DAFTAR PUSTAKA

1. Alam, A, J : *Borland Delphi 6.0*, Elex Media Computindo, Jakarta, 2001.
2. Chartrand, G dan Ortrad R, O : *Applied and Algorithmic Graph Theory*, Mc.Graw-Hill, Inc.
3. Deo Narsingh : *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall of India.
4. Kadir, A : *Dasar Pemrograman Delphi 5.0*, Andi Offset, Yogyakarta, 2001.
5. Martina, I, Ir. : *Delphi 5.0*, Elex Media Computindo, Jakarta,2000
6. Munir, R, Ir : *Buku Teks Ilmu Komputer Matematika Diskret*, Informatika Bandung, 2002.

Lampiran 1 : Program Utama

```
program PPathCounter;

uses
  Forms,
  PathCounter in 'PathCounter.pas'{Path},
  VarGlobal in 'VarGlobal.pas',
  AboutBx in 'AboutBx.pas'{AboutBox},
  EditPosisi in 'EditPosisi.pas'{FormEditPosisi},
  Help in 'Help.pas'{FHelp};

{$R *.RES}

Begin

  Application.Initialize;
  Application.CreateForm(TPath,Path);
  Application.CreateForm(TAboutBox,AboutBox);
  Application.CreateForm(TFormEditPosisi,FormEditPosisi);
  Application.CreateForm(TFHelp,FHelp);
  Application.Run;

end.
```

Lampiran 2 : Unit PathCounter

```
unit PathCounter;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms,Dialogs,Menus,ComCtrls,Grids,StdCtrls,Spin,VarGlobal
  Buttons, ExtDlgs;

type
   Tpath = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    New: TMenuItem;
    N: TMenuItem;
    Proses: TMenuItem;
    Run: TMenuItem;
    BuatGraph: TMenuItem;
    Help: TMenuItem;
    About: TMenuItem;
    PageControl1: TPageControl;
    TabSheetInput: TTabSheet;
    TabSheetOutput: TTabSheet;
    TabSheetPathPath: TTabSheet;
    TabSheetElementer: TTabSheet;
    StringGridInput: TStringGrid;
    Label1: TLabel;
    SpinEditAsal: TSpinEdit;
    Label2: TLabel;
    SpinEditTujuan: TSpinEdit;
    Label3: TLabel;
    SpinEditWalk: TSpinEdit;
    ButtonNext: TButton;
    Label4: TLabel;
    EWalk: TEdit;
    Label5: TLabel;
    EJPath: TEdit;
    MemoPath: TMemo;
    ButtonKlik1: TButton;
    Label6: TLabel;
    EWalki: TEdit;
    Label7: TLabel;
    EBPath: TEdit;
    MemoElementer: TMemo;
    ButtonKlik2: TButton;
    Label8: TLabel;
    EElemen: TEdit;
    SaveDialogPath: TSaveDialog;
```

```
    OpenDialogPath: TOpenDialog;
    StringGridOutput: TStringGrid;
    Exit: TMenuItem;
    SpeedButtonOK: TSpeedButton;
    SpeedButtonKeluar: TSpeedButton;
    Label9: TLabel;
    Label10: TLabel;
    procedure ExitClick(Sender: TObject);
    procedure AboutClick(Sender: TObject);
    procedure StringGridInputKeyUp(Sender: TObject; var Key:
                                   Word;Shift: TShiftState);
    procedure BitBtn1Click(Sender: TObject);
    procedure RefreshStringGrid;
    procedure RefreshOutput(Hasil:Matriks);
    procedure NewClick(Sender: TObject);
    procedure RunClick(Sender: TObject);
    procedure ButtonNextClick(Sender: TObject);
    procedure SpinEditAsalChange(Sender: TObject);
    procedure SpinEditTujuanChange(Sender: TObject);
    procedure SpinEditWalkChange(Sender: TObject);
    procedure ButtonKlik1Click(Sender: TObject);
    procedure ButtonKlik2Click(Sender: TObject);
    procedure BuatGraphClick(Sender: TObject);
    procedure   FormClose(Sender:   TObject;   var   Action:
                          TCloseAction);
    procedure SpeedButtonOKClick(Sender: TObject);
    procedure SpeedButtonKeluarClick(Sender: TObject);
    procedure Help1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Path: TPath;
  Counter:Integer;

implementation

uses AboutBx, EditPosisi,Help ;
{$R *.DFM}

procedure TPath.ExitClick(Sender: TObject);
begin
  Close;
end;
```

```
procedure TPath.AboutClick(Sender: TObject);
begin
  AboutBox.ShowModal;
end;

procedure TPath.BitBtn1Click(Sender: TObject);
begin
  StringGridInput.Rowcount:=Ordo+1;
  StringGridInput.Colcount:=Ordo+1;
  RefreshStringGrid;
end;

procedure TPath.RefreshStringGrid;
var i,j :integer;
begin
  StringGridInput.Rowcount:=Ordo+1;
  StringGridInput.Colcount:=Ordo+1;
  StringGridInput.Cells [0,0]:='Vertex';
  for i:=1 to ordo do
  begin
    StringGridInput.Cells [0,i]:=IntToStr(i);
    StringGridInput.Cells [i,0]:=IntToStr(i);
  end;
  for i:= 1 to ordo do
    for j:= 1 to ordo do
      StringGridInput.Cells[j,i]:=IntToStr(Mtr[i,j]);
  StringGridInput.Refresh;
  SpinEditAsal.Value:=Asal;
  SpinEditTujuan.Value:=Tujuan;
  SpinEditWalk.Value:=Walk;
  EWalk.Text:='1';
  EJPath.Text:=IntToStr(Mtr[Asal,Tujuan]);
  Label5.Caption:='JumlahPath
  ('+IntToStr(Asal)+','+IntToStr(Tujuan)+')';
end;

procedure TPath.RefreshOutput(Hasil:Matriks);
var i,j:integer;
begin
  StringGridOutput.Rowcount:=Ordo+1;
  StringGridOutput.Colcount:=Ordo+1;
  StringGridOutput.Cells [0,0]:='Vertex';
  for i:=1 to ordo do
  begin
    StringGridOutput.Cells [0,i]:=IntToStr(i);
    StringGridOutput.Cells [i,0]:=IntToStr(i);
  end;
  for i:= 1 to ordo do
    for j:= 1 to ordo do
      StringGridOutput.Cells[j,i]:=IntToStr(Hasil[i,j]);
  StringGridOutput.Refresh;
```

```
end;

procedure  TPath.StringGridInputKeyUp(Sender:  TObject;  var
           Key: Word;Shift: TShiftState);
var i,j : integer;
begin
  i := StringGridInput.Col;
  j := StringGridInput.Row;
  if i=j then
    StringGridInput.Cells[i,j]:='0'
  else
  if (StringGridInput.Cells[i,j]='0') or
     (StringGridInput.Cells[i,j]='1') then
     StringGridInput.Cells[j,i]:= StringGridInput.Cells[i,j]
  else
     StringGridInput.Cells[i,j]:=StringGridInput.Cells[j,i];
  Mtr[i,j]:=StrToInt(StringGridInput.Cells[j,i]);
  Mtr[j,i]:=StrToInt(StringGridInput.Cells[i,j]);
end;

procedure TPath.NewClick(Sender: TObject);
var i,j:integer;
begin
  PageControl1.ActivePage:=TabSheetInput;
  ordo:=5;
  For i:=1 to ordo do
    For j:=1 to ordo do
      Mtr[i,j]:=0;
  BantuNumSimpul:=0;
  Asal:=1;
  Tujuan:=1;
  Walk:=1;
  EWalk.Text:=IntToStr(Kali);
  EJPath.Text:=IntToStr(Mtr[asal,tujuan]);
  Label5.Caption:='Jumlah
  Path['+IntToStr(asal)+','+IntToStr(tujuan)+']';
  RefreshOutput(Mtr);
  RefreshStringGrid;
  MemoPath.Clear;
  MemoElementer.Clear;
  EWalki.Text:='';
  EBPath.Text:='';
  EElemen.Text:='';
  FormEditPosisi.ImageGraph.Canvas.Rectangle(0,0,
  FormEditPosisi.ImageGraph.Picture.Width,
  FormEditPosisi.ImageGraph.Picture.Height);
end;
```

```
procedure Kalikan_Mtr(Var m:Integer; Var MKali:Matriks; Var
                      MatA:Matriks; Var MatB:Matriks);
Var i,j,k:integer;
begin
  For i:=1 to m do
    For j:=1 to m do
    begin
      MKali [i,j]:=0;
      For k:=1 to m do
        MKali [i,j]:=MKali[i,j]+MatA[i,k]*MatB[k,j]
    end;
end;

procedure TPath.RunClick(Sender: TObject);
Var asal,tujuan:integer;
begin
  PageControl1.ActivePage:=TabSheetOutput;
  Mtr1:=Mtr;
  Kali:=1;
  B:=Mtr1;
  Asal:=SpinEditAsal.Value;
  Tujuan:=SpinEditTujuan.Value;
  EJpath.Text:=IntToStr(Mtr1[asal,tujuan]);
  MemoPath.Clear;
  MemoElementer.Clear;
  EWalki.Text:='';
  EBPath.Text:='' ;
  RefreshStringGrid;
  RefreshOutPut(Mtr1);
end;

procedure TPath.ButtonNextClick(Sender: TObject);
var asal,tujuan,walk:integer;
begin
  asal:=SpinEditAsal.value;
  tujuan:=SpinEditTujuan.value;
  walk:=SpinEditWalk.value;
  if kali<walk then
  begin
    RefreshStringGrid;
    kali:=kali+1;
    Kalikan_Mtr(Ordo,A2,Mtr1,B);
    Mtr1:=A2;
    RefreshOutput(A2);
    EJPath.Text:=IntToStr(A2[asal,tujuan]);
    EWalk.Text:=IntToStr(kali);
    Label5.Caption:='Jumlah
    Path('+IntToStr(asal)+','+IntToStr(Tujuan)+')';
   end;
end;
```

```
procedure TPath.SpinEditAsalChange(Sender: TObject);
var Ed1:integer;
begin
  Ed1:=SpinEditAsal.Value;
  if(Ed1<ordo)then
    Asal:=Ed1
  else Asal:=ordo;
end;

procedure TPath.SpinEditTujuanChange(Sender: TObject);
var Ed2:integer;
begin
  Ed2:=SpinEditTujuan.Value;
  if(Ed2<ordo)then
    Tujuan:=Ed2
  else Tujuan:=ordo;
end;

procedure TPath.SpinEditWalkChange(Sender: TObject);
begin
  Walk:=SpinEditWalk.Value
end;

procedure TPath.ButtonKlik1Click(Sender: TObject);
var jmlpath:integer;
    path     :vektor;

Function
Rwalk(ordo,i,walk,asal,tujuan:integer;path:vektor):integer;
var sum,j:integer;
    S    :String;
begin
  if i=walk then
  begin
    if Mtr[Asal,Tujuan]=1
    then
    begin
      RWalk:=1;
      path[i]:=Asal;
      Inc(Counter);
      S:=(IntToStr(Counter)+':');
      for j:=1 to walk do S:=S+IntToStr(path[j])+'';
      if Counter mod 1000=0 then MemoPath.Lines.Clear;
      MemoPath.Lines.Add(S+IntToStr(Tujuan));
    end
    else RWalk:=0;
  end
  else
  begin
    Sum:=0;
    for j:=1 to ordo do
```

```
        if Mtr[Asal,j]=1 then
        begin
          Path[i]:=Asal;
          Sum:=Sum+RWalk(ordo,i+1,Walk,j,Tujuan,path);
        end;
      RWalk:=Sum;
    end;
  end;

  begin
    Counter:=0;
    MemoPath.Clear;
    Asal:=SpinEditAsal.Value;
    Tujuan:=SpinEditTujuan.Value;
    Walk:=SpinEditWalk.Value;
    MemoPath.Lines.Add('PATH-PATHNYA ADALAH:');
    JmlPath:=RWalk(ordo,1,Walk,Asal,Tujuan,Path);
    EWalki.Text:=IntToStr(Walk);
    EBPath.Text:=IntToStr(JmlPath);
    if JmlPath=0 then
    begin
      MemoPath.Clear;
      MemoPath.Lines.Add('PATHNYA TIDAK ADA');
    end;
    Label5.Caption:='Jumlah
    Path['+IntToStr(Asal)+','+IntToStr(Tujuan)+']=';
  end;

  procedure TPath.ButtonKlik2Click(Sender: TObject);
  Var
      Counter2,jmlPath2 : integer;
      Path:vektor;
      nomer:string;

  Function
  RWalkNoReEntrant(ordo,i,walk,asal,tujuan:integer;
                   path:vektor):integer;
  Function NoDuplicate(path:vektor):Boolean;
  Var Duplicate:boolean;
      Kunjung:vektor;
      i,j:integer;
  begin
    For i:=1 to ordo do Kunjung[i]:=0;
    For j:=1 to Walk do Inc(Kunjung[Path[j]]);
    Duplicate:=False;
    For i:= 1 to ordo do
      For j := 1 to walk do
        if (kunjung[i]>=2)or(Path[j]=tujuan) then
          Duplicate:=True;
    NoDuplicate:=Not(Duplicate);
  end;
```

```
    Var sum,j:integer;
        S    :String;
    begin
      if i=Walk then
      begin
        if Mtr[Asal,Tujuan]=1
        then
        begin
          RWalkNoReEntrant:=1;
          path[i]:=asal;
          Inc(Counter);
          S:=(IntToStr(Counter)+':');
          for j:=1 to Walk do S:=S+IntToStr(path[j])+'';
          if NoDuplicate(path) then
          begin
            Inc(Counter2);
            Nomer:=(IntToStr(Counter2)+':'+'Path ke-');
            MemoElementer.Lines.Add(Nomer+S+IntToStr(Tujuan));
            if Counter2 mod 1000=0
            then
            MemoElementer.Lines.Clear;
          end;
        end
        else RWalkNoReEntrant:=0;
      end
      else
      begin
        Sum:=0;
        for j:=1 to ordo do
          if Mtr[Asal,j]=1 then
          begin
            Path[i]:=Asal;
            Sum:=Sum+RWalkNoReEntrant(ordo,i+1,walk,j,tujuan,
            path);
          end;
          RWalkNoReEntrant:=Sum;
    end;
  end;
  begin
    Counter:=0;
    Counter2:=0;
    MemoElementer.Clear;
    Asal:=SpinEditAsal.Value;
    Tujuan:=SpinEditTujuan.Value;
    Walk:=SpinEditWalk.Value;
    MemoElementer.Lines.Add('PATH-PATH      YANG      ELEMENTER
                             ADALAH:');
    JmlPath2:=RWalkNoReEntrant(ordo,1,Walk,Asal,Tujuan,Path);
    EElemen.Text:=IntToStr(Counter2);
    if counter2=0 then
    begin
```

```
      if counter=0 then
      begin
        MemoElementer.Clear;
        MemoElementer.Lines.Add('TIDAK ADA PATH-NYA');
      end
      else
      begin
          MemoElementer.Clear;
          MemoElementer.Lines.Add('TIDAK   ADA   PATH   YANG
          ELEMENTER');
      end;
    end;
  end;

  procedure TPath.BuatGraphClick(Sender: TObject);
  begin
    FormEditPosisi.ShowModal;
    RefreshStringGrid;
    PageControl1.ActivePage:=TabSheetInput;
  end;

  procedure  TPath.FormClose(Sender:  TObject;  var  Action:
  TCloseAction);
  var TombolTanggapan:Word;
  begin
    TombolTanggapan:=MessageDlg('Anda   Ingin   Menutup   PATH
    COUNTER ini ?',mtConfirmation,[mbYES,mbNO],0);
    if TombolTanggapan=mrYes
    then
      Action:=CaFree
    else
      Action:=CaNone;
  end;

  procedure TPath.SpeedButtonOKClick(Sender: TObject);
  Var asal,tujuan:integer;
  begin
    PageControl1.ActivePage:=TabSheetOutput;
    Mtr1:=Mtr;
    Kali:=1;
    B:=Mtr1;
    Asal:=SpinEditAsal.Value;
    Tujuan:=SpinEditTujuan.Value;
    EJPath.Text:=IntToStr(Mtr1[asal,tujuan]);
    MemoPath.Clear;
    MemoElementer.Clear;
    EWalk.Text:='';
    EBPath.Text:='' ;
    RefreshStringGrid;
    RefreshOutPut(Mtr1);
  end;
```

```
procedure TPath.SpeedButtonKeluarClick(Sender: TObject);
begin
  Close;
end;

procedure TPath.Help1Click(Sender: TObject);
begin
  FHelp.ShowModal;
end;

end.
```

Lampiran 3: Unit VarGlobal

```
unit VarGlobal;

interface
TYPE Matriks = ARRAY [1..100,1..100] of integer;
  Type TipeVektor=array[1..100] of integer;
ArraySimpul=Array[1..100] of
                record
                  NamaSimpul :String;
                  PosX,PosY :integer;
                end;
        Vektor=Array[1..100] of integer;

VAR Mtr,Mtr1,A2,B,MPath : Matriks;
    m,n,i,j,k,x,y : Integer;
    Vp : Vektor;

VAR FileName : string;
        f    : Text;
        ordo,asal,jmlpath,tujuan,walk : integer;
        Kali : Integer;
        BantuXAsli : Matriks;
        BantuNumSimpul : Integer;
        BantuSimpul : ArraySimpul;
        BantuBackGround : String;
implementation

Begin
  Ordo:=5;
end.
```

Lampiran 4 : Unit EditPosisi

```
unit EditPosisi;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls,
  Forms,   Dialogs,StdCtrls,   Buttons,   ExtCtrls,VarGlobal,
  ExtDlgs;

type
  TFormEditPosisi = class(TForm)
    ImageGraph: TImage;
    GroupBoxEdit: TGroupBox;
    RadioButtonVertex: TRadioButton;
    RadioButtonEdge: TRadioButton;
    Panel1: TPanel;
    BitBtnTerima: TBitBtn;
    BitBtnBatal: TBitBtn;
    FunctionTermasukSimpul(VarX,Y,SimpulKe:Integer):Boolean;
    FunctionTermasukJalan(Var
            X,Y,SimpulAsal,SimpulTujuan:Integer):Boolean;
    procedure  ImageGraphMouseDown(Sender:  TObject;  Button:
            TMouseButton;Shift: TShiftState;X, Y: Integer);
    procedure  ImageGraphMouseUp(Sender:  TObject;  Button:
            TMouseButton;Shift:TShiftState; X, Y: Integer);
    procedure FormShow(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure  ImageGraphMouseMove(Sender:  TObject;  Shift:
            TShiftState; X,Y: Integer);
    procedure FormResize(Sender: TObject);
    procedure BitBtnTerimaClick(Sender: TObject);


  private
    { Private declarations }
    Accept:Boolean;
  public
    { Public declarations }
  end;

var
  FormEditPosisi: TFormEditPosisi;

implementation
  Var SimpulAsal,SimpulTujuan:integer;
{$R *.DFM}
```

```
Function TFormEditPosisi.TermasukSimpul
         (var X,Y,SimpulKe:Integer):Boolean;
var Hasil:Boolean;
  i:Integer;
begin
  Hasil:=False;
  for i:=1 to BantuNumSimpul do
  begin
    if(BantuSimpul[i].PosX-3<=X)and
    (X<=BantuSimpul[i].PosX+3) and
    (BantuSimpul[i].PosY-3<=Y)and (Y<=BantuSimpul[i].PosY+3)
    then
    begin
      Simpulke:=i;
      X:=BantuSimpul[i].PosX;
      Y:=BantuSimpul[i].PosY;
      Hasil:=True;
    end;
  end;
  TermasukSimpul:=Hasil;
end;

Function TFormEditPosisi.TermasukJalan
         (var X,Y,SimpulAsal,SimpulTujuan:Integer):Boolean;
var Hasil:Boolean;
  i,j:Integer;
  m,c:real;
begin
  Hasil:=False;
  for i:=1 to BantuNumSimpul-1 do
    for j:=i+1 to BantuNumSimpul do
      if (BantuSimpul[i].PosX=BantuSimpul[j].PosX) then
      begin
        if(BantuSimpul[i].PosX-3<=X)and
        (X<=BantuSimpul[i].PosX+3) and
        ((BantuSimpul[i].PosY<Y) and (Y<BantuSimpul[j].PosY)
        or
        ((BantuSimpul[j].PosY<Y)and(Y<BantuSimpul[i].PosY)))
        then
        if  (BantuXAsli[i,j]>0)  and  (BantuXAsli[i,j]<9999)
        then
        begin
          SimpulAsal:=i;
          SimpulTujuan:=j;
          Hasil:=True;
        end;
      end
      else
      begin
```

```
            m:=(BantuSimpul[j].PosY-
            BantuSimpul[i].PosY)/(BantuSimpul[j].PosX-
            BantuSimpul[i].PosX);
            c:=BantuSimpul[j].PosY-m*BantuSimpul[j].PosX;
            if (m*X+c-3<=Y) and (Y<=m*X+c+3) and
               (
                ((BantuSimpul[i].PosY<Y)and
                (Y<BantuSimpul[j].PosY))
               or
                ((BantuSimpul[j].PosY<Y)and
                (Y<BantuSimpul[i].PosY))
                )
               and
               (
               ((BantuSimpul[i].PosX<X)and
                (X<BantuSimpul[j].PosX))
               or
               ((BantuSimpul[j].PosX<X)and
                (X<BantuSimpul[i].PosX))
                )
               then
               if (BantuXAsli[i,j]>0) and (BantuXAsli[i,j]<9999)
               then
               begin
                 SimpulAsal:=i;
                 SimpulTujuan:=j;
                 Hasil:=True;
               end;
         end;
         TermasukJalan:=Hasil;
end;

procedure TFormEditPosisi.ImageGraphMouseDown
          (Sender:  TObject;Button:  TMouseButton;  Shift:
          TShiftState; X, Y: Integer);
var i:integer;
    simpulke:integer;
    LebarX,LebarY:Integer;

Procedure BeriNamaSimpul;
var SimpulSementara:Integer;

Function AdaSimpulBernama(Nama:String):Boolean;
var i:integer;
begin
  Result:=False;
  for i:=1 to BantuNumSimpul do
    if Nama=BantuSimpul[i].NamaSimpul then
      Result:=True;
end;
```

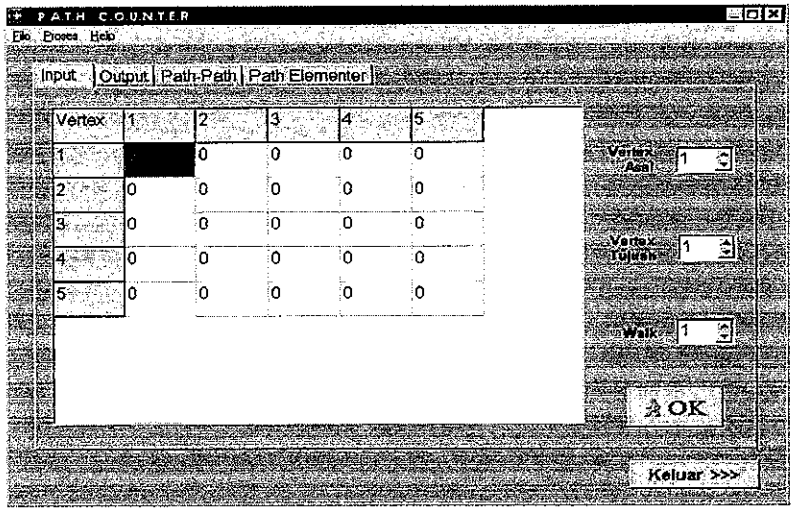
```
begin
  SimpulSementara:=BantuNumSimpul;
  While    AdaSimpulBernama(IntToStr(SimpulSementara))    do
  inc(SimpulSementara);
  BantuSimpul[BantuNumSimpul].NamaSimpul:=IntToStr(Simpul
  Sementara);
end;

begin
  X:=Round(X*ImageGraph.Picture.Width/ImageGraph.Width);
  Y:=Round(Y*ImageGraph.Picture.Height/ImageGraph.Height);
  LebarX:=Round(ImageGraph.Picture.Width/ImageGraph.Width);
  LebarY:=Round(ImageGraph.Picture.Height/ImageGraph.Height)
  if LebarX>LebarY then
    ImageGraph.Canvas.Pen.Width:=LebarX
  else
    ImageGraph.Canvas.Pen.Width:=LebarY;
  if Button=mbLeft then
  begin
    if RadioButtonVertex.Checked then
    begin
      if Not(TermasukSimpul(X,Y,SimpulKe)) then
      begin
        ImageGraph.Canvas.Ellipse(X-4*LebarX,Y-
        4*LebarY,X+4*LebarX,Y+4*LebarY);
        Inc(BantuNumSimpul);
        for i:=1 to BantuNumSimpul do
        begin
          BantuXAsli[i,BantuNumSimpul]:=9999;
          BantuXAsli[BantuNumSimpul,i]:=9999;
        end;
        BantuSimpul[BantuNumSimpul].PosX:=X;
        BantuSimpul[BantuNumSimpul].PosY:=Y;
        BeriNamaSimpul;
      end;
    end
    else if RadioButtonEdge.Checked then
    begin
      if TermasukSimpul(X,Y,SimpulAsal)then
      begin
        ImageGraph.Canvas.MoveTo(X,Y);
        Accept:=True;
      end;
    end
  end
  else if Button=mbRight then
  begin
    if RadioButtonVertex.Checked then
    begin
      if TermasukSimpul(X,Y,SimpulAsal) then
      begin
```

```
          ImageGraph.Canvas.Pen.Color:=clRed;
          ImageGraph.Canvas.MoveTo(X-4*LebarX,Y-4*LebarY);
          ImageGraph.Canvas.LineTo(X+4*LebarX,Y+4*LebarY);
          ImageGraph.Canvas.MoveTo(X-4*LebarX,Y+4*LebarY);
          ImageGraph.Canvas.LineTo(X+4*LebarX,Y-4*LebarY);
          ImageGraph.Canvas.Pen.Color:=clBlack;
          for i:=1 to BantuNumSimpul do
          begin
            BantuSimpul[SimpulAsal]:=BantuSimpul[BantuNum
            Simpul];
            BantuXAsli[SimpulAsal,i]:=BantuXAsli[BantuNum
            Simpul,i];
            BantuXAsli[i,SimpulAsal]:=BantuXAsli[i,BantuNum
            Simpul];
          end;
          dec(BantuNumSimpul);
        end;
      end
      else if RadioButtonEdge.Checked then
      begin
        if TermasukJalan(X,Y,SimpulAsal,SimpulTujuan) then
        begin
          ImageGraph.Canvas.Pen.Color:=clRed;
          ImageGraph.Canvas.MoveTo(BantuSimpul[SimpulAsal].
          PosX,BantuSimpul[SimpulAsal].PosY);
          ImageGraph.Canvas.LineTo(BantuSimpul[SimpulTujuan].
          PosX,BantuSimpul[SimpulTujuan].PosY);
          BantuXAsli[SimpulAsal,SimpulTujuan]:=9999;
          BantuXAsli[SimpulTujuan,SimpulAsal]:=9999;
          ImageGraph.Canvas.Pen.Color:=clBlack;
        end;
      end;
    end;
end;

procedure TFormEditPosisi.ImageGraphMouseUp(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
   X:=Round(X*ImageGraph.Picture.Width/ImageGraph.Width);
   Y:=Round(Y*ImageGraph.Picture.Height/ImageGraph.Height);
   if Button=mbLeft then
   begin
     if RadioButtonEdge.Checked then
     begin
       if TermasukSimpul(X,Y,SimpulTujuan) and Accept then
       begin
         ImageGraph.Canvas.LineTo(X,Y);
         BantuXAsli[SimpulAsal,SimpulTujuan]:=round(sqrt(
          sqr(BantuSimpul[SimpulAsal].PosX-BantuSimpul
          [SimpulTujuan].PosX)+sqr(BantuSimpul[SimpulAsal].
          PosY-BantuSimpul[SimpulTujuan].PosY)));
```

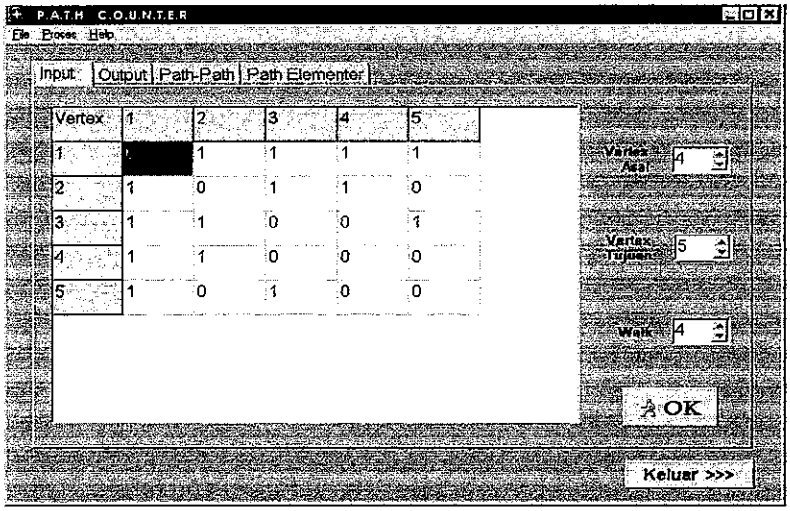
```
            BantuXAsli[SimpulTujuan,SimpulAsal]:=BantuXAsli
            [SimpulAsal,SimpulTujuan];
          end;
        end;
        Accept:=False;
      end;
end;

procedure TFormEditPosisi.FormShow(Sender: TObject);
var i,j:integer;
LebarX,LebarY:Integer;
begin
  BantuBackGround:='Default.bmp';
  ImageGraph.Picture.LoadFromFile(BantuBackGround);
  LebarX:=Round(ImageGraph.Picture.Width/ImageGraph.Width);
  LebarY:=Round(ImageGraph.Picture.Height/ImageGraph.Height)
  if LebarX>LebarY then
    ImageGraph.Canvas.Pen.Width:=LebarX
  else
    ImageGraph.Canvas.Pen.Width:=LebarY;
  for i:=1 to BantuNumSimpul do
    ImageGraph.Canvas.Ellipse(BantuSimpul[i].PosX-
    4*LebarX,BantuSimpul[i].PosY-4*LebarY,BantuSimpul[i].
    PosX+4*LebarX,BantuSimpul[i].PosY+4*LebarY);
      for i:=1 to BantuNumSimpul do
        for j:=i+1 to BantuNumSimpul do
        if (BantuXAsli[i,j]<>9999) then
        begin
          ImageGraph.Canvas.MoveTo(BantuSimpul[i].PosX,
          BantuSimpul[i].PosY);
          ImageGraph.Canvas.LineTo(BantuSimpul[j].PosX,
          BantuSimpul[j].PosY);
        end;
end;

procedure TFormEditPosisi.FormActivate(Sender: TObject);
begin
  Accept:=False;
end;

procedure TFormEditPosisi.ImageGraphMouseMove
        (Sender: TObject;Shift: TShiftState; X, Y: Integer);
var Simpulke,SimpulA,SimpulB:integer;
begin
  X:=Round(X*ImageGraph.Picture.Width/ImageGraph.Width);
  Y:=Round(Y*ImageGraph.Picture.Height/ImageGraph.Height);
    if TermasukSimpul(X,Y,Simpulke)then
    begin
      ImageGraph.Hint:=BantuSimpul[SimpulKe].NamaSimpul;
      ImageGraph.ShowHint:=True;
    end
```

```
    else if TermasukJalan(X,Y,SimpulA,SimpulB) then
    begin
      ImageGraph.Hint:=BantuSimpul[SimpulA].NamaSimpul+'-
      '+BantuSimpul[SimpulB].NamaSimpul+'='
      +IntToStr(BantuXAsli[SimpulA,SimpulB]);
      ImageGraph.ShowHint:=True;
    end
    else
      ImageGraph.ShowHint:=False;
end;

procedure TFormEditPosisi.FormResize(Sender: TObject);
begin
  if FormEditPosisi.Width<645 then
     FormEditPosisi.Width:=645;
  ImageGraph.Width:=FormEditPosisi.Width-30;
  ImageGraph.Width:=FormEditPosisi.Height-220;
  GroupBoxEdit.Top:=FormEditPosisi.Height-170;
  Panel1.Top:=FormEditPosisi.Height-170;
end;

procedureTFormEditPosisi.BitBtnTerimaClick(Sender: TObject);
var i,j:integer;
begin
  ordo:=BantuNumSimpul;
  for i:=1 to ordo do
    for j:= 1 to ordo do
      if BantuXAsli[i,j]=9999
      then Mtr[i,j]:=0
      else Mtr[i,j]:=1;
      begin
        FormEditPosisi.Close;
      end;
end;

end.
```

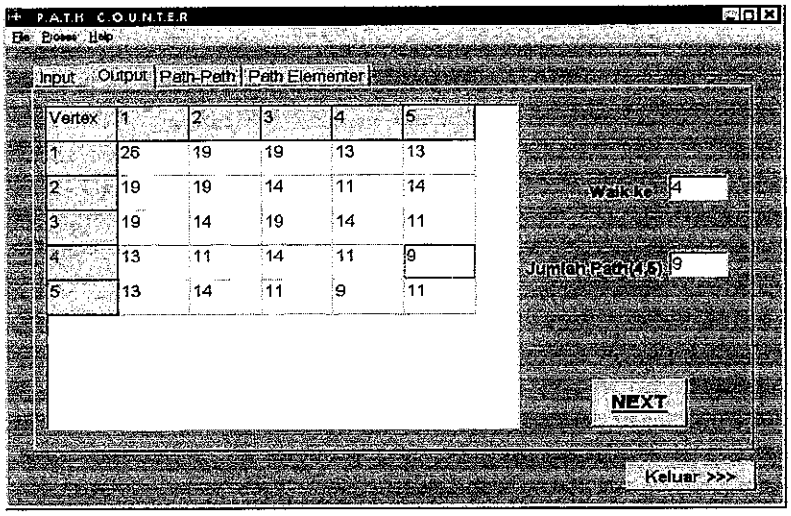
Lampiran 5: Tampilan Awal



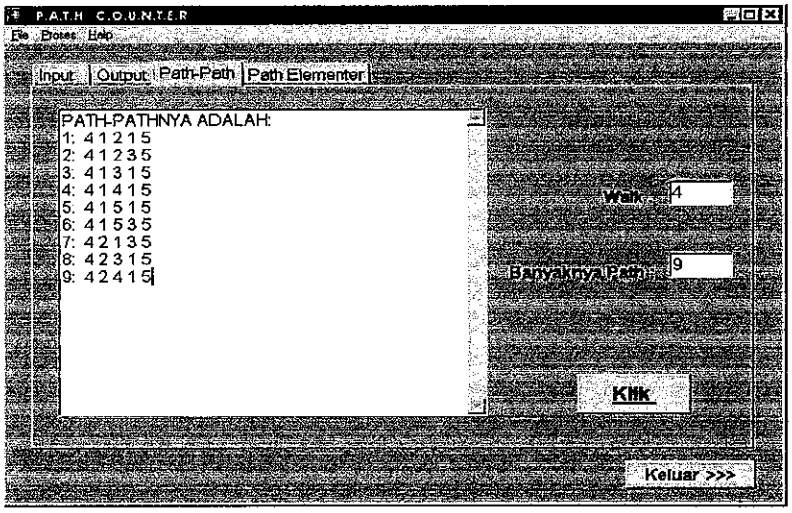
Lampiran 6: Tampilan Input Lewat Keyboard



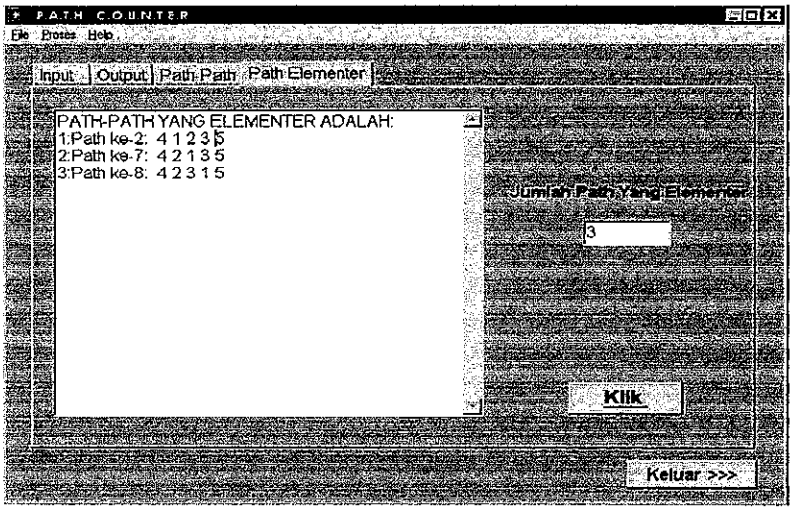
Lampiran 7 : Tampilan Output Data Dari Keyboard



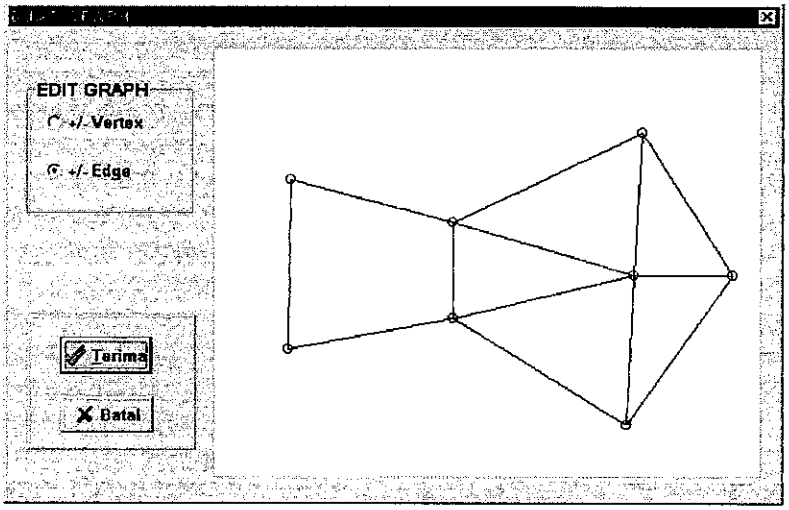
Lampiran 8 : Path-Path dari Vertex 4 ke Vertex 5



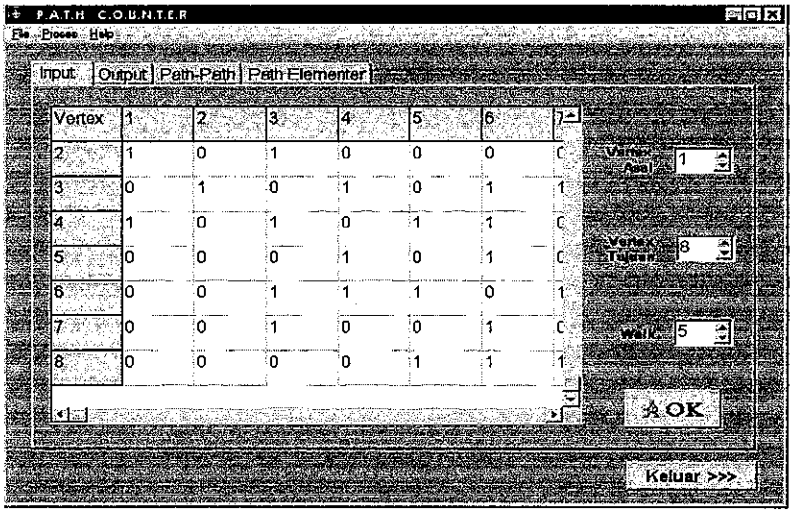
Lampiran 9 : Path Elementer dari Vertex 4 ke Vertex 5



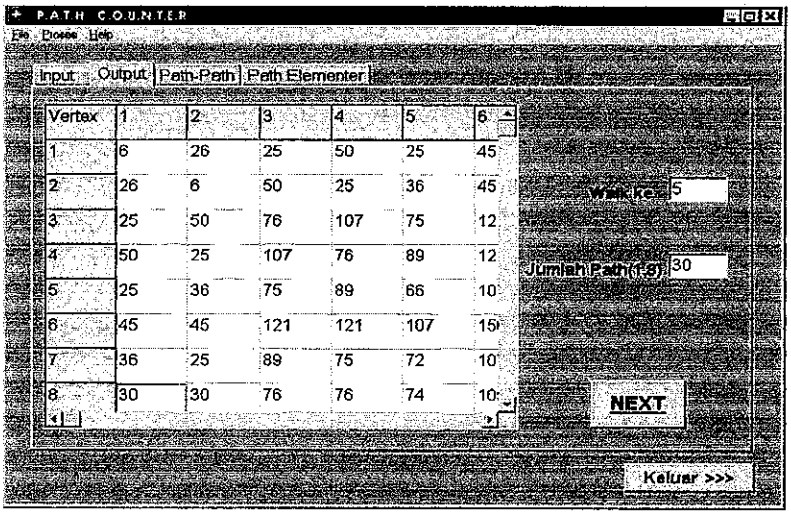
Lampiran 10 : Graph yang dibuat Secara Manual



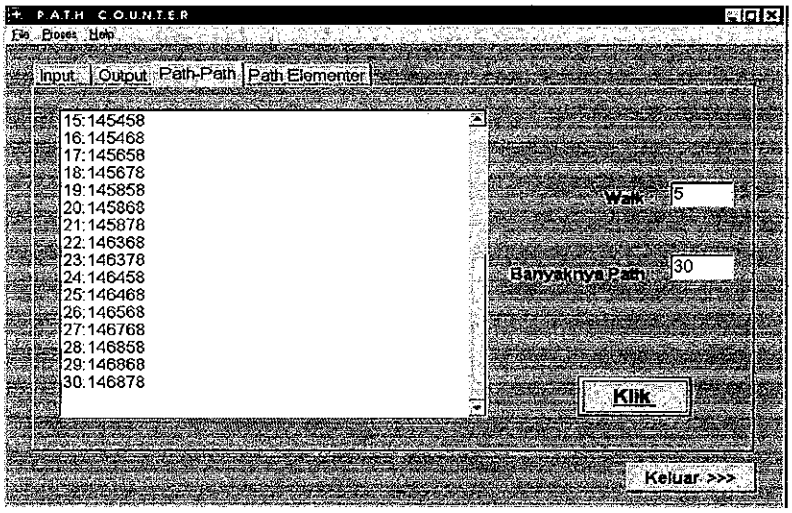
Lampiran 11 :Input Data dari Graph



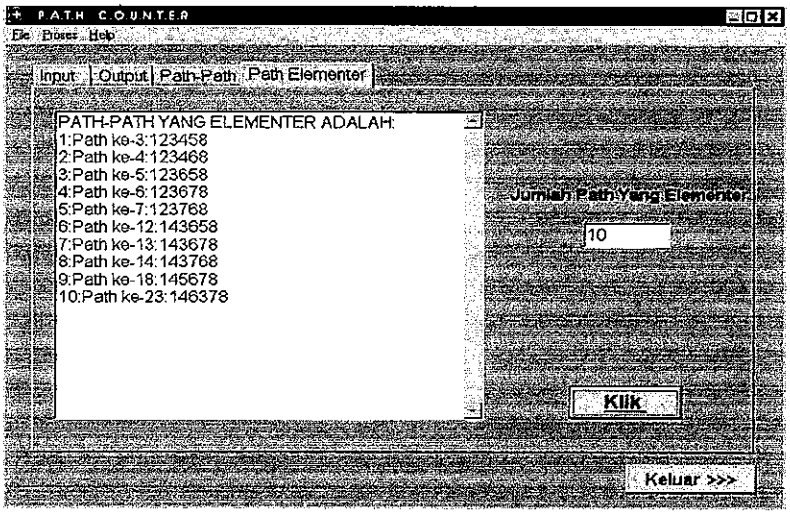
Lampiran 12 :Output Data dari Graph



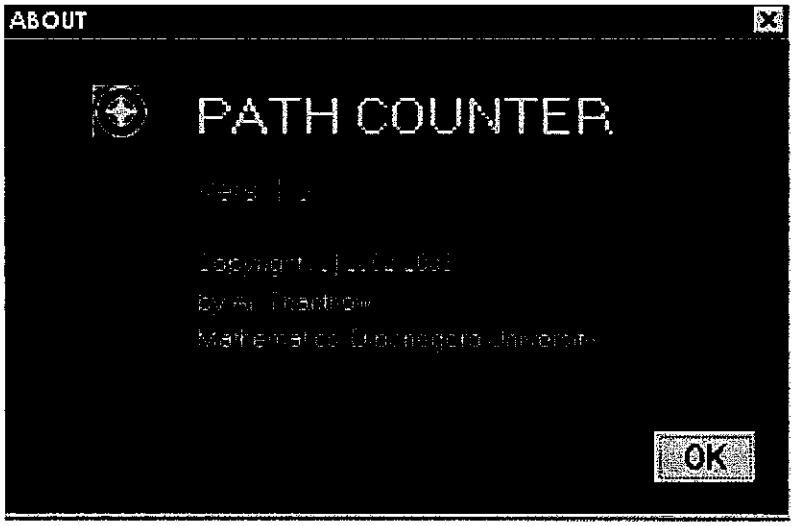
Lampiran 13 : Path-path dari Vertex 1 ke Vertex 8



Lampiran 14 : Path yang Elementer dari Vertex 1 ke Vertex 8



Lampiran 15 : Tampilan About



Lampiran 16 : Tampilan Help