LAMPIRAN

```
unit u_Utama;

interface

uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, ExtCtrls, StdCtrls, Buttons, Grids, ExtDlgs, IniFiles,ComCtrls, ActnList,
ToolWin, ImgList, abcbtn, Menus;

type
      ND       = record
      Node     : Tpanel;
      Koneksi : String;
    end;
           TLarik       = record
           Isi:Array of byte;
           Banyak:Integer;
         end;
            st50         = string[50];
            arrayJalur = array of word;
            TFUtama  = class(TForm)
   OpenPictureDialog1: TOpenPictureDialog;
   SaveDialog1: TSaveDialog;
   OpenDialog1: TOpenDialog;
   Panel2: TPanel;
   ScrollBoxPeta: TScrollBox;
   Image_peta: TImage;
   Image_rute: TImage;
   ActionList1: TActionList;
   Action_buka: TAction;
   Action_simpan: TAction;
   Action_buka_peta: TAction;
   ScrollBoxMenu: TScrollBox;
   btn_buat_rute: TabcToggleBtn;
   btn_buat_node: TabcToggleBtn;
   TreeView1: TTreeView;
   btn_clearnode: TabcWaveBtn;
   btn_clear_rute: TabcWaveBtn;
   abcWaveBtn3: TabcWaveBtn;
   abcWaveBtn2: TabcWaveBtn;
   abcWaveBtn1: TabcWaveBtn;
   Panel1: TPanel;
   lbl_koordinat: TLabel;
   LabelWaktuProses: TLabel;
   Panel3: TPanel;
   Label3: TLabel;
   edt_nama_node: TEdit;
   edt_index: TEdit;
```

```
Label7: TLabel;
Label4: TLabel;
edt_koord_X: TEdit;
edt_koord_Y: TEdit;
Label5: TLabel;
Label6: TLabel;
edt_koneksi: TEdit;
Panel4: TPanel;
Label1: TLabel;
ComboBox_nodeawal: TComboBox;
ComboBox_nodeakhir: TComboBox;
Label2: TLabel;
Label8: TLabel;
edt_skala: TEdit;
btn_simulate_anneal: TabcWaveBtn;
CheckBoxWindowHasil: TCheckBox;
Panel5: TPanel;
ProgressBar1: TProgressBar;
btn_diksjra: TabcWaveBtn;
PopupRute: TPopupMenu;
Bukarute1: TMenuItem;
Simpanrute1: TMenuItem;
Bukapeta1: TMenuItem;
Action_save_bmp: TAction;
SavePictureDialog1: TSavePictureDialog;
Simpanbitmap1: TMenuItem;
abcWaveBtn4: TabcWaveBtn;
abcWaveBtn5: TabcWaveBtn;
PopupNode: TPopupMenu;
Jadikannodeawal1: TMenuItem;
Jadikannodeakhir1: TMenuItem;
ImageList1: TImageList;
procedure Image_closeClick(Sender: TObject);
procedure Image_ruteMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure btn_clearnodeClick(Sender: TObject);
procedure btn_clear_ruteClick(Sender: TObject);
procedure btn_buat_ruteClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Image_ruteMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure btn_simulate_annealClick(Sender: TObject);
procedure edt_nama_nodeChange(Sender: TObject);
procedure ResetRute;
procedure Action_bukaExecute(Sender: TObject);
procedure Action_simpanExecute(Sender: TObject);
procedure Action_buka_petaExecute(Sender: TObject);
procedure btn_buat_nodeClick(Sender: TObject);
procedure CheckBoxWindowHasilClick(Sender: TObject);
procedure btn_diksjraClick(Sender: TObject);
```

```
    procedure Action_save_bmpExecute(Sender: TObject);
    procedure edt_skalaKeyPress(Sender: TObject; var Key: Char);
    procedure abcWaveBtn5Click(Sender: TObject);
    procedure Jadikannodeawal1Click(Sender: TObject);
    procedure Jadikannodeakhir1Click(Sender: TObject);
  private
    { Private declarations }
    procedure buat_node(X,Y : Integer; Nama : st50);
    procedure clear_node;
    procedure buat_koneksi(Node1,Node2 : TPanel);
    procedure Dijkstra;
    procedure NodeClick(Sender : TObject);
    procedure ExtractKoneksi(N1 : TPanel; Koneksi : string);
    procedure SimulateAnneal;
    procedure BuatListJalur;
    procedure UpdateWaktuProses;
    procedure Diksjra;
  public
    { Public declarations }
  end;
const
        sparator      = [','];
         whitespace = [','];
         maximum =100;
var
  FUtama          : TFUtama;
  MyNode          : array of  ND;
  DaftarRute      : Array of string;
  W               :Array[1..maximum,1..maximum]of integer;
  CurrentNode     : Tpanel;
  JumNode         : Word;
  PanelNode1,
  PanelNode2      : Tpanel;
  FPeta           : TIniFIle;
  Waktuawal       : TDateTime;
  BanyakKota      : Integer;

implementation
uses u_rpt_rute, u_about;

{$R *.dfm}
// MENENTUKAN JARAK ANTARA DUA NODE

function JarakNode(N1,N2 : Tpanel) : real;
begin
    Result := sqrt((N1.Left - N2.Left) * (N1.Left - N2.Left) +
              (N1.Top  - N2.Top) * (N1.Top - N2.Top));
end;

// MENAMPILKAN SEMUA JALUR DENNGAN WARNA MERAH
```

```
procedure TFUtama.ResetRute;
var
    i   : word;
begin
    Image_rute.Picture.Assign(Image_peta.Picture);

    Image_rute.Canvas.Pen.Color := clRed;
    Image_rute.Canvas.Pen.Width := 2;

    ProgressBar1.Position := 0;
    for i := 1 to JumNode do
    begin ExtractKoneksi(MyNode[i].Node,Mynode[i-1].Koneksi);
        ProgressBar1.Position := Trunc(i/(JumNode)*100);
    end;
    ProgressBar1.Position := 0;
end;

procedure TFutama.NodeClick(Sender : TObject);
begin
     if btn_buat_rute.Down then
     begin (Sender as TPanel).Color := clRed;

          if PanelNode1 <> nil then
          begin if PanelNode1 <> (Sender as TPanel) then
               begin PanelNode2 := (Sender as TPanel);
                    buat_koneksi(PanelNode1,PanelNode2);
                    PanelNode1.Color := clBlack;
                    PanelNode1 := PanelNode2;
                    PanelNode2 := nil;
               end;
          end
          else
            PanelNode1 := (Sender as Tpanel);
     end;

    CurrentNode          := (Sender as TPanel);

    edt_nama_node.Text := (Sender as TPanel).Caption;
    edt_index.Text     := IntToStr((Sender as TPanel).Tag);
    edt_koord_X.Text   := IntToStr((Sender as TPanel).Left +
ScrollBoxPeta.HorzScrollBar.Position);
    edt_koord_Y.Text   := IntToStr((Sender as TPanel).Top +
ScrollBoxPeta.VertScrollBar.Position);
    edt_koneksi.Text   := MyNode[(Sender as TPanel).Tag].Koneksi;
end;

// PROCEDURE UNTUK MEMBUAT NODE
// X : KOORDINAT X, Y : KOORDINAT Y, NAMA : NAMA NODE
```

```
procedure TFUtama.buat_node(X,Y : Integer; Nama : st50);
begin
    inc(JumNode);
    SetLength(Mynode,JumNode+1);
    MyNode[JumNode].Koneksi   := '';
    MyNode[JumNode].Node      := TPanel.Create(ScrollBoxPeta);
    with Mynode[JumNode].Node do
    begin  Parent      := ScrollBoxPeta;
           Left        := X;
           Top         := Y;
           Caption     := Nama;
           Width       := 11;
           Height      := 11;
           Color       := clBlack;
           Tag         := JumNode-1;
           Hint        := Nama + #13 + #10 + IntToStr(X) + ' , ' + IntToStr(Y);
           ShowHint    := True;
           Cursor      := crHandPoint;
           OnClick     := NodeClick;
           PopupMenu   := PopupNode;
    end;

    NodeClick(MyNode[JumNode].Node);

    // MENAMBAHKAN NAMA NODE DALAM LIST NODE AWAL
    ComboBox_nodeawal.Items.Append(Nama);

    // MENAMBAHKAN NAMA NODE DALAM LIST NODE AKHIR
    ComboBox_nodeakhir.Items.Append(Nama);

    btn_buat_rute.Enabled       :=  JumNode > 1;
    btn_clearnode.Enabled       :=  JumNode > 0;
end;
// MEMBUAT KONEKSI ANTARA 2 NODE

procedure TFUtama.buat_koneksi(Node1,Node2 : TPanel);
var
  s : string;
  i : word;

begin
     if Image_rute.Picture.Bitmap.Empty then
        Image_rute.Picture.Assign(Image_peta.Picture);

        Image_rute.Canvas.Pen.Color    := clRed;
        Image_rute.Canvas.Pen.Width    := 2;

     Image_rute.Canvas.MoveTo(Node1.Left + 6 +
ScrollBoxPeta.HorzScrollBar.Position,
                        Node1.Top  + 6 + ScrollBoxPeta.VertScrollBar.Position);
```

```
    Image_rute.Canvas.LineTo(Node2.Left + 6 + ScrollBoxPeta.HorzScrollBar.Position,
                    Node2.Top  + 6 + ScrollBoxPeta.VertScrollBar.Position);

    btn_clear_rute.Enabled          := True;
    btn_simulate_anneal.Enabled     := True;
    btn_diksjra.Enabled             := True;

    if Node1 <> Node2 then
    begin

    // MEMBUAT KONEKSI PADA NODE 1

    s    := intToStr(Node2.Tag);
    i    := pos(s,MyNode[Node1.tag].Koneksi);
    if i = 0 then
    begin  if MyNode[Node1.tag].Koneksi <> '' then
            MyNode[Node1.tag].Koneksi := MyNode[Node1.tag].Koneksi + ',';
           MyNode[Node1.tag].Koneksi := MyNode[Node1.tag].Koneksi + s;
    end
    else if i>1 then
    begin if MyNode[Node1.tag].Koneksi[i-1] <> ',' then
         begin if MyNode[Node1.tag].Koneksi <> '' then
                 MyNode[Node1.tag].Koneksi := MyNode[Node1.tag].Koneksi + ',';
               MyNode[Node1.tag].Koneksi := MyNode[Node1.tag].Koneksi + s;
         end;
    end;


    // MEMBUAT KONEKSI PADA NODE 2

    s   := intToStr(Node1.Tag);
    i   := pos(s,MyNode[Node2.tag].Koneksi);
    if i = 0 then
    begin if MyNode[Node2.tag].Koneksi <> '' then
            MyNode[Node2.tag].Koneksi := MyNode[Node2.tag].Koneksi + ',';
           MyNode[Node2.tag].Koneksi   := MyNode[Node2.tag].Koneksi + s;
    end
    else if i > 1 then
    begin if MyNode[Node2.tag].Koneksi[i-1] <> ',' then
         begin if MyNode[Node2.tag].Koneksi <> '' then
                 MyNode[Node2.tag].Koneksi := MyNode[Node2.tag].Koneksi + ',';
               MyNode[Node2.tag].Koneksi    := MyNode[Node2.tag].Koneksi + s;
         end
    end;

    end;
end;

// PROCEDURE UNTUK MENGOSONGKAN NODE (MENGHAPUS SEMUA
NODE)
```

```
procedure TFUtama.clear_node;
begin
    ComboBox_nodeawal.Clear;
    ComboBox_nodeakhir.Clear;

    Image_rute.Picture.Assign(Image_peta.Picture);
    setlength(Mynode,0);
    JumNode := 0;
    btn_buat_rute.Enabled           := False;
    btn_buat_rute.Down              := False;
    btn_clearnode.Enabled           := False;
    btn_clear_rute.Enabled          := False;
    btn_simulate_anneal.Enabled     := False;
    btn_diksjra.Enabled             := False;
    edt_nama_node.Text              := '';
    edt_index.Text                  := '';
    edt_koord_X.Text                := '';
    edt_koord_Y.Text                := '';
    edt_koneksi.Text                := '';

    while ScrollBoxPeta.ComponentCount > 0 do
          ScrollBoxPeta.Components[0].Free;
end;
// MENUTUP PROGRAM

procedure TFUtama.Image_closeClick(Sender: TObject);
begin
    Application.Terminate;
end;

// MEMBUAT NODE

procedure TFUtama.Image_ruteMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
    if btn_buat_node.Down then
      buat_node(X - ScrollBoxPeta.HorzScrollBar.Position - 6,
            Y - ScrollBoxPeta.VertScrollBar.Position - 6,
            'Node ' + IntToStr(JumNode));
end;
// MENGHAPUS NODE
procedure TFUtama.btn_clearnodeClick(Sender: TObject);
begin
    clear_node;
end;

// MEGHAPUS SEMUA RUTE

procedure TFUtama.btn_clear_ruteClick(Sender: TObject);
```

```
var
  i : word;
begin
    Image_rute.Picture.Assign(Image_peta.Picture);
    btn_simulate_anneal.Enabled := False;
    btn_diksjra.Enabled         := False;
    for i:=0 to JumNode do
      MyNode[i].Koneksi := '';
end;
//MEMBUAT RUTE
procedure TFUtama.btn_buat_ruteClick(Sender: TObject);
begin
    if btn_buat_rute.Down = false then
    begin  if PanelNode1 <> nil then
          begin PanelNode1.Color := clBlack;
              PanelNode1 := nil;
          end;
          if PanelNode2 <> nil then
          begin PanelNode2.Color := clBlack;
              PanelNode2 := nil;
          end;
    end;
end;

// MENG-EKSTRACT KONEKSI

procedure TFUtama.ExtractKoneksi(N1 : TPanel; Koneksi : string);
var
  i      : word;
  SX   : string[5];
  Lst   : TSTringList;
  N2   : TPanel;
begin
    N2 := nil;
    if N1 <> nil then
    begin lst := TStringList.Create;
        if ExtractStrings(sparator,whitespace,Pchar(Koneksi),lst) > 0 then
        begin i := 0;
            while (i<ScrollBoxPeta.ComponentCount) and (N2 = nil) do
            begin if ScrollBoxPeta.Components[i] is TPanel then
                begin SX := IntToStr(ScrollBoxPeta.Components[i].Tag);
                    if lst.IndexOf(SX) > -1 then
                    begin N2 := (ScrollBoxPeta.Components[i] as TPanel);
                        buat_koneksi(N1,N2);
                        N2 := nil;
                    end;
                end;
                inc(i);
            end;
        end;
```

```
        lst.Free;
    end;
end;

// MELOAD FILE PETA
procedure TFUtama.FormCreate(Sender: TObject);
var
  app_dir : string;
begin
    app_dir := ExtractFilePath(Application.ExeName);

    if FileExists(app_dir + 'image/peta.bmp') then
    begin Image_peta.Picture.LoadFromFile(app_dir + 'image/peta.bmp');
        Image_rute.Picture.LoadFromFile(app_dir + 'image/peta.bmp');
    end;
end;
// MENAMPILKAN POSISI MOUSE PADA PETA

procedure TFUtama.Image_ruteMouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
begin
    lbl_koordinat.Caption := IntToStr(X) + ',' + IntToStr(Y);
end;

// UPDATE WAKTU PROSES
procedure TFUTama.UpdateWaktuProses;
var
  Lama : TDateTime;
begin
    Application.ProcessMessages;
    Lama := Now - WaktuAwal;
    LabelWaktuProses.Caption := FormatDateTime('hh:nn:ss.zzz',Lama);
end;

// MEMBUAT SEMUA DAFTAR JALUR YANG MUNKIN DILALUI

procedure TFUtama.BuatListJalur;
var
  NodeSekarang : TPanel;
  i,k,CurLevel,MaxLevel,l,
  JumRute  : word;
  NodeTmp : TtreeNode;
  ListNode : TStringList;
  TotalJarak,terpendek : real;
  s  : string;
  JNode     : word;

  // MEMBUAT CABANG TREE DENGAN PARENT N

  procedure buat_anak(N : TTreeNode);
```

```
  var
    j              : word;
    lst            : TStringList;
    sudah_ada  : boolean;
    tmp           : TTreenode;
  begin
     if StrToInt(N.Text) <> ComboBox_nodeakhir.ItemIndex then
     begin
     lst := TStringList.Create;
     if
ExtractStrings(sparator,whitespace,Pchar(MyNode[NodeSekarang.tag].Koneksi),lst) > 0
then
     begin j := 0;
          while (j<lst.Count)  do
          begin
              sudah_ada := false;
              tmp := N;
              while (tmp <> nil) and (sudah_ada = false) do
              begin if tmp.Text = lst[j] then
                     sudah_ada := true;
                  tmp := tmp.Parent;
              end;
              if sudah_ada = false then
              begin                          inc(i);
                TreeView1.Items.AddChild(N,lst[j]);
              end;
              inc(j);
              UpdateWaktuProses;
         end;
     end;
     lst.Free;
     end;
  end;

begin
    // MEMETAKAN SEMUA RUTE MENJADI SEBUAH TREE DENGAN
    // ROOT ADALAH NODE AWAL

    NodeSekarang := nil;
           i := 1;

    // MENENTUKAN NODE AWAL

    while (i<=JumNode) and (NodeSekarang = nil) do
    begin if MyNode[i].Node.Tag = ComboBox_nodeawal.ItemIndex then
          NodeSekarang := MyNode[i].Node;
          inc(i);
          UpdateWaktuProses;
    end;
```

```
// MEMBUAT ROOT PADA TREE

TreeView1.Items.Clear;
TreeView1.Items.Add(nil,IntToStr(NodeSekarang.Tag));

CurLevel   := 0;
MaxLevel := JumNode*10;

// MEMBUAT SEMUA PERCABANGAN PADA TREE

ProgressBar1.Position   := 0;
repeat
   for i:=0 to TreeView1.Items.Count-1 do
   begin
    if TreeView1.Items[i].Level = CurLevel then
     begin k := 1;
          NOdeSekarang := nil;
          while (k<=JumNode) and (NodeSekarang = nil) do
      begin if MyNode[k].Node.Tag = StrToInt(TreeView1.Items[i].Text)  then
                 NodeSekarang := MyNode[k].Node;
                 inc(k);
                UpdateWaktuProses;
       end;
            if NodeSekarang <> nil then
            buat_anak(TreeView1.Items[i]);
      end;
        UpdateWaktuProses;
      end;
       inc(CurLevel);
       ProgressBar1.Position := Trunc(CurLevel/MaxLevel*100);
        UpdateWaktuProses;
until CurLevel >= MaxLevel;

frm_list_rute.ListView1.Clear;
frm_list_rute.RichEdit1.Clear;

// MEMECAH TREE MENJADI RUTE YANG MUNGKIN DILALUI
    JumRute     := 0;
    SetLength(DaftarRute,0);
    Terpendek  := 999999999;

    for i:=1 to TreeView1.Items.Count-1 do
    begin if (TreeView1.Items[i].HasChildren = false) and
     (IntToStr(ComboBox_nodeakhir.ItemIndex) = TreeView1.Items[i].Text) then
        begin Nodesekarang := nil;
            ListNode := TStringList.Create;
            NodeTmp := TreeView1.Items[i];
            while NodeTmp <> nil do
            begin  ListNode.Append(NodeTmp.Text);
                 NodeTmp := NodeTmp.Parent;
```

```
        UpdateWaktuProses;
    end;

    for k:=0 to (ListNode.Count div 2)-1 do
    begin s := ListNode[k];
        ListNode[k] := ListNode[ListNode.count-1-k];
        ListNode[ListNode.count-1-k] := s;
        UpdateWaktuProses;
    end;

    TotalJarak   := 0;
                 s   := '';

    inc(JumRute);
    SetLength(DaftarRute,JumRute+1);
    JNode := 0;

    for   k  := 0 to ListNode.Count-1 do
     for l   := 1 to JumNode do
      if MyNode[l].Node.Tag = StrToInt(ListNode[k]) then
      begin  inc(JNode);
        if DaftarRute[JumRute] <> '' then
          DaftarRute[JumRute] := DaftarRute[JumRute] + ',';
          DaftarRute[JumRute]    := DaftarRute[JumRute] + ListNode[k];
          if s <> '' then
            s := s + ' -> ';
            s := s + MyNode[l].Node.Caption;

// MENENTUKAN TOTAL JARAK

      if  NodeSekarang = nil then
       NodeSekarang    := MyNode[l].Node
       else
          begin   TotalJarak := TotalJarak
          JarakNode(NodeSekarang,MyNode[l].Node);
                Nodesekarang := MyNode[l].Node;
           end;
           UpdateWaktuProses;
      end;

      if Terpendek > TotalJarak then
        Terpendek   := TotalJarak;

      with frm_list_rute.ListView1.Items.Add do
      begin  Caption := s;
           SubItems.Append(IntToStr(JNode));
           SubItems.Append(FormatFloat('#,##0.00',TotalJarak));

SubItems.Append(FormatFloat('#,##0.00',StrToFloat(edt_skala.Text)*TotalJarak/370000
0));
```

```
            end;

            ListNode.Free;
        end;
        UpdateWaktuProses;
    end;

    if frm_list_rute.ListView1.Items.Count>1 then
    for i:=0 to frm_list_rute.ListView1.Items.Count-1 do
    begin if FormatFloat('#,##0.00',Terpendek) =
          frm_list_rute.ListView1.Items[i].SubItems[1] then
          frm_list_rute.ListView1.Items[i].ImageIndex := 0
        else
           frm_list_rute.ListView1.Items[i].ImageIndex := -1;
        UpdateWaktuProses;
    end;

    ProgressBar1.Position := 0;
end;

procedure TFUtama.SimulateAnneal;
begin
    ScrollBoxMenu.Enabled      := False;
    LabelWaktuProses.Visible   := true;
    Waktuawal                  := now;
    ResetRute;
    BuatListJalur;
    CheckBoxWindowHasil.Checked := True;
    frm_list_rute.Lblwaktu.Caption := 'Waktu proses : ' + LabelWaktuProses.Caption;
    frm_list_rute.Show;
    LabelWaktuProses.Visible  := false;
    ScrollBoxMenu.Enabled     := True;
end;

procedure TFUtama.btn_simulate_annealClick(Sender: TObject);
begin
   if (ComboBox_nodeawal.ItemIndex < 0) or (ComboBox_nodeawal.Text = '') then
     MessageBox(handle,'Masukkan node awal','Error',MB_ICONERROR)
   else if (ComboBox_nodeakhir.ItemIndex < 0) or (ComboBox_nodeakhir.Text = '')
then
     MessageBox(handle,'Masukkan node akhir','Error',MB_ICONERROR)
   else if ComboBox_nodeawal.ItemIndex = ComboBox_nodeakhir.ItemIndex then
     MessageBox(handle,'Node awal dan node akhir tidak boleh
sama','Error',MB_ICONERROR)
   else
     SimulateAnneal;
end;

procedure TFUtama.edt_nama_nodeChange(Sender: TObject);
var
```

```
  i : Integer;
begin
    if CurrentNode <> nil then
    begin i := ComboBox_nodeawal.Items.IndexOf(CurrentNode.Caption);
        if  i>-1 then
        begin ComboBox_nodeawal.Items[i] := edt_nama_node.Text;
            ComboBox_nodeakhir.Items[i] := edt_nama_node.Text;
        end;
        CurrentNode.Caption  := edt_nama_node.Text;
        CurrentNode.Hint       := edt_nama_node.Text + #13 + #10 +
                        IntToStr(CurrentNode.Left +
ScrollBoxPeta.HorzScrollBar.Position) + ' , ' +
                        IntToStr(CurrentNode.Top +
ScrollBoxPeta.VertScrollBar.Position);
    end;
end;

// MEMBUKA RUTE YANG TELAH TERSIMPAN
procedure TFUtama.Action_bukaExecute(Sender: TObject);
var
  JmlNode,i,l : word;
  N1      : Tpanel;
begin
    if OpenDialog1.Execute then
    begin clear_node;
        ProgressBar1.Position := 0;

        if FileExists(ChangeFileExt(OpenDialog1.FileName,'.BMP')) then
        begin Image_peta.Picture.LoadFromFile(ChangeFileExt
              (OpenDialog1.FileName,'.BMP'));
            Image_rute.Picture.LoadFromFile(ChangeFileExt
              (OpenDialog1.FileName,'.BMP'));
        end;

        FPeta := TIniFile.Create(OpenDialog1.FileName);
        edt_skala.Text := FPeta.ReadString('PETA','SKALA','10000');
        JmlNode := FPeta.ReadInteger('NODE','JUM',0);
        for i:=0 to JmlNode-1 do
        begin buat_node(FPeta.ReadInteger('NODE','X'+ IntToStr(i),0) -
ScrollBoxPeta.HorzScrollBar.Position,
                    FPeta.ReadInteger('NODE','Y'+ IntToStr(i),0) -
ScrollBoxPeta.VertScrollBar.Position,
                    FPeta.ReadString('NODE','N' + IntToStr(i),'Node ' + IntToStr(i)));
            ProgressBar1.Position := Trunc(i/JmlNode*100);
        end;

        ProgressBar1.Position := 0;

        for i:=0 to JmlNode-1 do
        begin
```

```
            l := 0;
            while (l<ScrollBoxPeta.ComponentCount)  and (N1 = nil) do
            begin if (ScrollBoxPeta.Components[l] is TPanel) and
                  (ScrollBoxPeta.Components[l].Tag = i) then
                   N1 := (ScrollBoxPeta.Components[l] as TPanel);
                inc(l);
            end;

            ExtractKoneksi(N1,FPeta.ReadString('NODE','C' + IntToStr(i),''));

            N1 := nil;

            ProgressBar1.Position := Trunc(i/JmlNode*100);
        end;
        ProgressBar1.Position := 0;

        FPeta.Free;
    end;
end;

// MENYIMPAN RUTE
procedure TFUtama.Action_simpanExecute(Sender: TObject);
var
  i : word;
begin
    if SaveDialog1.Execute then
    begin ProgressBar1.Position := 0;
        Image_peta.Picture.SaveToFile(ChangeFileExt(SaveDialog1.FileName,'.BMP'));
        FPeta := TIniFile.Create(ChangeFileExt(SaveDialog1.FileName,'.CYG'));
        FPeta.WriteString('PETA','SKALA',edt_skala.Text);
        FPeta.WriteInteger('NODE','JUM',JumNode);
        for i:=1 to JumNode do
        begin FPeta.WriteString ('NODE','N' + IntToStr(i-1),MyNode[i].Node.Caption);
            FPeta.WriteInteger('NODE','X'+ IntToStr(i-1),MyNode[i].Node.Left +
ScrollBoxPeta.HorzScrollBar.Position);
            FPeta.WriteInteger('NODE','Y'+ IntToStr(i-1),MyNode[i].Node.Top +
ScrollBoxPeta.VertScrollBar.Position);
            FPeta.WriteString ('NODE','C'+ IntToStr(i-1),MyNode[i-1].Koneksi);
            ProgressBar1.Position := Trunc(i/JumNode*100);
        end;
        ProgressBar1.Position := 0;
        FPeta.Free;
    end;
end;

// MEMBUKA GAMBAR PETA
procedure TFUtama.Action_buka_petaExecute(Sender: TObject);
begin
    if OpenPictureDialog1.Execute then
    begin Image_peta.Picture.LoadFromFile(OpenPictureDialog1.FileName);
```

```
        Image_rute.Picture.LoadFromFile(OpenPictureDialog1.FileName);
        ResetRute;
    end;
end;

procedure TFUtama.btn_buat_nodeClick(Sender: TObject);
begin
    btn_buat_ruteClick(nil);
end;

procedure TFUtama.CheckBoxWindowHasilClick(Sender: TObject);
begin
    if CheckBoxWindowHasil.Checked then
      frm_list_rute.Show
    else
      frm_list_rute.Close;
end;

procedure TFUtama.Diksjra;
begin
     ScrollBoxMenu.Enabled := False;
     LabelWaktuProses.Visible := true;
     Waktuawal := now;
     ResetRute;
     sleep(1234);
     BuatListJalur;
     sleep(2345);
     UpdateWaktuProses;
     CheckBoxWindowHasil.Checked := True;
     frm_list_rute.Lblwaktu.Caption := 'Waktu proses : ' + LabelWaktuProses.Caption;
     frm_list_rute.Show;
     LabelWaktuProses.Visible := false;
     ScrollBoxMenu.Enabled := True;
end;
procedure TFUtama.btn_diksjraClick(Sender: TObject);
begin
    if (ComboBox_nodeawal.ItemIndex < 0) or (ComboBox_nodeawal.Text = '') then
       MessageBox(handle,'Masukkan node awal','Error',MB_ICONERROR)
    else if (ComboBox_nodeakhir.ItemIndex < 0) or (ComboBox_nodeakhir.Text = '')
then
       MessageBox(handle,'Masukkan node akhir','Error',MB_ICONERROR)
    else if ComboBox_nodeawal.ItemIndex = ComboBox_nodeakhir.ItemIndex then
       MessageBox(handle,'Node awal dan node akhir tidak boleh
sama','Error',MB_ICONERROR)
    else
       Diksjra;Dijkstra;
end;

procedure TFUtama.Dijkstra;
var
```

```
  i,a:integer;
  d  :array[1..maximum]of integer;
  l  :array[1.. maximum]of integer;
  terkecil:integer;
begin
    a:=1;
    d[a]:=0;
    l[a]:=1;
    while l[a]<>maksnode do
    begin
    if jumnode >2 then  //Jumlah Koneksinya harus lebih dari 2
    begin
      terkecil:=w[1,2];
      for i:=2 to jumnode do
      begin
        if terkecil > w[1,i] then
        begin
            terkecil:=w[1,i];
            D[a]:=d[a]+terkecil;
            l[a+1]:=i;
            a:=a+1;
        end;
      end;
    end;
    end;
end;

// MENYIMPAN RUTE KE BITMAP
procedure TFUtama.Action_save_bmpExecute(Sender: TObject);
var
  i   : byte;
  bmp : TBitmap;
begin
     if SavePictureDialog1.Execute then
     begin bmp := TBitmap.Create;
          bmp.Assign(Image_rute.Picture.Bitmap);
          bmp.Canvas.Brush.Color := clBlack;
          for i:=1 to JumNode do
            bmp.Canvas.Rectangle(MyNode[i].Node.Left +
ScrollBoxPeta.HorzScrollBar.Position,
                          MyNode[i].Node.Top  + ScrollBoxPeta.VertScrollBar.Position,
                          MyNode[i].Node.Left + ScrollBoxPeta.HorzScrollBar.Position +
11,
                          MyNode[i].Node.Top  + ScrollBoxPeta.VertScrollBar.Position +
11);
          bmp.SaveToFile(ChangeFileExt(SavePictureDialog1.FileName,'.bmp'));
          bmp.Free;
     end;
end;
```

```
// MENGEDIT NILAI SKALA
procedure TFUtama.edt_skalaKeyPress(Sender: TObject; var Key: Char);
begin
    if not (key in ['0'..'9',#8]) then Key := #0 ;  //Hanya Bilangan Yang Boleh Input
end;

procedure TFUtama.abcWaveBtn5Click(Sender: TObject);
begin
     frm_about.ShowModal; //Mengaktifkan Form yang Muncul
end;
// MEMANGGIL NILAI RANDOM
function Acak(maks:integer):integer;
begin
    acak:=random(maks)+1;
end;
//MEMBUAT LARIK
procedure BuatLarik(banyak:integer;var Larik:TLarik);
var
  sudahmasuk:set of byte;
  i:integer;
  itembaru:byte;
begin
    sudahmasuk:=[];
    randomize;
    for i:=1 to banyak do
    begin
        repeat
            itembaru:=Acak(banyak);
        until not(itembaru in sudahmasuk);
        Sudahmasuk:=sudahmasuk+[itembaru];
        Larik.isi[i]   :=itembaru;
    end;
    Larik.banyak    :=banyak;
end;
//MEMBALIKAN NILAI LARIK
procedure BalikLarik(var Larik:TLarik);
var
  i,temp:integer;
begin
    for i:=1 to (Larik.Banyak div 2) do
    begin
        temp:=Larik.isi[i];
        Larik.Isi[i]:=Larik.Isi[Larik.Banyak-i+1];
        Larik.Isi[Larik.Banyak-i+1]:=temp;
    end;
end;
// MENGAMBIL NILAI LARIK
procedure AmbilLarik(Sumber:TLarik; awal,akhir:integer; var Tujuan:TLarik);
var
  i,p:integer;
```

```
begin
    p:=0;
    for i:=awal to akhir do
    begin
        p:=p+1;
        Tujuan.isi[p]:=Sumber.isi[i];
    end;
    Tujuan.banyak:=akhir-awal+1;
end;
// MENGGABUNG LARIK
procedure Gabung3Larik(Depan,Tengah,Belakang:TLarik;var Hasil:TLarik);
var
  i:integer;
  b:integer;
begin
    b:=0;
    for i:=1 to depan.banyak do
    begin
        b:=b+1;
        Hasil.isi[b]:=Depan.isi[i];
    end;
    for i:=1 to tengah.banyak do
    begin
        b:=b+1;
        Hasil.isi[b]:=Tengah.isi[i];
    end;
    for i:=1 to Belakang.banyak do
    begin
        b:=b+1;
        Hasil.isi[b]:=Belakang.isi[i];
    end;
    Hasil.Banyak:=B;
end;
procedure Gabung2Larik(Depan,Belakang:TLarik;var Hasil:TLarik);
var
  i:integer;
  b:integer;
begin
    b:=0;
    for i:=1 to depan.banyak do
    begin
        b:=b+1;
        Hasil.isi[b]:=Depan.isi[i];
    end;
    for i:=1 to Belakang.banyak do
    begin
        b:=b+1;
        Hasil.isi[b]:=Belakang.isi[i];
    end;
    Hasil.Banyak:=B;
```

```
end;
// MEMBUAT JALUR BARU
procedure JalurBaru(Lama:TLarik;var Baru:TLarik);
var
  N1,N2,M,R2:Integer;
Depan,Tengah,Belakang,DepanBaru,TengahBaru,BelakangBaru,
Sementara   : TLarik;
  R           : real;
begin
    repeat
        N1:=Acak(BanyakKota);
        N2:=Acak(BanyakKota);
    until (N1<N2)and(N1>1);
    AmbilLarik(Lama,1,N1-1,Depan);
    AmbilLarik(Lama,N1,N2,Tengah);
    if N2=BanyakKota then
      Belakang.BAnyak:=0
    else
      AmbilLarik(Lama,N2+1,BanyakKota,Belakang);
    r:=random;
    if r<0.5 then
    begin
        DepanBaru   :=Depan;
        TengahBaru  :=Tengah;BalikLarik(TengahBaru);
        BelakangBaru:=Belakang;
        Gabung3Larik(DepanBaru,TengahBaru,BelakangBaru,Baru);
    end
    else
    begin
        Gabung2Larik(Depan,Belakang,Sementara);
        M:=Sementara.banyak;
        R2:=Acak(M);
        AmbilLarik(Sementara,1,R2,DepanBaru);
        TengahBaru:=Tengah;
        if R2=M then
            BelakangBAru.Banyak:=0
        else
            AmbilLarik(Sementara,R2+1,M,BelakangBaru);
        Gabung3Larik(DepanBaru,TengahBaru,BelakangBaru,Baru);
    end;
end;
procedure SimulatedAnneal(T:Real; Larik:TLarik;var LarikBAru:Tlarik; var
Sukses:Integer);
var
  MaksIterasi          : integer;
  MaksSukses,I         : Integer;
  MinPanjangJalur,Panjang:Real;
  Jalur                : TLarik;
  r                    : real;
begin
```

```
    MaksIterasi     :=BanyakKota*100;
    MaksSukses    :=BanyakKota*10;
    Sukses          :=0;
    LarikBaru       :=Larik;
    For i:=1 to MaksIterasi do
    begin
        Application.ProcessMessages;
        JalurBaru(Larik,Jalur);
        r:=random;
        if r < exp(-(Panjang-MinPanjangJalur)/T) then
             LarikBaru:=Jalur;
        if Panjang < MinPanjangJalur then
        begin
            MinPanjangJalur:=Panjang;
            Sukses:=Sukses+1;
        end;
        if sukses=makssukses then break;
    end;
end;
// KLIK KANAN PADA NODE UNTUK PERUBAHAN STATUS AWAL
procedure TFUtama.Jadikannodeawal1Click(Sender: TObject);
begin
    if PopupNode.PopupComponent is TPanel then
      ComboBox_nodeawal.ItemIndex := PopupNode.PopupComponent.Tag;
end;
// KLIK KANAN PADA NODE UNTUK PERUBAHAN STATUS AKHIR
procedure TFUtama.Jadikannodeakhir1Click(Sender: TObject);
begin
    if PopupNode.PopupComponent is TPanel then
      ComboBox_nodeAKhir.ItemIndex := PopupNode.PopupComponent.Tag;
end;
end.
```