LAMPIRAN A

Listing Program Aplikasi

```
unit U_Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, Gauges, jpeg, SMSComp, DB, DBTables,
  Grids, DBGrids;

type
  TF_Main = class(TForm)
    Panel1: TPanel;
    pnlTime: TPanel;
    pnlDate: TPanel;
    lblJudulTA: TLabel;
    Memo1: TMemo;
    Splitter1: TSplitter;
    sbConnect: TSpeedButton;
    sbDatabases: TSpeedButton;
    gbStatus: TGroupBox;
    stStatus: TStaticText;
    gBattery: TGauge;
    gSignal: TGauge;
    lblBattery: TLabel;
    lblSignal: TLabel;
    lblJudulMain: TLabel;
    Image1: TImage;
    sbProcess: TSpeedButton;
    lblSource: TLabel;
    lblDestination: TLabel;
    eDestination: TEdit;
    eSource: TEdit;
    Timer1: TTimer;
    OxygenSMS1: TOxygenSMS;
    Timer2: TTimer;
    sbAbout: TSpeedButton;
    gbConnection: TGroupBox;
    cbAutoResponder: TCheckBox;
    qProcess: TQuery;
    Timer3: TTimer;
    qTanda: TQuery;
    sbDataUser: TSpeedButton;
    Splitter2: TSplitter;
    procedure sbDatabasesClick(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure sbConnectClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure Timer2Timer(Sender: TObject);
```

```
    procedure sbAboutClick(Sender: TObject);
    procedure sbProcessClick(Sender: TObject);
    procedure Timer3Timer(Sender: TObject);
    procedure OxygenSMS1SMSMessageReceived(Index: Integer; Time: TDateTime;
      Text, Send: String; Pict: TBitmap);
    procedure sbDataUserClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  F_Main: TF_Main;
  Connect : Boolean;

implementation

uses U_Data, U_About, U_ShowModal, U_User;

{$R *.dfm}

Function GetConnect : Boolean;
begin
  Connect := false;
  if F_Main.OxygenSMS1.Open then
  Connect := true;
  Result := Connect;
end;

Procedure Connected;
begin
  if not Connect then exit;
  with F_Main do
  begin
  sbConnect.Caption := 'Disconnect';
  sbConnect.Glyph.LoadFromFile('D:\Document\Program\disconnect.bmp');
  stStatus.Caption := 'Connected';
  stStatus.Font.Color := clLime;
  eSource.Enabled := false;
  eDestination.Enabled := false;
  Application.ProcessMessages;
  case OxygenSMS1.BatteryLevel of
    0..29 : gBattery.ForeColor := clRed;
    30..69 : gBattery.ForeColor := clYellow;
    70..100 : gBattery.ForeColor := clBlue;
  end;
  gBattery.Progress := OxygenSMS1.BatteryLevel;
  Application.ProcessMessages;
```

```
  case OxygenSMS1.SignalLevel of
    0..29 : gSignal.ForeColor := clRed;
    30..69 : gSignal.ForeColor := clYellow;
    70..100 : gSignal.ForeColor := clBlue;
  end;
  gSignal.Progress := OxygenSMS1.SignalLevel;
  Application.ProcessMessages;
  Timer2.Enabled := true;
  end;
end;

procedure TF_Main.sbDatabasesClick(Sender: TObject);
begin
  F_Data.Table1.Refresh;
  F_Data.Show;
  F_Data.WindowState := wsMaximized;
  F_Data.eSearch.SetFocus;
end;

procedure TF_Main.Timer1Timer(Sender: TObject);
begin
  pnlTime.Caption := TimeToStr(time);
  pnlDate.Caption := FormatDateTime('ddd, d mmm yyyy',date);
  LblJudulTA.Left := LblJudulTA.Left-4;
  if LblJudulTA.Left<-450 then LblJudulTA.Left:=820;
end;

procedure TF_Main.FormCreate(Sender: TObject);
begin
  cbAutoResponder.Enabled := false;
  pnlTime.Caption := TimeToStr(time);
  pnlTime.Font.Color := clLime;
  pnlTime.Font.Size := 10;
  pnlDate.Caption := FormatDateTime('ddd, d mmm yyyy',date);
  pnlDate.Font.Color := clLime;
  pnlDate.Font.Size := 10;
  LblJudulTA.Left := 820;
  stStatus.Font.Size := 12;
  eSource.CharCase := ecUpperCase;
  eDestination.CharCase := ecUpperCase;
  Timer2.Enabled := false;
  F_Main.WindowState := wsMaximized ;
  Memo1.Clear;
  Memo1.Enabled := false;
end;
```

```
procedure TF_Main.FormShow(Sender: TObject);
begin
  eSource.SetFocus;
end;

procedure TF_Main.sbConnectClick(Sender: TObject);
begin
  if Connect then
  begin
    Connect := false;
    Timer2.Enabled := false;
    sbConnect.Caption := 'Connect to HP';
    sbConnect.Glyph.LoadFromFile('D:\Document\Program\connect.bmp');
    gSignal.Progress := 0;
    gBattery.Progress := 0;
    stStatus.Caption := 'Disconnect';
    stStatus.font.Color := clred;
    eSource.Enabled := true;
    eDestination.Enabled := true;
  end
  else
  begin
    if not GetConnect then
    begin
      Messagedlg('Failed to connect',mtInformation,[mbOK],0);
      OxygenSMS1.Close;
    end
    else
      Connected;
  end;
end;

procedure TF_Main.FormClose(Sender: TObject; var Action: TCloseAction);
var str : string;
begin
  str := 'Anda yakin akan keluar dari aplikasi ini?';
  if (Application.MessageBox(PChar(str),'Information',MB_YESNO or
MB_ICONQUESTION)
  = IDNO) then Application.Run
  else OxygenSMS1.Close;
end;

procedure TF_Main.Timer2Timer(Sender: TObject);
begin
  if not OxygenSMS1.CheckConnection then
  begin
    Timer2.Enabled := false;
    Application.ProcessMessages;
    gBattery.Progress := OxygenSMS1.BatteryLevel;
    gSignal.Progress := OxygenSMS1.SignalLevel;
```

```
    sbConnect.Caption := 'Connect to HP';
    sbConnect.Glyph.LoadFromFile('D:\Document\Program\connect.bmp');
    gSignal.Progress := 0;
    gBattery.Progress := 0;
    stStatus.Caption := 'Disconnect';
    stStatus.font.Color := clred;
    if OxygenSMS1.Open then
    begin
      sbConnect.Caption := 'Disconnect';
      sbConnect.Glyph.LoadFromFile('D:\Document\Program\disconnect.bmp');
      stStatus.Caption := 'Connected';
      stStatus.Font.Color := clLime;
    end
    else
    begin
      stStatus.Caption := 'Disconnect';
      stStatus.font.Color := clred;
      sbConnect.Caption := 'Connect to HP';
      sbConnect.Glyph.LoadFromFile('D:\Document\Program\connect.bmp');
    end;
    Application.ProcessMessages;
    Timer2.Enabled := true;
  end
  else
  begin
    Application.ProcessMessages;
      case OxygenSMS1.BatteryLevel of
        0..29 : gBattery.ForeColor := clRed;
        30..69 : gBattery.ForeColor := clYellow;
        70..100 : gBattery.ForeColor := clBlue;
      end;
      gBattery.Progress := OxygenSMS1.BatteryLevel;
      Application.ProcessMessages;
      case OxygenSMS1.SignalLevel of
        0..29 : gSignal.ForeColor := clRed;
        30..69 : gSignal.ForeColor := clYellow;
        70..100 : gSignal.ForeColor := clBlue;
      end;
      gSignal.Progress := OxygenSMS1.SignalLevel;
      Application.ProcessMessages;
  end;
end;
```

```
procedure TF_Main.sbAboutClick(Sender: TObject);
begin
  F_About.show;
  F_About.WindowState := wsMaximized;
end;

procedure TF_Main.sbProcessClick(Sender: TObject);
var s,k : String;
    u,v,w : Real;
begin
  with qTanda do
  begin
    SQL.Clear;
    SQL.Add('UPDATE tbljalan SET Tanda =:t');
    Prepare;
    ParamByName('t').AsString := '';
    ExecSQL;
  end;
  s :='SELECT * FROM tbljalan WHERE Kota1 = "%s" OR Kota2 = "%s"';
  qProcess.Close;
  qProcess.SQL.Clear;
  qProcess.SQL.Add(Format(s,[eSource.Text,eSource.Text]));
  qProcess.Open;
  if (eSource.Text = '') or (eDestination.Text = '') then
  begin
    Memo1.Lines.Clear;
    ShowMessage('Source dan Destination harus diisi');
  end
  else if (qProcess.FieldByName('Kota1').AsString <> '') and
  (qProcess.FieldByName('Kota2').AsString <> '') then
  begin
    s := 'SELECT * FROM tbljalan WHERE Kota1 = "%s" '+
    'OR Kota2 = "%s"';
    qProcess.Close;
    qProcess.SQL.Clear;
    qProcess.SQL.Add(Format(s,[eDestination.Text,eDestination.Text]));
    qProcess.Open;
    if (qProcess.FieldByName('Kota1').AsString <> '') and
    (qProcess.FieldByName('Kota2').AsString <> '') then
    begin
      s := 'SELECT * FROM tbljalan WHERE Kota1 = "%s" '+
      'OR Kota2 = "%s"';
      qProcess.Close;
      qProcess.SQL.Clear;
      qProcess.SQL.Add(Format(s,[eSource.Text,eSource.Text]));
      qProcess.Open;
      qProcess.First;
      u := qProcess.Fields[3].AsFloat;
      while not qProcess.Eof do
      begin
```

```
  qProcess.Next;
  v := qProcess.Fields[3].AsFloat;
  if u < v then u := u else u := v;
end;
with qTanda do
begin
  SQL.Clear;
  SQL.Add('UPDATE tbljalan SET Tanda =:t WHERE kota1=:k1 or kota2=:k2');
  Prepare;
  ParamByName('t').AsString := 'X';
  ParamByName('k1').AsString := eSource.Text;
  ParamByName('k2').AsString := eSource.Text;
  ExecSQL;
end;
s := 'SELECT * FROM tbljalan WHERE Jarak_dlm_Km = %10.2f ' +
'AND (Kota1="%s" OR Kota2="%s")';
qProcess.Close;
qProcess.SQL.Clear;
qProcess.SQL.Add(Format(s,[u,eSource.Text,eSource.Text]));
qProcess.Open;
if (qProcess.FieldByName('Kota1').AsString = eSource.Text) then
k := qProcess.Fields[2].AsString else k := qProcess.Fields[1].AsString;
Memo1.Lines.Clear;
Memo1.Lines.Add(eSource.Text);
Memo1.Lines.Add(k);
while not (k = eDestination.Text) do
begin
  s := 'SELECT * FROM tbljalan WHERE Tanda NOT IN ("X")'+
  'AND (Kota1 = "%s" or Kota2 = "%s")';
  qProcess.Close;
  qProcess.SQL.Clear;
  qProcess.SQL.Add(Format(s,[k,k]));
  qProcess.Open;
  qProcess.First;
  w := qProcess.Fields[3].AsFloat;
  while not qProcess.Eof do
  begin
    qProcess.Next;
    v := qProcess.Fields[3].AsFloat;
    if w < v then w := w else w := v;
  end;
  with qTanda do
  begin
    SQL.Clear;
    SQL.Add('UPDATE tbljalan SET Tanda =:t WHERE kota1=:k1 or kota2=:k2');
    Prepare;
    ParamByName('t').AsString := 'X';
    ParamByName('k1').AsString := k;
    ParamByName('k2').AsString := k;
    ExecSQL;
```

```
      end;
      s := 'SELECT * FROM tbljalan WHERE Jarak_dlm_Km = %10.2f ' +
      'AND (Kota1="%s" OR Kota2="%s")';
      qProcess.Close;
      qProcess.SQL.Clear;
      qProcess.SQL.Add(Format(s,[w,k,k]));
      qProcess.Open;
      if (qProcess.FieldByName('Kota1').AsString = k) then
      k := qProcess.Fields[2].AsString else k := qProcess.Fields[1].AsString;
      Memo1.Lines.Add(k);
      u := u + w;
    end;
    Memo1.Lines.Add('Total Jarak Terpendek = ' +FloatToStr(u)+ ' Km');
   end
   else
   begin
    ShowMessage('Data tidak ditemukan');
    Memo1.Lines.Clear;
   end;
  end
  else
  begin
   ShowMessage('Data tidak ditemukan');
   Memo1.Lines.Clear;
  end;
end;

procedure TF_Main.Timer3Timer(Sender: TObject);
begin
  if not connect then
  cbAutoResponder.Enabled := false
  else cbAutoResponder.Enabled := true;
end;

procedure TF_Main.OxygenSMS1SMSMessageReceived(Index: Integer;
  Time: TDateTime; Text, Send: String; Pict: TBitmap);
var s,k,m,ka,kt,format1,format2 : string;
   u,v,w : Real;
   i,space,l,t : integer;
begin
  if (cbAutoResponder.Checked) then
  begin
   with F_Data do
   begin
    Table2.Append;
    Table2.FieldByName('NoPengirim').AsString := Send;
    Table2.FieldByName('IsiPesan').AsString := Text;
    Table2.FieldByName('Waktu').AsDateTime := Time;
    Table2.Post;
   end;
```

```
space := 0;
l := Length(Text);
for i := 4 to Length(Text) do
begin
  m := copy(Text,i,1);
  if (m = ' ') and (space = 0) then space := i;
end;
t := space+4;
ka := copy(Text,4,space-4);
kt := copy(Text,t,l-t+1);
format1 := copy(Text,1,3);
format2 := copy(Text,space+1,3);
if (format1 = 'KA ') and (format2 = 'KT ') then
begin
with qTanda do
  begin
    SQL.Clear;
    SQL.Add('UPDATE tbljalan SET Tanda =:t');
    Prepare;
    ParamByName('t').AsString := '';
    ExecSQL;
  end;
  s :='SELECT * FROM tbljalan WHERE Kota1 = "%s" OR Kota2 = "%s"';
  qProcess.Close;
  qProcess.SQL.Clear;
  qProcess.SQL.Add(Format(s,[ka,ka]));
  qProcess.Open;
  if (qProcess.FieldByName('Kota1').AsString <> '') and
  (qProcess.FieldByName('Kota2').AsString <> '') then
  begin
    s := 'SELECT * FROM tbljalan WHERE Kota1 = "%s" '+
    'OR Kota2 = "%s"';
    qProcess.Close;
    qProcess.SQL.Clear;
    qProcess.SQL.Add(Format(s,[kt,kt]));
    qProcess.Open;
    if (qProcess.FieldByName('Kota1').AsString <> '') and
    (qProcess.FieldByName('Kota2').AsString <> '') then
    begin
      s := 'SELECT * FROM tbljalan WHERE Kota1 = "%s" '+
      'OR Kota2 = "%s"';
      qProcess.Close;
      qProcess.SQL.Clear;
      qProcess.SQL.Add(Format(s,[ka,ka]));
      qProcess.Open;
      qProcess.First;
      u := qProcess.Fields[3].AsFloat;
      while not qProcess.Eof do
      begin
        qProcess.Next;
```

```
  v := qProcess.Fields[3].AsFloat;
  if u < v then u := u else u := v;
end;
with qTanda do
begin
  SQL.Clear;
  SQL.Add('UPDATE tbljalan SET Tanda =:t WHERE kota1=:k1 or kota2=:k2');
  Prepare;
  ParamByName('t').AsString := 'X';
  ParamByName('k1').AsString := ka;
  ParamByName('k2').AsString := ka;
  ExecSQL;
end;
s := 'SELECT * FROM tbljalan WHERE Jarak_dlm_Km = %10.2f ' +
'AND (Kota1="%s" OR Kota2="%s")';
qProcess.Close;
qProcess.SQL.Clear;
qProcess.SQL.Add(Format(s,[u,ka,ka]));
qProcess.Open;
if (qProcess.FieldByName('Kota1').AsString = ka) then
k := qProcess.Fields[2].AsString else k := qProcess.Fields[1].AsString;
Memo1.Lines.Clear;
Memo1.Lines.Add(ka);
Memo1.Lines.Add(k);
while not (k = kt) do
begin
  s := 'SELECT * FROM tbljalan WHERE Tanda NOT IN ("X")'+
  'AND (Kota1 = "%s" or Kota2 = "%s")';
  qProcess.Close;
  qProcess.SQL.Clear;
  qProcess.SQL.Add(Format(s,[k,k]));
  qProcess.Open;
  qProcess.First;
  w := qProcess.Fields[3].AsFloat;
  while not qProcess.Eof do
  begin
    qProcess.Next;
    v := qProcess.Fields[3].AsFloat;
    if w < v then w := w else w := v;
  end;
  with qTanda do
  begin
    SQL.Clear;
    SQL.Add('UPDATE tbljalan SET Tanda =:t WHERE kota1=:k1 or kota2=:k2');
    Prepare;
    ParamByName('t').AsString := 'X';
    ParamByName('k1').AsString := k;
    ParamByName('k2').AsString := k;
    ExecSQL;
  end;
```

```
        s := 'SELECT * FROM tbljalan WHERE Jarak_dlm_Km = %10.2f ' +
        'AND (Kota1="%s" OR Kota2="%s")';
        qProcess.Close;
        qProcess.SQL.Clear;
        qProcess.SQL.Add(Format(s,[w,k,k]));
        qProcess.Open;
        if (qProcess.FieldByName('Kota1').AsString = k) then
        k := qProcess.Fields[2].AsString else k := qProcess.Fields[1].AsString;
        Memo1.Lines.Add(k);
        u := u + w;
       end;
       Memo1.Lines.Add('Total Jarak Terpendek = ' +FloatToStr(u)+ ' Km');
       s := Memo1.Lines.Text;
       OxygenSMS1.SendSMSMessage(Send,s,167,true,false,nil);
     end
     else
     begin
       Memo1.Lines.Clear;
       s := 'Data tidak ditemukan';
       OxygenSMS1.SendSMSMessage(Send,s,167,true,false,nil);
     end;
   end
   else
   begin
     Memo1.Lines.Clear;
     s := 'Data tidak ditemukan';
     OxygenSMS1.SendSMSMessage(Send,s,167,true,false,nil);
   end;
  end
  else
  begin
    s := 'FORMAT SALAH, KETIK KA<SPASI><NAMA KOTA ASAL><SPASI>KT'+
    '<SPASI><NAMA KOTA TUJUAN> KIRIM KE NOMOR INI.';
    OxygenSMS1.SendSMSMessage(Send,s,167,true,false,nil);
  end;
  end;
end;

procedure TF_Main.sbDataUserClick(Sender: TObject);
begin
  F_User.Show;
end;

end.
```

```
unit U_Data;

interface

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

type
  TF_Data = class(TForm)
    Database1: TDatabase;
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    lblJudulData: TLabel;
    btnAdd: TButton;
    btnDelete: TButton;
    btnEdit: TButton;
    btnClose: TButton;
    eSearch: TEdit;
    lblSearch: TLabel;
    btnSearch: TButton;
    qSearch: TQuery;
    DBGrid2: TDBGrid;
    DataSource2: TDataSource;
    qAdd: TQuery;
    Table2: TTable;
    procedure btnCloseClick(Sender: TObject);
    procedure btnAddClick(Sender: TObject);
    procedure btnEditClick(Sender: TObject);
    procedure btnDeleteClick(Sender: TObject);
    procedure btnSearchClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  F_Data: TF_Data;

implementation

uses U_Main, U_About, U_ShowModal, U_User;

{$R *.dfm}
```

```
procedure TF_Data.btnCloseClick(Sender: TObject);
begin
  Close;
end;

procedure TF_Data.btnAddClick(Sender: TObject);
var s : string;
begin
  F_ShowModal := TF_ShowModal.Create(Self);
  With F_ShowModal do
  begin
    eKota1.Text := '';
    eKota2.Text := '';
    eJarak.Text := '';
    Caption := 'Add Data';
    lblJudulShow.Caption := 'Menu Menambah Data';
    ActiveControl := eKota1;
  end;

  F_ShowModal.ShowModal;
  if (F_ShowModal.ModalResult = mrOK) then
  begin
    with Table1, F_ShowModal do
    begin
      try
        StrToFloat(eJarak.Text);
      except
        on EConvertError do
        begin
          ShowMessage('Jarak harus diisi dengan bilangan');
          F_ShowModal.Free;
        end;
      end;
      if (eKota1.Text = '') or (eKota2.Text = '') or
      (eJarak.Text = '') then ShowMessage('Ada data yang belum diisi')
      else
      begin
        s := 'SELECT * FROM tbljalan WHERE Kota1 = "%s" '+
        'AND Kota2 = "%s"';
        qAdd.Close;
        qAdd.SQL.Clear;
        qAdd.SQL.Add(Format(s,[eKota1.Text,eKota2.Text]));
        qAdd.Open;
        if ((qAdd.FieldByName('Kota1').AsString = eKota1.Text) and
        (qAdd.FieldByName('Kota2').AsString = eKota2.Text)) then
        ShowMessage('Nama kota tersebut telah terdaftar')
        else
        begin
          s := 'SELECT * FROM tbljalan WHERE Kota2 = "%s" '+
          'AND Kota1 = "%s"';
```

```
        qAdd.Close;
        qAdd.SQL.Clear;
        qAdd.SQL.Add(Format(s,[eKota1.Text,eKota2.Text]));
        qAdd.Open;
        if ((qAdd.FieldByName('Kota2').AsString = eKota1.Text) and
        (qAdd.FieldByName('Kota1').AsString = eKota2.Text)) then
        ShowMessage('Nama kota tersebut telah terdaftar')
        else
        begin
          Append;
          FieldByName('Kota1').AsString := eKota1.Text;
          FieldByName('Kota2').AsString := eKota2.Text;
          FieldByName('Jarak_dlm_Km').AsString := eJarak.Text;
          Post;
          F_ShowModal.Free;
        end;
      end;
    end;
  end;
 end;
 Table1.Refresh;
end;

procedure TF_Data.btnEditClick(Sender: TObject);
var s : string;
begin
 F_ShowModal := TF_ShowModal.Create(Self);
 With F_ShowModal do
 begin
  eKota1.Text := Table1.Fields[1].AsString;
  eKota2.Text := Table1.Fields[2].AsString;
  eJarak.Text := Table1.Fields[3].AsString;
  Caption := 'Edit Data';
  lblJudulShow.Caption := 'Menu Mengedit Data';
  ActiveControl := eKota1;
 end;

 F_ShowModal.ShowModal;

 if (F_ShowModal.ModalResult = mrOK) then
 begin
  with Table1, F_ShowModal do
  begin
    try
      StrToFloat(eJarak.Text);
    except
      on EConvertError do
      begin
        ShowMessage ('Jarak harus diisi dengan bilangan');
        F_ShowModal.Free;
```

```
      end;
    end;
    if (eKota1.Text = '') or (eKota2.Text = '') or
    (eJarak.Text = '') then ShowMessage('Ada data yang belum diisi')
    else
    begin
      s := 'SELECT * FROM tbljalan WHERE Kota1 = "%s" '+
      'AND Kota2 = "%s"';
      qAdd.Close;
      qAdd.SQL.Clear;
      qAdd.SQL.Add(Format(s,[eKota1.Text,eKota2.Text]));
      qAdd.Open;
      if ((qAdd.FieldByName('Kota1').AsString = eKota1.Text) and
      ((qAdd.FieldByName('Kota2').AsString = eKota2.Text) and
      (qAdd.FieldByName('Jarak_dlm_Km').AsString = eJarak.Text))) then
        ShowMessage('Nama kota tersebut telah terdaftar')
      else
      begin
        s := 'SELECT * FROM tbljalan WHERE Kota2 = "%s" '+
        'AND Kota1 = "%s"';
        qAdd.Close;
        qAdd.SQL.Clear;
        qAdd.SQL.Add(Format(s,[eKota1.Text,eKota2.Text]));
        qAdd.Open;
        if ((qAdd.FieldByName('Kota2').AsString = eKota1.Text) and
        ((qAdd.FieldByName('Kota1').AsString = eKota2.Text) and
        (qAdd.FieldByName('Jarak_dlm_Km').AsString = eJarak.Text))) then
        ShowMessage('Nama kota tersebut telah terdaftar')
        else
        begin
          Edit;
          FieldByName('Kota1').AsString := eKota1.Text;
          FieldByName('Kota2').AsString := eKota2.Text;
          FieldByName('Jarak_dlm_Km').AsString := eJarak.Text;
          Post;
          F_ShowModal.Free;
        end;
      end;
    end;
  end;
 end;
 Table1.Refresh;
end;
```

```
procedure TF_Data.btnDeleteClick(Sender: TObject);
var str : string;
begin
  str := 'Anda yakin akan menghapus jalan antara kota '
  + Table1.Fields[1].AsString + ' dan ' + Table1.Fields[2].AsString +'?';
  if (Application.MessageBox(PChar(str),'WARNING',MB_YESNO or
MB_ICONEXCLAMATION)
  = IDYES) then
  Table1.Delete;
end;

procedure TF_Data.btnSearchClick(Sender: TObject);
var s : string;
begin
  s := 'SELECT Kode_Jalan,Kota1,Kota2,Jarak_dlm_Km FROM tbljalan WHERE Kota1
= "%s" '+
  'OR Kota2 = "%s" ORDER BY Kota1,Kota2';
  qSearch.Close;
  qSearch.SQL.Clear;
  qSearch.SQL.Add(Format(s,[eSearch.Text,eSearch.Text]));
  qSearch.Open;
  if (qSearch.FieldByName('Kota1').AsString = '') or
  (qSearch.FieldByName('Kota1').AsString = '') then
  ShowMessage('Data tidak ditemukan');
end;

procedure TF_Data.FormShow(Sender: TObject);
begin
  eSearch.SetFocus;
end;

end.
```

LAMPIRAN B

Foto Alat

Gambar B.1 Induk Kabel Data

Gambar B.2 Kabel Data untuk Berbagai Tipe Nokia

Gambar B.3 Perhubungan Antara Komputer dengan Kabel Data

LAMPIRAN C

Peta yang Diamati

Gambar C.1 Peta Riau

LAMPIRAN D

Tabel Data Jalan

Tabel D.1 Tabel Data Jalan

Kode_Jalan	Kota1	Kota2	Jarak_dlm_Km
1	KUBU	SEDINGINAN	71.5
2	SEDINGINAN	UJUNGTANJUNG	35.75
3	UJUNGTANJUNG	BAGANSIAPIAPI	75.4
4	UJUNGTANJUNG	DUMAI	36.4
5	UJUNGTANJUNG	DURI	49.72
6	DUMAI	DURI	58.5
7	DURI	PETAPAHAN	140
8	DURI	MINAS	81.25
9	PETAPAHAN	MINAS	85.8
10	PETAPAHAN	AIRTIRIS	28.27
11	PETAPAHAN	BANGKINANG	30.55
12	PETAPAHAN	BATUBESURAT	52
13	PETAPAHAN	TANDUN	49.72
14	PETAPAHAN	SIMANINIK	45.5
15	MINAS	PEKANBARU	27.95
16	PEKANBARU	SIMPANGTIGA	8.77
17	SIMPANGTIGA	AIRTIRIS	46.8
18	SIMPANGTIGA	LIPATKAIN	66.92
19	LIPATKAIN	AIRTIRIS	63.05
20	AIRTIRIS	BANGKINANG	6.52
21	BANGKINANG	BATUBESURAT	40.95
22	BANGKINANG	TANDUN	56.55
23	BATUBESURAT	ROKAN	54.6
24	BATUBESURAT	TANDUN	39
25	TANDUN	ROKAN	30.55
26	TANDUN	UJUNGBATU	18.85
27	TANDUN	SIMANINIK	33.15
28	ROKAN	UJUNGBATU	25.08
29	SIMANINIK	KOTALAMA	19.5
30	UJUNGBATU	KOTALAMA	22.75

31	UJUNGBATU	PASIRPANGARAIAN	37.05
32	KOTALAMA	KOTATENGAH	44.79
33	KOTALAMA	PASIRPANGARAIAN	51.03
34	PASIRPANGARAIAN	KOTATENGAH	55.25
35	PASIRPANGARAIAN	DALUDALU	34.45
36	KOTATENGAH	DALUDALU	52
37	LIPATKAIN	MUARALEMBU	55.25
38	MUARALEMBU	LUBUKJAMBI	58.5
39	MUARALEMBU	TELUKKUANTAN	36.73
40	LUBUKJAMBI	TELUKKUANTAN	30.23
41	TELUKKUANTAN	LANGGAM	122.85
42	TELUKKUANTAN	BASERAH	31.2
43	LANGGAM	BASERAH	115.05
44	BASERAH	CARENT	22.1
45	CARENT	PERENAP	22.75
46	CARENT	AIRMOLEK	63.05
47	PERENAP	AIRMOLEK	59.8
48	AIRMOLEK	SOREKSATU	79.62
49	AIRMOLEK	SEBRIDA	64.85
50	AIRMOLEK	RENGAT	39
51	SOREKSATU	DAYUN	109.53
52	DAYUN	SIAKSRIINDRAPURA	14.95
53	SEBRIDA	KERINTANG	26
54	SEBRIDA	RENGAT	67.27
55	KERINTANG	KEMUNING	15.44
56	RENGAT	TELUKKIAMBANG	41.6
57	TELUKKIAMBANG	SUNGAISALAK	23.08
58	SUNGAISALAK	TEMBILAHAN	27.3
59	SEBRIDA	PERENAP	90.5
60	SEBRIDA	LUBUKJAMBI	126.75

LAMPIRAN E

Perancangan Aplikasi

Menu utama pada aplikasi terdiri dari beberapa komponen yaitu :

1. *SpeedButton*

1.1	sbDatabases	:	Untuk melihat data jalan.
1.1.1	Database1	:	Sebagai *database* aplikasi.
1.1.2	Table1	:	Tabel untuk data jalan.
1.1.3	Table2	:	Tabel untuk data SMS.
1.1.4	DataSource1	:	Untuk mengkoneksi antara table1 dan DBGrid1.
1.1.5	DataSource2	:	Untuk mengkoneksi antara qSearch dan DBGrid2.
1.1.6	qAdd	:	*Query* untuk menambah data.
1.1.7	qSearch	:	*Query* untuk mencari data.
1.1.8	DBGrid1	:	Untuk tampilan semua data jalan.
1.1.9	DBGrid2	:	Untuk tampilan nama kota yang dicari.
1.1.10	btnAdd	:	Untuk menambah data jalan.
1.1.11	btnEdit	:	Untuk mengedit data jalan.
1.1.12	btnDelete	:	Untuk menghapus data jalan.
1.1.13	btnClose	:	Untuk menutup *form database*.
1.1.14	btnSearch	:	Untuk mencari nama kota.
1.1.15	lblJudulData	:	Label untuk judul *form database*.
1.1.16	lblSearch	:	Label untuk *search*.
1.1.17	eSearch	:	Isian untuk *search*.
1.2	sbAbout	:	Untuk mengetahui informasi program.
1.3	sbConnect	:	Untuk membuat koneksi dengan *handset server*.
1.4	sbDataUser	:	Untuk melihat data SMS masuk.
1.5	sbProcess	:	Untuk melakukan proses pencarian rute terpendek.

2. *GroupBox*

2.1	gbConnection	:	Terdiri dari sbConnect, sbDataUser, dan cbAutoResponder.
2.2	gbStatus	:	Terdiri dari stStatus, lblSignal, lblBattery, gSignal, dan gBattery.

3. *CheckBox*

3.1	cbAutoResponder	:	Untuk mengaktifkan mode otomatis.

4. *StaticText*

4.1	stStatus	:	Untuk informasi status koneksi.

5. *Gauge*

5.1	gSignal	:	Untuk informasi level sinyal handset server.
5.2	gBattery	:	Untuk informasi level baterai handset server.

6. *Label*

6.1	lblJudulMain	:	Judul pada *form* menu utama.
6.2	lblSource	:	Label untuk kota asal.
6.3	lblDestination	:	Label untuk kota tujuan.

7. *Image*

7.1	Image1	:	Gambar peta propinsi Riau

8. *Edit*

8.1	eSource	:	Isian untuk kota asal.
8.2	eDestination	:	Isian untuk kota tujuan.

9. *Splitter*

9.1	Splitter1	:	Sebagai tanda pembatas.
9.2	Splitter2	:	Sebagai tanda pembatas.

10. *Memo*

10.1	Memo1	:	Tampilan hasil pencarian rute terpendek.

11. *Panel*

11.1	pnlDate	:	Info tanggal.
11.2	pnlTime	:	Info waktu.
11.3	Panel1	:	Info judul Tugas Akhir.

12. *Timer*

12.1	Timer1	:	Untuk animasi judul Tugas Akhir pada Panel1.
12.2	Timer2	:	Untuk perubahan level sinyal dan baterai.
12.3	Timer3	:	Untuk perubahan pada status cbAutoResponder.

13. *OxygenSMS*

13.1	OxygenSMS1	:	Untuk menerima dan mengirim SMS.

14. *Query*

14.1	qProcess	:	Proses pencarian rute terpendek.
14.2	qTanda	:	Proses penandaan.