  if not TryStrToFloat(EditHarga.Text,
Harga, FS) or (Harga <= 0) then
  begin
```

```
  ShowMessage('Masukkan harga yang valid.');
  Exit;
 end;
 // Hitung total harga
 Total := Volume * Harga;
 IDUraian := StrToIntDef(EditIDUraian.Text, 0);
 // Gunakan transaksi untuk menghindari kesalahan saat penyimpanan
 FDConnection1.StartTransaction;
 try
  FDQuery1.Close;
  FDQuery1.SQL.Text := 'INSERT INTO tbl_rab_detail (id_uraian, id_barang, volume, total_harga) ' +
                       'VALUES (:id_uraian, :id_barang, :volume, :total_harga)';

FDQuery1.ParamByName('id_uraian').AsInteger := IDUraian;

FDQuery1.ParamByName('id_barang').AsInteger := BarangID;

FDQuery1.ParamByName('volume').AsFloat := Volume;

FDQuery1.ParamByName('total_harga').AsFloat := Total;
  FDQuery1.ExecSQL;
  // Commit transaksi
  FDConnection1.Commit;
  ShowMessage('Data berhasil disimpan.');
  LoadData(IDUraian);
 except
  on E: Exception do
  begin
   FDConnection1.Rollback;
   ShowMessage('Terjadi kesalahan saat menyimpan data: ' + E.Message);
  end;
 end;
end;


procedure TFormKelolaBarang.btnHapusClick(Sender: TObject);
var
 IDDetail: Integer;
begin
 if DBGrid1.DataSource.DataSet.IsEmpty then
 begin
  ShowMessage('Tidak ada data yang dipilih.');
  Exit;
 end;
 IDDetail := DBGrid1.DataSource.DataSet.FieldByName('id_detail').AsInteger;
 if MessageDlg('Apakah Anda yakin ingin menghapus data ini?', mtConfirmation, [mbYes, mbNo], 0) = mrYes then
 begin
  FDQuery1.Close;
  FDQuery1.SQL.Text := 'DELETE FROM tbl_rab_detail WHERE id_detail = :id_detail';
FDQuery1.ParamByName('id_detail').AsInteger := IDDetail;
  FDQuery1.ExecSQL;
  ShowMessage('Data berhasil dihapus.');
```

```
    LoadData(FSelectedID); // Refresh data
  end;
end;
procedure TFormKelolaBarang.HitungTotal;
var
  Volume, Harga, Total: Double;
  FS: TFormatSettings;
begin
  // Pastikan Delphi membaca desimal dengan benar
  FS := TFormatSettings.Create;
  FS.DecimalSeparator := '.';
  // Konversi string ke float dengan format desimal yang benar
  if not TryStrToFloat(EditVolume.Text, Volume, FS) then
    Volume := 0;
  if not TryStrToFloat(EditHarga.Text, Harga, FS) then
    Harga := 0;
  // Hitung total harga
  Total := Volume * Harga;
  // Tampilkan hasil dengan format yang rapi
  EditTotalHarga.Text := Total.ToString;
end;

procedure TFormKelolaBarang.ComboBoxBarangChange(Sender: TObject);
var BarangID: Integer;
begin
  if ComboBoxBarang.ItemIndex = -1 then Exit;
  BarangID := Integer(ComboBoxBarang.Items.Objects[ComboBoxBarang.ItemIndex]);
  FDQueryBarang.Close;
  FDQueryBarang.SQL.Text := 'SELECT * FROM tbl_barang WHERE id_barang = :id';

FDQueryBarang.ParamByName('id').AsInteger := BarangID;
  FDQueryBarang.Open;
  if not FDQueryBarang.IsEmpty then
  begin
    EditSatuan.Text := FDQueryBarang.FieldByName('satuan').AsString;
    EditHarga.Text := FDQueryBarang.FieldByName('harga').AsString;
    HitungTotal;
  end
  else
  begin
    EditSatuan.Clear;
    EditHarga.Clear;
  end;
end;
procedure TFormKelolaBarang.EditVolumeKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  HitungTotal;
end;
procedure TFormKelolaBarang.FormShow(Sender: TObject);
begin
```

```
 ComboBoxBarang.Items.Clear;
 FDQueryBarang.Open;
 FDQueryBarang.First;
 while not FDQueryBarang.Eof do
 begin
ComboBoxBarang.Items.AddObject(FDQueryBarang.FieldByName('nama_barang').AsString,
TObject(FDQueryBarang.FieldByName('id_barang').AsInteger));
   FDQueryBarang.Next;
 end;
end;
procedure TFormKelolaBarang.LoadData(ASelectedID: Integer);
begin
 FSelectedID := ASelectedID;

 if FSelectedID = 0 then
 begin
   ShowMessage('ID RAB tidak valid!');
   Exit;
 end;
 if not FDConnection1.Connected then
   FDConnection1.Connected := True;
 FDQuery1.Close;
 FDQuery1.SQL.Text :=
   'SELECT d.id_detail, b.nama_barang, d.volume, b.satuan, b.harga, d.total_harga ' +
   'FROM tbl_rab_detail d ' +
   'LEFT JOIN tbl_barang b ON d.id_barang = b.id_barang ' +
   'WHERE d.id_uraian = :id';
FDQuery1.ParamByName('id').AsInteger := FSelectedID;
 FDQuery1.Open;
 EditIDUraian.Text := FSelectedID.ToString();
end;
end.
```

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

Mahasiswa : EKA MULFIANI	Pembimbing I : Ahmad Selao, S.T.P,M.Sc
NIM : 220280053	Pembimbing II : Marlina, S.Kom.,M.Kom
Judul Skripsi : SISTEM INFORMASI RENCANA ANGGARAN BIAYA PADA PERUSAHAAN CV.PATODONGI	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Dperjelas RAB untuk apa ? proyek tertentu		Konsultasi 1 - Tinjau cv patodongi - Rencana Anggaran - laporan	2/2/2024
Konsultasi 2 Jika Rumah, jembatan/ proyek lainnya		Konsultasi 2 - Perbaiki: format penulisan - Pd latar belakang jelaskan apa Ditinjitanya cv - Perbaiki: Daftar Pustaka	5/2/2024
Konsultasi 3 Ada fitur (gambar) detail itg /konstruksi dll		Konsultasi 3 - Buat Desain Sistem berjalan dlm bentuk flowchart	
Konsultasi 4 Perbaiki flowchart/ Digram usecase lengkapi desain interface		Konsultasi 4 - Buat 1 Halaman Kerangka Pikir	19/02/2024
Konsultasi 5 Acc seminar proposal		Konsultasi 5 ACC Ujian Proposal	1/04/2024

Lanjut ke halaman sebelah...

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

KARTU MONITORING BIMBINGAN

MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

Mahasiswa : EKA MULFIANI	Pembimbing I : Ahmad Selao, S.T.P,M.Sc
NIM : 220280053	Pembimbing II : Marlina, S.Kom.,M.Kom
Judul Skripsi : SISTEM INFORMASI RENCANA ANGGARAN BIAYA PADA PERUSAHAAN CV.PATODONGI	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 Silahkan tulisan pada gambar out put/hasil biar di baca!		Konsultasi 1 - Password : gunakan tanda • atau * - Lengkapi Aplikasinya	
Konsultasi 2 Buat contoh cetakan RAB nya, perbaiki & bertahap (Sampel)		Konsultasi 2 - Foto perusahaan pd Home/Tampilan Utama - Pd Tombol Cetak.	
Konsultasi 3 Bahan referensi yg ada di lampirkan!		Konsultasi 3 langsung terlihat laporan dlm bentuk Pdf	
Konsultasi 4 Perlu uji software sebelum maju ujian!		Konsultasi 4 - Lanjutkan ke dokumen	
Konsultasi 5 Khususnya ada cuplikan project (Kuasaran Asing) dijadikan dasar & buat RAB.		Konsultasi 5 - Perbaiki Format penulisan pd daftar Isi - Bhs Asing dicetak Miring	

Lanjut ke halaman sebelah...

Perhatian :

- Mahasiswa wajib konsultasi minimal 5 kali
- Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
- Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
- Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik

Lanjutan ...

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 6 Acc Seminar Proposal telah dilaksanakan		Konsultasi 6 - Tambahkan Rancangan Database nya - Berupa tabelnya. - Relasi Database	
Konsultasi 7		Konsultasi 7 Acc Ujian Hasil	
Konsultasi 8		Konsultasi 8 Perbaiki Saran dari Penguji pd saat Hasil	
Konsultasi 9		Konsultasi 9 Acc Ujian Tutup	
Konsultasi 10		Konsultasi 10	

Parepare, 24 Februari 2024

Mengetahui
Ketua Program Studi

Marlina, S.Kom.,M.Kom.
NBM. 1162 680

Mahasiswa

EKA MULFIANI
NIM. 220280053

Perhatian :

1. Mahasiswa wajib konsultasi minimal 5 kali
2. Kartu ini wajib dibawa oleh mahasiswa disetiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib dilampirkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak timbal balik