

Lampiran 1 : Kode Program Function CekGraph

```
1.  function CekGraph(S:PSimpul):integer;
2.  var BCekS,
3.      SStop : PSimpul;
4.      n1,n2 : integer;
5.      Siklus : byte;
6.      SSiklus,
7.      T1,T2 : string3;

8.  function CariSimpulStop(S : PSimpul):PSimpul;
9.  begin
10.     if S <> nil then
11.     begin
12.         if S^.KeGaris = nil then
13.             CariSimpulStop := S
14.         else
15.         begin
16.             repeat
17.                 S :=S^.NextSimp;
18.             until (S = nil) or (S^.KeGaris = nil);
19.             CariSimpulStop := S;
20.         end
21.     end
22.     else
23.         CariSimpulStop := nil;
24.     end;

25.  procedure CekLintasan(Simpul,SimpPrev : PSimpul;var k:byte);
26.  var BGaris : PGaris;
27.  begin
28.     if Simpul^.KeGaris = nil then
29.     begin
30.         k := k+1;
31.         if Siklus <= 1 then
32.             SSiklus := '000';
33.         end
34.     else
35.     begin
36.         if Simpul^.StatusSimp = 0 then
37.         begin
38.             Simpul^.StatusSimp := 1;
39.             Simpul^.KjSimp := k;
40.             BGaris := Simpul^.KeGaris;
41.             BGaris^.StatusDef := true;
42.             SimpPrev:= Simpul;
43.             Simpul := BGaris^.KeSimp;
44.             CekLintasan(Simpul,SimpPrev,k);
45.             Simpul := SimpPrev;
46.             if Simpul^.ODSimp > 1 then
47.             begin
48.                 BGaris := Simpul^.KeGaris;
49.                 while BGaris <> nil do
50.                 begin
51.                     if BGaris^.statusdef <> true then
52.                     begin
53.                         SimpPrev:= Simpul;
54.                         Simpul := BGaris^.KeSimp;
55.                         CekLintasan(Simpul,SimpPrev,k);
56.                     end;
57.                     BGaris := BGaris^.NextGrs;
```

```

58.         end;
59.     end;
60. end
61. else
62. begin
63.     if simpul^.KjSimp = k then
64.         begin
65.             if SSiklus = '000' then
66.                 begin
67.                     SSiklus := Simpul^.NamaSimp;
68.                     Siklus := 1;
69.                 end
70.             else if SSiklus = Simpul^.NamaSimp then
71.                 Siklus := Siklus + 1;
72.                 if Siklus >= 2 then CekLintasan(SSStop, nil,k);
73.                 while (Simpul^.ODSimp = 1) do
74.                     begin
75.                         BGaris := CariGrsDefault(Simpul);
76.                         SimpPrev := Simpul;
77.                         Simpul := BGaris^.KeSimp;
78.                         if SSiklus = Simpul^.NamaSimp then
79.                             Siklus := Siklus + 1;
80.                             if Siklus >= 2 then CekLintasan(SSStop, nil,k);
81.                         end;
82.                         ChGrsDefault(Simpul);
83.                         T1 := Simpul^.NamaSimp;
84.                         T2 := Simpul^.KeGaris^.KeSimp^.NamaSimp;
85.                         HapusGaris(SS,T1,T2);
86.                         SisipSimpul(SS,SA,T1,T2);
87.                     end;
88.                 if Siklus >= 2 then CekLintasan(SSStop, nil,k)
89.                 else
90.                     begin
91.                         BGaris := CariGrsDefault(Simpul);
92.                         SimpPrev := Simpul;
93.                         Simpul := BGaris^.KeSimp;
94.                         CekLintasan(Simpul,SimpPrev,k);
95.                     end;
96.                 end;
97.     end;
98. end;

```

```
102. begin
103.   n1:=0;
104.   n2:=0;
105.   Siklus := 0;
106.   SSiklus := '000';
107.   if S = nil then CekGraph := 4
108.   else
109.     begin
110.       SStop := CariSimpulStop(S);
111.       BCekS := S;
112.       while BCekS <> nil do
113.         begin
114.           if BCekS^.ODSimp = 0 then
115.             n1 := n1+1;
116.           if BCekS^.IDSimp = 0 then
117.             n2 := n2+1;
118.           BCekS := BCekS^.NextSimp;
119.         end;
120.         if (n2>1) then
121.           CekGraph := 2
122.         else if (n1<>1) then
123.           CekGraph := 1
124.         else
125.           begin
126.             CekLintasan(SS,SP,k);
127.             k := 1;
128.             ResetNol(SS);
129.             if Siklus >= 2 then
130.               CekGraph := 3
131.             else
132.               CekGraph := 0;
133.           end;
134.         end;
135.       end;
```

Lampiran 2 : Kode Program Procedure CariLintasan

```
1. procedure CariLintasan(Simpul, SimpPrev : PSimpul; var k:byte);
2. var BGaris : PGaris;
3. begin
4.   if Simpul^.KeGaris = nil then
5.     begin
6.       n:=n+1;
7.       list[n]:= Simpul;
8.       write('Lintasan ',k:2,' : ');
9.       for m:= 1 to n do
10.        write(list[m]^>NamaSimp,' ');
11.        writeln;
12.        k := k+1;
13.     end
14.   else
15.     begin
16.       if Simpul^.StatusSimp = 0 then
17.         begin
18.           n:=n+1;
19.           list[n]:=simpul;
20.           Simpul^.StatusSimp := 1;
21.           Simpul^.KjSimp := k;
22.           BGaris := Simpul^.KeGaris;
23.           BGaris^.StatusDef := true;
24.           SimpPrev:= Simpul;
25.           Simpul := BGaris^.KeSimp;
26.           CariLintasan(Simpul, SimpPrev, k);
27.           Simpul := SimpPrev;
28.           if Simpul^.ODSimp > 1 then
29.             begin
30.               BGaris := Simpul^.KeGaris;
31.               while BGaris <> nil do
32.                 begin
33.                   if BGaris^.statusdef <> true then
34.                     begin
35.                       mn := 0;
36.                       for m := 1 to n do
37.                         if list[m] = Simpul then mn := m;
38.                         if mn <> 0 then n := mn
39.                         else list[n] := Simpul;
40.                         SimpPrev := Simpul;
41.                         Simpul := Bgaris^.KeSimp;
42.                         CariLintasan(BGaris^.KeSimp, Simpul, k);
43.                       end;
44.                       BGaris := BGaris^.NextGrs;
45.                     end;
46.                   end;
47.                 end
48.               else
49.                 begin
50.                   n:=n+1;
51.                   list[n]:=simpul;
52.                   BGaris := CariGrsDefault(Simpul);
53.                   SimpPrev := Simpul;
54.                   Simpul := BGaris^.KeSimp;
55.                   CariLintasan(Simpul, SimpPrev, k);
56.                 end;
57.             end;
58.           end;
```

Lampiran 3 : Kode Program Procedure Treesplay

```
1. {*****
2. * Procedure Operasi Splaying Pohon Splay *
3. *****}

4. procedure Treesplay(var pohon:tree; target:char);
5. var anakkanan, anakkiri, bantu, anak, lastsmall, firstlarge, null : tree;

6. procedure linkkiri;
7. begin
8.     lastsmall^.kanan := bantu;
9.     lastsmall      := bantu;
10.    bantu           := bantu^.kanan;
11. end;{* akhir procedure linkkiri *}

12. procedure linkkanan;
13. begin
14.     firstlarge^.kiri := bantu;
15.     firstlarge      := bantu;
16.     bantu           := bantu^.kiri;
17. end;{* akhir procedure linkkanan *}

18. procedure rotasikiri;
19. begin
20.     anakkanan := bantu^.kanan;
21.     bantu^.kanan:= anakkanan^.kiri;
22.     anakkanan^.kiri:= bantu;
23.     bantu := anakkanan;
24. end;{* akhir procedure rotasikiri *}

25. procedure rotasikanan;
26. begin
27.     anakkiri := bantu^.kiri;
28.     bantu^.kiri:= anakkiri^.kanan;
29.     anakkiri^.kanan:= bantu;
30.     bantu := anakkiri;
31. end;{* akhir procedure rotasikanan *}

32. {***** Bagian Operasi Splaying *****}

33. begin
34.     null^.info := target;
35.     bantu := pohon;
36.     lastsmall := null;
37.     firstlarge := null;
38.     if bantu^.info = target then
39.     begin
40.         textcolor(merah_t+kedip);writeln(' ':20,bel,ps2);
41.         textcolor(magenta_t);write(' ':6,'Simpul "',target,'" yang');
42.         writeln(' ':6,'sudah menjadi akar');
43.         writeln(' ':6,'sehingga pohon yang dihasilkan akan sama');
44.         writeln(' ':6,'dengan pohon splay sebelumnya');
45.         exit;
46.         readln;
47.     end;
```

```

48. while bantu^.info <> target do
49.     if bantu^.info < target then
50.         begin
51.             anak := bantu^.kanan;
52.             if target = anak^.info then linkkiri
53.             else if target > anak^.info then
54.                 begin
55.                     rotasikiri;
56.                     linkkiri
57.                 end
58.             else
59.                 begin
60.                     linkkiri;
61.                     linkkanan
62.                 end
63.             end
64.         else
65.             begin
66.                 anak := bantu^.kiri;
67.                 if target = anak^.info then linkkanan
68.                 else
69.                     if target < anak^.info then
70.                         begin
71.                             rotasikanan;
72.                             linkkanan
73.                         end
74.                     else
75.                         begin
76.                             linkkanan;
77.                             linkkiri
78.                         end
79.                     end;
80.                 if bantu = null then
81.                     begin
82.                         writeln(' c. Target telah disisipkan sebagai akar ');
83.                         pohon := baru(target);
84.                         pohon^.kiri := null;
85.                         pohon^.kanan := null;
86.                         bantu      := pohon;
87.                     end
88.                 else
89.                     writeln(' c. Target ditemukan dan menjadi akar');
90.                     lastsmall^.kanan := bantu^.kiri;
91.                     firstlarge^.kiri := bantu^.kanan;
92.                     bantu^.kanan    := null^.kiri;
93.                     bantu^.kiri     := null^.kanan;
94.                     pohon := bantu;
95.                     null^.info := char(0);
96.                     null^.kiri := null;
97.                     null^.kanan := null;
98.                 end; (* akhir procedure TreeSpaly *)

```

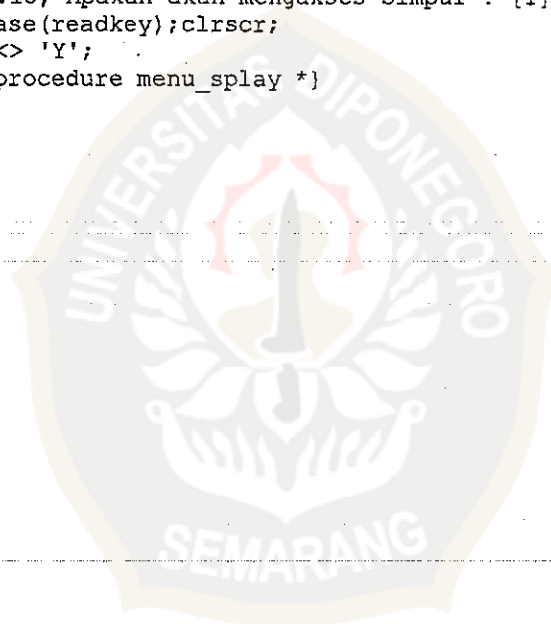
Lampiran 4 : Kode Program Procedure Menu_splay

```
1. {*****
2. *      Bagian Menu Pohon Splay      *
3. *****}

4. {***** Procedure Penyajian Menu Splay *****}

5. procedure menu_splay(var pohon : tree);
6. begin
7.   clrscr;gotoxy(20,1); textcolor(hijau_t);
8.   writeln('<<< PROSES PENYAJIAN POHON SPLAY >>>');
9.   gotoxy(20,2);writeln('Dengan preorder, inorder, dan postorder');
10.  gotoxy(20,2);writeln('=====');
11.  writeln;textcolor(cyan_t);
12.  writeln(' 1. Proses Pemasukkan Data');
13.  write('   Masukkan Untaian Simpul :');
14.  while not eoln do
15.  begin
16.    read(huruf);
17.    sisip_pohon(akar,huruf);
18.  end;
19.  readln;
20.  textcolor(cyan_t);
21.  writeln(' 2. Proses Pembacaan Simpul Pohon Biner Sebelum Splaying');
22.  textcolor(biru_t);
23.  write('   a. Preorder   : ');r_pre(akar);readln;
24.  write('   b. Inorder    : ');r_in(akar);readln;
25.  write('   c. Postorder   : ');r_pos(akar);readln;
26.  lagi :='Y';
27.  repeat
28.    textcolor(cyan_t);
29.    writeln(' 3. Proses pemasukkan data yang diakses ');
30.    write('   a. Masukkan Simpul yang Diakses : ');
31.    read(huruf);cari_simpul(ada, akar, huruf);
32.    if not (ada) then
33.    begin
34.      textcolor(merah_t+kedip);writeln(' ':20,bel,ps2);
35.      textcolor(magenta_t);
36.      write('   Simpul "',huruf,'" tidak ada dalam pohon');
37.      writeln('   << Pilihan Proses >>');
38.      writeln('   [1]. Menyisipkan Simpul');
39.      writeln('   [0]. Keluar dari Proses');
40.      while (pilihan<0) or (pilihan>3) do
41.      begin
42.        textcolor(merah_t);
43.        write(' ':7,'Tentukan pilihan anda (0-1) : ');
44.        readln(pilihan);writeln;
45.      end;
46.      case pilihan of
47.      1 : begin
48.          textcolor(putih_t);
49.          write('   b. Masukkan simpul yang disisipkan :');
50.          while not eoln do
51.          begin
52.            read(huruf); sisip_simpul(akar, huruf);
53.          end;
54.        end;
```

```
55.     2 : begin
56.         Writeln(' Tekan <enter> untuk Keluar Dari Proses ...! ');
57.         readln;exit;
58.     end
59. end;
60. end;
61. treesplay(akar, huruf);
62. textcolor(cyan_t);writeln;
63. Writeln('4. Proses Pembacaan Simpul Setelah Operasi Splaying');
64. textcolor(biru_t);
65. write('  a. Preorder : ');r_pre(akar);readln;writeln;
66. write('  b. Inorder : ');r_in(akar);readln;
67. write('  c. Posorder : ');r_pos(akar);readln;
68. writeln;
69. textcolor(putih_t);
70. write(' ':15,'Apakah akan mengakses Simpul ? [Y]a / [T]idak :');
71. lagi:=upcase(readkey);clrscr;
72. until lagi <> 'Y';
73. end;{* akhir procedure menu_splay *}
```



Lampiran 5 : Kode Program untuk Mendapatkan Satu Basis Set Lintasan Flowgraph

```

{$M 65300,0,655000}
{* Program untuk menentukan satu basis set lintasan dari flowgraph *
 *          disusun oleh : Sulardi (J 2A0 96 060)          *}
program FlowGraph;
uses crt;
type PSimpul = ^Node;
      PGaris  = ^Edge;
      String3 = string[3];
      Node    = record
        NamaSimp   : string3;
        KjSimp     : byte;
        StatusSimp : byte;
        KeGaris    : PGaris;
        NextSimp   : PSimpul;
        ODSimp     : byte;
        IDSimp     : byte;
      end;
      Edge = record
        NamaGrs   : string3;
        KeSimp    : PSimpul;
        NextGrs   : PGaris;
        StatusDef : Boolean;
      end;
      Data = record
        awal   : string3;
        tujuan : string3;
      end;
      string23 = string[23];
      RecordMenu = record
        PosisiMenu       : array[1..7] of byte;
        KarakterMenu     : array[1..7] of char;
        PosisiKarakterMenu : array[1..7] of byte;
        IsiMenu          : array[1..7] of string23;
      end;
      pixel = record
        karakter : char;
        atribut  : byte;
      end;
bufferlayar = array[1..25,1..80] of pixel;
const DataMenu : RecordMenu =
  (PosisiMenu : (9,10,11,12,13,14,15);
   KarakterMenu : 'IMBdScE';
   PosisiKarakterMenu : (28,28,28,29,28,28,29);
   IsiMenu : (' Informasi Menu           ', ' Masukan dari Keyboard ',
             ' Buka File                   ', ' Edit Flowgraph        ',
             ' Simpan ke File                   ', ' Cetak Lintasan        ',
             ' Keluar                               '));
DataMenuEdit : RecordMenu =
  (PosisiMenu : (9,10,11,12,13,14,0);
   KarakterMenu : 'IDSHGe ';
   PosisiKarakterMenu : (27,37,27,27,33,28,0);
   IsiMenu : ('Informasi Menu Edit      ', 'Tampilkan Daftar Simpul',
             'Sisip Simpul / Garis      ', 'Hapus Simpul           ',
             'Hapus Garis                 ', 'Ke Menu Utama         ', ''));
CR = ('Copy Right (c) 2001 Sulardi FMIPA UNDIP');
IsiMenuUtama : KetMenu =
  ('Memuat keterangan dari masing-masing pilihan menu ',
  'Memberi masukan flowgraph dari keyboard           ',
  'Memberi masukan flowgraph dari data di file       ',
  'Melakukan perubahan pada data flowgraph           ',
  'Menyimpan data flowgraph yang ada di memori ke file ',
  'Menampilkan lintasan yang dihasilkan dari flowgraph ');

```

```

IsiMenuEdit : KetMenu =
('Memuat keterangan dari masing-masing pilihan menu      ',
'Menampilkan daftar simpul flowgraph yang ada di memori',
'Menyisipkan pasangan simpul (garis) dari flowgraph     ',
'Menghapus sebuah simpul flowgraph                      ',
'Menghapus sebuah garis yang ada pada flowgraph         ',
'Kembali ke menu utama                                 ',
',',');

var k,n,mn,m      : byte;
    ss,sa,sp      : psimpul;
    list          : array[1..100] of PSimpul;
    g             : pgaris;
    Layar         : BufferLayar absolute $B800:$0000;
    LayarMenuUtama,LayarMenuEdit,LayarMasukan : BufferLayar;

procedure Petunjuk(P:string);
var x,y,n:integer;
begin
    textcolor(7);
    textbackground(8);
    gotoxy(1,24);write(P);
    x:=wherex;
    y:=wherey;
    n:=81-x;
    gotoxy(x,y);write(' ':n);
    textcolor(7);
    textbackground(1);
end;

procedure Peringatan(P:string);
var x,y,n:integer;
begin
    textcolor(7);
    textbackground(4);write(#7);
    gotoxy(1,24);write(P);
    x:=wherex;
    y:=wherey;
    n:=81-x;
    gotoxy(x,y);write(' ':n);
    textbackground(1);textcolor(7);
end;

procedure TungguTekan(Tunda : integer);
var i,j,x : integer;
    T,tekan : char;
begin
    i := 0;
    x := Tunda*10;
    repeat
        i := i+1;
        delay(10);
        j := (i mod 79)+1;
    until (keypressed) or (i > x);
    if keypressed then
        begin
            tekan := readkey;
            write(#7);
        end;
    end;
end;

```

```

procedure AturanPenggunaan;
var i : byte;
begin
  textbackground(1);textcolor(2);clrscr;
  for i := 1 to 79 do
    begin
      gotoxy(i,2);write('■'); gotoxy(i,25);write('■');
    end;
    textcolor(14);
    gotoxy(25,2);write('PETUNJUK PENGGUNAAN PROGRAM');
    gotoxy(1,4);write(' Petunjuk Pemberian Masukan');
    gotoxy(1,17);write(' Petunjuk Proses Cetak Lintasan');
    gotoxy(25,23);write('Tekan sembarang tombol ... !');
    textcolor(7);
    gotoxy(1,12);write(' Masukan dari Keyboard');
    gotoxy(48,13);write(' Edit Flowgraph');
    gotoxy(61,11);write(' Buka File');
    gotoxy(55,15);write(' Simpan ke File');textcolor(2);
    gotoxy(1,5);write(' 1. Masukan yang diberikan berupa karakter ASCII untuk menamai
      simpul flograph');
    gotoxy(1,6);
    write(' 2. Syarat-syarat masukan agar dapat diproses dan memberikan keluaran : ');
    gotoxy(1,7);
    write(' a) Terdapat sebuah simpul start dan sebuah simpul stop');
    gotoxy(1,8);
    write(' b) dari simpul start dapat dibuat lintasan yang melewati setiap simpul');
    gotoxy(1,9);
    write(' c) dari masing-masing simpul dapat dibuat lintasan yang melewati simpul stop');
    gotoxy(1,10);
    write(' 3. Masukan data dapat diberikan dengan membaca file yang menyimpan data ');
    gotoxy(1,11);write(' flowgraph. Pemanggilan file dilakukan dari pilihan menu');
    gotoxy(1,12);
    write(' 4. Untuk memulai pemasukan data flowgraph harus dimulai dari pilihan menu ');
    gotoxy(26,13);write(' tidak boleh dari menu');
    gotoxy(1,14);
    write(' 5. Agar data flowgraph aman, sebelum melakukan proses lebih lanjut, simpan ');
    gotoxy(1,15);write(' data flowgraph dalam bentuk file dari pilihan menu');
    gotoxy(1,18);
    write(' Program akan memberikan keluaran satu basis set lintasan flowgraph apabila');
    gotoxy(1,19);
    write(' masukan flowgraph telah memenuhi syarat-syarat masukan di atas. Apabila ');
    gotoxy(1,20);
    write(' masukan flowgraph belum lengkap, maka program akan meminta pemakai untuk');
    gotoxy(1,21);write(' melengkapinya dari menu edit');
    gotoxy(53,25);TungguTekan(500);
  end;

function Initsimpul(T:string3):PSimpul;
var Init : PSimpul;
begin
  new(Init);
  with Init do
    begin
      NamaSimp := T;
      KjSimp := 0;
      StatusSimp := 0;
      KeGaris := nil;
      NextSimp := nil;
      ODSimp := 0;
      IDSimp := 0;
    end;
  InitSimpul := Init;
end;

```

```

function InitGaris(G:string3):PGaris;
var Init : PGaris;
begin
  new(Init);
  with Init^ do
  begin
    NamaGrs := G;
    StatusDef := false;
    KeSimp := nil;
    NextGrs := nil;
  end;
  InitGaris := Init;
end;

function CariSimpul(S:PSimpul;C:string3):PSimpul;
var BCari : PSimpul;
begin
  if S <> nil then
  begin
    BCari:=S;
    if BCari^.NamaSimp = C then CariSimpul := BCari
    else
    begin
      repeat
        BCari:=BCari^.NextSimp;
      until (BCari = nil) or (BCari^.NamaSimp = C);
      CariSimpul:=BCari;
    end
  end
  else CariSimpul := nil;
end;

procedure SisipSimpul(var SS,SA : PSimpul; T1,T2: string3);
var SSimp : PSimpul;
  BGaris,SGrS : PGaris;
  sudah:boolean;
begin
  if SS = nil then
  begin
    SS := InitSimpul(T1);
    SA := SS;
    SSimp:= SS;
  end
  else
  begin
    SSimp := CariSimpul(SS,T1);
    if SSimp = nil then
    begin
      SSimp := InitSimpul(T1);
      SA^.NextSimp := SSimp;
      SA := SSimp;
    end;
  end;
  if SSimp^.KeGaris = nil then
  begin
    SGrS := InitGaris(T2);
    SSimp^.KeGaris := SGrS;
    BGaris := SGrS;
    SSimp:=CariSimpul(SS,T2);
    if SSimp = nil then
    begin
      SSimp := InitSimpul(T2);
      SA^.NextSimp:=SSimp;

```

```

    SA := SSimp;
  end;
  BGaris^.KeSimp := SSimp;
end
else
begin
  sudah := false;
  BGaris := SSimp^.KeGaris;
  while BGaris <> nil do
  begin
    if BGaris^.NamaGrs = T2 then
      sudah := true;
      BGaris := BGaris^.NextGrs;
    end;
    if sudah = true then
      Peringatan('Garis tersebut telah ada !')
    else
      begin
        SGrS := InitGaris(T2);
        BGaris:=SSimp^.KeGaris;
        while BGaris^.NextGrs <> nil do
          BGaris := BGaris^.NextGrs;
          BGaris^.NextGrs := SGrS;
          BGaris := SGrS;
        SSimp:=CariSimpul(SS,T2);
        if SSimp = nil then
          begin
            SSimp := InitSimpul(T2);
            SA^.NextSimp:=SSimp;
            SA := SSimp;
          end;
          BGaris^.KeSimp := SSimp;
        end;
      end;
    end;
  end;
end;

function CariGaris(S:PSimpul;NG:string3):PGaris;
var G : PGaris;
begin
  G:=S^.KeGaris;
  while (G<>nil) and (G^.NamaGrs <> NG) do
    G:=G^.NextGrs;
  CariGaris := G;
end;

procedure HapusGaris(SS:PSimpul;T1,T2:string3);
var S : PSimpul;
    G,GPrev:PGaris;
begin
  S:=CariSimpul(SS,T1);
  if S = nil then
    Peringatan('Garis tersebut tidak ada !')
  else
    begin
      G:=CariGaris(S,T2);
      if G = nil then
        Peringatan('Garis tersebut tidak ada !')
      else
        begin
          G := S^.KeGaris;
          GPrev := G;
          while G^.NamaGrs <> T2 do
            begin

```

```

    GPrev := G;
    G := G^.NextGrS;
end;
if GPrev = G then
begin
    GPrev := GPrev^.NextGrS;
    S^.KeGaris := GPrev;
    dispose(G);
end
else
begin
    GPrev^.NextGrS := G^.NextGrS;
    dispose(G);
end;
Petunjuk('Garis telah dihapus ');
end;
end;
end;

procedure HapusSimpul(SS:PSimpul;var SA:PSimpul;T1:string3);
var S,SPrev : PSimpul;
    G,GH : PGaris;
    T : string3;
begin
    S := CariSimpul(SS,T1);
    if S = nil then
begin
    Peringatan('Simpul tersebut tidak ada.... tekan <enter> ');
    readln;
end
else if S = SS then
    Peringatan('Simpul ini tidak boleh dihapus karena merupakan simpul start ! ')
else
begin
    S := SS;
    while S <> nil do
begin
    G := CariGaris(S,T1);
    if G <> nil then
begin
    T := S^.NamaSimp;
    HapusGaris(SS,T,T1);
end;
    S := S^.NextSimp;
end;
    S := CariSimpul(SS,T1);
    G := S^.KeGaris;
    S^.KeGaris := nil;
    while G <> nil do
begin
    GH := G;
    G := G^.NextGrS;
    GH^.NextGrS := nil;
    dispose(GH);
end;
    S:=SS;
    while (S^.NamaSimp <> T1) do
begin
    SPrev := S;
    S := S^.NextSimp;
end;
end;
end;

```

```

    if S = SA then
    begin
        SA := SPrev;
        SA^.NextSimp := nil;
    end;
    SPrev^.NextSimp := S^.NextSimp;
    S^.NextSimp := nil;
    dispose(S);
    Petunjuk(' Simpul tersebut telah dihapus....');
end;
end;

```

```

procedure HapusFlowGraph(var SS,SA:PSimpul);
var T,G:string3;
    S:PSimpul;
begin
    S:=SS^.NextSimp;
    if S <> nil then
        repeat
            T := S^.NamaSimp;
            HapusSimpul(SS,SA,T);
            S := S^.NextSimp;
        until S = nil;
    SA := nil;
    SS := nil;
end;

```

```

procedure ChStGaris(var G : PGaris);
begin
    if G^.StatusDef <> false then
        G^.StatusDef := false
    else
        G^.StatusDef := true;
end;

```

```

function CariGrSDefault(Simp : PSimpul): PGaris;
var G : PGaris;
begin
    G := Simp^.KeGaris;
    while (G^.StatusDef <> true) do
        G := G^.NextGrS;
    CariGrSDefault := G;
end;

```

```

Procedure ChGrSDefault(S : PSimpul);
var G : PGaris;
begin
    G := CariGrSDefault(S);
    ChStGaris(G);
    if G^.NextGrS <> nil then
        ChStGaris(G^.NextGrS)
    else
        begin
            G := S^.KeGaris;
            ChStGaris(G);
        end;
end;
end;

```

```

procedure ResetNol(Simpul:PSimpul);
var S : PSimpul;
    G : PGaris;
begin
    S := Simpul;

```

```

while S <> nil do
begin
  G := S^.KeGaris;
  while G <> nil do
  begin
    G^.StatusDef := false;
    G := G^.NextGrS;
  end;
  S^.StatusSimp := 0;
  S^.KjSimp := 0;
  S := S^.NextSimp;
end;
end;

procedure HitungODSimpul(var Simpul : PSimpul);
var n : byte;
  BHitung : PSimpul;
  BGaris : PGaris;
begin
  n := 0;
  BHitung := Simpul;
  while BHitung <> nil do
  begin
    BGaris := BHitung^.KeGaris;
    while BGaris <> nil do
    begin
      n:=n+1;
      BGaris := BGaris^.NextGrS;
    end;
    BHitung^.ODSimp := n;
    n := 0;
    BHitung := BHitung^.NextSimp;
  end;
end;

procedure HitungIDSimpul(var Simpul:PSimpul);
var n:byte;
  BHitungS1,BHitungS2 : PSimpul;
  BHitungG : PGaris;
begin
  BHitungS1 := Simpul;
  while BHitungS1 <> nil do
  begin
    BHitungS2 := Simpul;
    n:=0;
    while BHitungS2 <> nil do
    begin
      BHitungG := BHitungS2^.KeGaris;
      while BHitungG <> nil do
      begin
        if BHitungG^.KeSimp = BHitungS1 then
          n:=n+1;
          BHitungG := BHitungG^.NextGrS;
        end;
        BHitungS2 := BHitungS2^.NextSimp;
      end;
      BHitungS1^.IDSimp := n;
      BHitungS1 := BHitungS1^.NextSimp;
    end;
  end;
end;

```



```

procedure FormBSFile(var NamaFile:string);
begin
  textbackground(0);textcolor(2);
  gotoxy(18,13);write(' ');
  gotoxy(18,14);write(' Masukan nama file yang akan di gunakan ');
  gotoxy(18,15);write(' dengan nama drive tempat penyimpanan ');
  gotoxy(18,16);write(' Nama file : ');
  gotoxy(18,17);write(' ');
  textbackground(7);
  gotoxy(32,16);write(' ');
  gotoxy(33,16);readln(NamaFile);
  textbackground(1);textcolor(7);
end;

procedure SimpanKeFile(var SS:PSimpul);
var DataSimp : data;
    FileGraph : file of Data;
    NamaFile : string;
    S : PSimpul;
begin
  S:=SS;
  FormBSFile(NamaFile);
  textbackground(0);textcolor(2);
  gotoxy(20,14);write(' ');
  gotoxy(20,15);write(' Tunggu sedang menyimpan ');
  gotoxy(20,16);write(' ');
  textbackground(1);textcolor(7);
  Assign(FileGraph,NamaFile);
  {$I-}
  rewrite(FileGraph);
  if IOresult <> 0 then
  begin
    textbackground(0);
    gotoxy(20,15);write(' Drive tidak siap ..... ');
    peringatan('Tekan sembarang tombol untuk ke menu utama ');
    readln;exit;textbackground(1);
  end;
  {$I+}
  while S <> nil do
  begin
    G := S^.KeGaris;
    DataSimp.awal := S^.NamaSimp;
    if G <> nil then
    begin
      while G <> nil do
      begin
        DataSimp.tujuan := G^.KeSimp^.NamaSimp;
        write(FileGraph,DataSimp);
        G := G^.NextGrs;
      end;
    end;
    S := S^.NextSimp;
  end;
  close(FileGraph);
end;

```

```

procedure BukaFile(var SS,SA:PSimpul);
var T1,T2      : string[3];
    IObuka,i,JumlahRecord: integer;
    jawab      : char;
    DataSimp   : data;
    NamaFile   : string;
    FileGraph  : file of Data;
begin
  textbackground(1);
  if SS <> nil then
  begin
    jawab:=' ';
    petunjuk(' Jika simpan, disimpan dari pilihan menu utama ');
    while not ((jawab='S') or (jawab='H')) do
    begin
      textbackground(3);textcolor(4);
      gotoxy(10,12);write(' ');
      gotoxy(10,13);write(' Terdapat flowgraph di memori simpan / hapus (S/H) ? ');
      gotoxy(10,14);write(' ');
      gotoxy(62,13);read(jawab);
      jawab:=upcase(jawab);textbackground(1);textcolor(7);
    end;
    if jawab = 'S' then exit;
    if jawab = 'H' then HapusFlowGraph(SS,SA);
  end;
  FormBSFile(NamaFile);
  Assign(FileGraph,NamaFile);
  {$I-}
  reset(FileGraph);
  {$I+}
  IObuka := IOresult;
  if IObuka <> 0 then
  begin
    Peringatan(' File tidak ditemukan ( tekan <enter> untuk ke menu utama ! )');
    textbackground(3);textcolor(4);
    gotoxy(19,13);write(' ');
    gotoxy(19,14);write(' File tidak ditemukan . . . ');
    gotoxy(19,15);write(' ');
    gotoxy(49,14);readln;textbackground(1);textcolor(7);
  end
  else
  begin
    JumlahRecord := FileSize(FileGraph);
    for i:= 1 to JumlahRecord do
    begin
      read(FileGraph,DataSimp);
      T1 := DataSimp.awal;
      T2 := DataSimp.tujuan;
      SisipSimpul(SS,SA,T1,T2);
    end;
    close(FileGraph);
  end;
end;

function CekGraph(S:PSimpul):integer;
var BCekS,
    SStop : PSimpul;
    n1,n2 : integer;
    Siklus : byte;
    SSiklus,
    T1,T2 : string3;

```

```

function CariSimpulStop(S : PSimpul):PSimpul;
begin
  if S <> nil then
    begin
      if S^.KeGaris = nil then
        CariSimpulStop := S
      else
        begin
          repeat
            S :=S^.NextSimp;
          until (S = nil) or (S^.KeGaris = nil);
          CariSimpulStop := S;
        end
      end
    else
      CariSimpulStop := nil;
    end;
end;

procedure CekLintasan(Simpul,SimpPrev : PSimpul;var k:byte);
var BGaris : PGaris;
begin
  if Simpul^.KeGaris = nil then
    begin
      k := k+1;
      if Siklus <= 1 then
        SSiklus := '000';
      end
    else
      begin
        if Simpul^.StatusSimp = 0 then
          begin
            Simpul^.StatusSimp := 1;
            Simpul^.KjSimp := k;
            BGaris := Simpul^.KeGaris;
            BGaris^.StatusDef := true;
            SimpPrev:= Simpul;
            Simpul := BGaris^.KeSimp;
            CekLintasan(Simpul,SimpPrev,k);
            Simpul := SimpPrev;
            if Simpul^.ODSimp > 1 then
              begin
                BGaris := Simpul^.KeGaris;
                while BGaris <> nil do
                  begin
                    if BGaris^.statusdef <> true then
                      begin
                        SimpPrev:= Simpul;
                        Simpul := BGaris^.KeSimp;
                        CekLintasan(Simpul,SimpPrev,k);
                      end;
                    BGaris := BGaris^.NextGrS;
                  end;
                end;
              end
            else
              begin
                if simpul^.KjSimp = k then
                  begin
                    if SSiklus = '000' then
                      begin
                        SSiklus := Simpul^.NamaSimp;
                        Siklus := 1;
                      end
                    end
                  end
                end
              end
            end
          end
        end
      end
    end;
end

```

```

else if SSiklus = Simpul^.NamaSimp then
  Siklus := Siklus + 1;
  if Siklus >= 2 then CekLintasan(SStop,nil,k);
  while (Simpul^.ODSimp = 1) and (Siklus < 2) do
    begin
      BGaris := CariGrsDefault(Simpul);
      SimpPrev := Simpul;
      Simpul := BGaris^.KeSimp;
      if SSiklus = Simpul^.NamaSimp then
        Siklus := Siklus + 1;
        if Siklus >= 2 then CekLintasan(SStop,nil,k);
      end;
      if Simpul^.ODSimp <> 1 then
        begin
          ChGrsDefault(Simpul);
          T1 := Simpul^.NamaSimp;
          T2 := Simpul^.KeGaris^.KeSimp^.NamaSimp;
          HapusGaris(SS,T1,T2);
          SisipSimpul(SS,SA,T1,T2);
        end;
      end;
      if Siklus >= 2 then CekLintasan(SStop,nil,k)
    else
      begin
        BGaris := CariGrsDefault(Simpul);
        SimpPrev := Simpul;
        Simpul := BGaris^.KeSimp;
        CekLintasan(Simpul,SimpPrev,k);
      end;
    end;
  end;
end;
end;
begin
  n1:=0;
  n2:=0;
  Siklus := 0;
  SSiklus := '000';
  if S = nil then CekGraph := 4
  else
    begin
      SStop := CariSimpulStop(S);
      BCeks := S;
      while BCeks <> nil do
        begin
          if BCeks^.ODSimp = 0 then n1 := n1+1;
          if BCeks^.IDSimp = 0 then n2 := n2+1;
          BCeks := BCeks^.NextSimp;
        end;
        if (n2>1) then CekGraph := 2
        else if (n1<>1) then CekGraph := 1
        else
          begin
            CekLintasan(SS,SP,k);
            k := 1;
            ResetNol(SS);
            if Siklus >= 2 then CekGraph := 3
            else CekGraph := 0;
          end;
        end;
      end;
    end;
end;
end;

```

```

procedure CariLintasan(Simpul,SimpPrev : PSimpul;var k:byte);
var BGaris : PGaris;
begin
  if Simpul^.KeGaris = nil then
  begin
    n:=n+1;
    list[n]:= Simpul;
    write('Lintasan ',k:2,' : ');
    for m:= 1 to n do
      write(list[m]^>NamaSimp,' ');
    writeln;
    k := k+1;
  end
  else
  begin
    if Simpul^.StatusSimp = 0 then
    begin
      n:=n+1;
      list[n]:=simpul;
      Simpul^.StatusSimp := 1;
      BGaris := Simpul^.KeGaris;
      BGaris^.StatusDef := true;
      SimpPrev:= Simpul;
      Simpul := BGaris^.KeSimp;
      CariLintasan(Simpul,SimpPrev,k);
      Simpul := SimpPrev;
      if Simpul^.ODSimp > 1 then
      begin
        BGaris := Simpul^.KeGaris;
        while BGaris <> nil do
        begin
          if BGaris^.statusdef <> true then
          begin
            mn := 0;
            for m := 1 to n do
              if list[m] = Simpul then mn := m;
            if mn <> 0 then n := mn
            else list[n] := Simpul;
            CariLintasan(BGaris^.KeSimp,Simpul,k);
          end;
          BGaris := BGaris^.NextGrS;
        end;
      end;
    end
    else
    begin
      n:=n+1;
      list[n]:=simpul;
      BGaris := CariGrSDefault(Simpul);
      SimpPrev := Simpul;
      Simpul := BGaris^.KeSimp;
      CariLintasan(Simpul,SimpPrev,k);
    end;
  end;
end;
end;

```

```

procedure ChAtributLayar(x1,y1,x2,y2,Atribut : byte);
var posisi : word;
    i,j      : byte;
begin
  for i := y1 to y2 do
  begin
    for j := x1 to x2 do
    begin
      posisi := ((i-1)*80+j)-1)*2;
      Mem[$B800:$0000+posisi+1] := Atribut;
    end;
  end;
end;

```

```

procedure KembalikanLayar(x1,y1,x2,y2:byte;var LayarSimpan : BufferLayar);
var posisi : word;
    i,j      : byte;
begin
  for i := y1 to y2 do
  begin
    for j := x1 to x2 do
    with LayarSimpan[i,j] do
    begin
      posisi := ((i-1)*80+j)-1)*2;
      Mem[$B800:$0000+posisi] := ord(karakter);
      Mem[$B800:$0000+posisi+1] := Atribut;
    end;
  end;
end;

```

```

procedure ChrJalan(var n:byte;LayarSimpan : BufferLayar);
var i,j,k,l,m : byte;
    posisi      : word;
begin
  if n < 80 then
  begin
    m:=n;l:=0;k:=80-n;
  end
  else if n > 159 then
  begin
    m:=1;n:=1;l:=1+1;k:=1;
  end
  else
  begin
    m:=80;l:=1+1;k:=1;
  end;
  for i:=1 to m do
  begin
    posisi := ((23*80+k)-1)*2;
    if (i+1) <= 80 then
      mem[$B800:$0000+posisi] := ord(LayarSimpan[24,i+1].karakter)
    else
      mem[$B800:$0000+posisi] := ord(0);
    k := k+1;
  end;
  n := n+1;
end;

```

```

procedure InformasiMenuUtama;
var i : integer;
begin
  textbackground(1);clrscr;
  textcolor(14);gotoxy(1,3);
  for i := 1 to 80 do
    begin
      gotoxy(i,3);write(' ');
      gotoxy(i,24);write(' ');
    end;
  gotoxy(33,3);write('INFORMASI MENU');
  for i:= 2 to 7 do
    begin
      gotoxy(1,i*3-1);write(DataMenu.IsiMenu[i]);
    end;
  textcolor(2);
  gotoxy(3,6);write('Memberi masukan flowgraph dari papan ketik');
  gotoxy(3,7);write('apabila terdapat data flowgraph pada memori akan dihapus');
  gotoxy(3,9);
  write('Memberi masukan flowgraph dengan membuka file yang menyimpan data flowgraph');
  gotoxy(3,10);
  write('nama file dan path penyimpanan diketikkan pada tempat yang disediakan');
  gotoxy(3,12);write('Melakukan editing terhadap masukan flowgraph');
  gotoxy(3,13);write('berupa penambahan, dan penghapusan simpul atau garis');
  gotoxy(3,15);write('Menyimpan flowgraph yang ada di memori ke media penyimpanan');
  gotoxy(3,16);write('disket atau hard disk');
  gotoxy(3,18);write('Menampilkan keluaran proses berupa satu set lintasan');
  gotoxy(3,19);write('dan nilai kompleksitas siklomatik dari flowgraph tersebut');
  gotoxy(3,21);write('Kembali ke DOS');
  gotoxy(3,22);write('data flowgraph di memori otomatis hilang');
  gotoxy(1,25);textcolor(14);
  write('Tekan <enter>');readln;
end;

procedure InformasiMenuEdit;
var i : integer;
begin
  textbackground(1);clrscr;
  textcolor(14);gotoxy(1,3);
  for i := 1 to 80 do
    begin
      gotoxy(i,3);write(' ');
      gotoxy(i,20);write(' ');
    end;
  gotoxy(30,3);write('INFORMASI MENU EDIT');
  for i:= 2 to 6 do
    begin
      gotoxy(1,i*3-1);write(DataMenuEdit.IsiMenu[i]);
    end;
  textcolor(2);
  gotoxy(3,6);write('Menampilkan daftar simpul yang tersimpan di memori');
  gotoxy(3,9);write('Menambah simpul atau garis pada flowgraph yang ada di memori');
  gotoxy(3,15);write('Menghapus garis flowgraph yang ada di memori');
  gotoxy(3,12);write('Menghapus simpul flowgraph yang ada di memori');
  gotoxy(3,18);write('Kembali ke menu utama');
  gotoxy(1,21);textcolor(14);
  write('Tekan <enter>');readln;
end;

```

```

procedure DaftarSimpul(SS : PSimpul);
var br,kl : byte;
    S      : PSimpul;
    G      : PGaris;
begin
    S:=SS;
    br:=8;
    kl:=4;
    while S <> nil do
    begin
        if (kl >= 64) and (br >= 22) then
        begin
            br := 8;kl := 4;
        end
        else if br >= 22 then
        begin
            br := 8;kl := kl + 8;
        end;
        gotoxy(kl,br);write(S^.NamaSimp);
        G :=S^.KeGaris;
        if G <> nil then
            while G <> nil do
            begin
                gotoxy(kl+3,br);write(chr(26),G^.KeSimp^.NamaSimp);
                if (kl >= 64) and (br >= 20) then
                begin
                    br := 8;kl := 4;
                end
                else if br >= 20 then
                begin
                    br := 8;kl := kl + 8;
                end
                else br := br + 1;
                G := G^.NextGrs;
            end
            else-br := br+1;
            S := S^.NextSimp;
        end;
        gotoxy(77,22);readln;
    end;

procedure InitFlowGraph(var GA,GAK : PSimpul;T : string3);
begin
    GA := InitSimpul(T);
    GAK := GA;
end;

procedure MasukanInisialisasi(var T:string3);
begin
    petunjuk('Masukkan karakter untuk menamai simpul start flowgraph kemudian tekan
    <enter> ');
    textbackground(2);
    textcolor(14);
    gotoxy(10,11);write(' ');
    gotoxy(10,12);write(' Masukkan nama simpul start flowgraph untuk inisialisasi ');
    gotoxy(10,13);write('          simpul : ');
    gotoxy(10,14);write(' ');
    gotoxy(36,13);readln(T);
    textcolor(7);textbackground(1);
end;

```



```

procedure MasukanGaris(GA : PSimpul;var GAK:PSimpul;T:string3;var G:string3);
begin
    petunjuk('Masukan karakter-karakter untuk menamai simpul asal dan simpul tujuan
              garis');
    gotoxy(14,3);write(T);
    petunjuk('Masukan karakter-karakter untuk menamai simpul asal dan simpul tujuan
              garis');
    gotoxy(14,4);readln(G);
    SisipSimpul(GA,GAK,T,G);
end;

```

```

procedure MasukanSimpul(var T:string3);
begin
    petunjuk('Masukan karakter-karakter untuk menamai simpul asal dan simpul tujuan
              garis');
    gotoxy(14,3);readln(T);
end;

```

```

procedure EditMasukan(SS : PSimpul;var SA : PSimpul);
var lagi : char;
    T,G : string3;
begin
    repeat
        KembaliLayar(14,3,20,3,LayarMasukan);
        KembaliLayar(14,4,20,4,LayarMasukan);
        MasukanSimpul(T);
        repeat
            KembaliLayar(14,4,20,4,LayarMasukan);
            MasukanGaris(SS,SA,T,G);
            lagi := ' ';
            while not ((lagi = 'Y') or (lagi = 'T')) do
                begin
                    gotoxy(40,3);write(' Ada cabang (Y/T) ? ');
                    lagi := upcase(readkey);
                end;
            KembaliLayar(40,3,70,3,LayarMasukan);
            until lagi = 'T';
            lagi := ' ';
            while not ((lagi = 'Y') or (lagi = 'T')) do
                begin
                    gotoxy(40,3);write(' Ada simpul yang lain (Y/T) ? ');
                    lagi := upcase(readkey);
                end;
            KembaliLayar(40,3,80,3,LayarMasukan);
            until lagi = 'T';
            DaftarSimpul(SS);
    end;

```

```

procedure Masukan(var SS,SA:PSimpul);
var x1,y1,x2,y2,br,kl : byte;
    T,G : string3;
    lagi : char;
    cek : boolean;
begin
    AturanPenggunaan;
    if SS <> nil then
        HapusFlowGraph(SS,SA);
        KembaliLayar(1,1,80,25,LayarMasukan);
        MasukanInisialisasi(T);
        InitFlowGraph(SS,SA,T);
        KembaliLayar(10,11,66,14,LayarMasukan);
        EditMasukan(SS,SA);
    end;

```

```

Function HitungKompleksitas(S:PSimpul):integer;
var n,e : integer;
    G : PGaris;
begin
    e := 0;
    n := 0;
    while S <> nil do
    begin
        G := S^.KeGaris;
        while G <> nil do
        begin
            e := e + 1 ;
            G := G^.NextGrS;
        end;
        n := n + 1;
        S := S^.NextSimp;
    end;
    HitungKompleksitas := e - n + 2;
end;

procedure CetakLintasan(SS : PSimpul);
var jawab : char;
    T,G : string3;
    S : PSimpul;
    gg : PGaris;
    cek : integer;
begin
    n := 0;
    ResetNol(SS);
    HitungODSimpul(SS);
    HitungIDSimpul(SS);
    S := SS;
    Cek := CekGraph(SS);
    petunjuk('Flowgraph sedang dicek kelengkapannya ');
    if cek = 0 then
    begin
        clrscr;
        petunjuk('Daftar basis set lintasan ');
        CariLintasan(SS,SP,k);
        write('Harga Kompleksitas : ',HitungKompleksitas(SS));
        readln;
    end
    else
    begin
        peringatan('Flowgraph belum memenuhi asumsi, tekan <enter> ');
        textbackground(3);textcolor(4);
        gotoxy(17,13);write(' ');
        gotoxy(17,14);
        case cek of
            1 : write(' Terdapat masalah pada simpul stop ');
            2 : write(' Terdapat lebih dari satu simpul start ');
            3 : write(' Terdapat siklus ');
            else
                write(' Flowgraph sama dengan nil ');
        end;
        gotoxy(17,15);write(' Silahkan lengkapi dari menu edit ');
        gotoxy(17,16);write(' ');
        textbackground(1);textcolor(7);readln;
    end;
end;

```

```

procedure EditTambahGaris(SS : PSimpul; var SA : PSimpul);
begin
  petunjuk(' Tekan <enter> dan ketikkan nama simpul yang akan disisipkan ');
  DaftarSimpul(SS);
  EditMasukan(SS,SA);
end;

```

```

procedure EditHapusSimpul(SS : PSimpul; var SA : PSimpul);
var T : string3;
begin
  petunjuk(' Tekan <enter> dan ketikkan nama simpul yang akan dihapus');
  DaftarSimpul(SS);
  textbackground(2);textcolor(14);
  gotoxy(20,4);write(' ');
  gotoxy(20,5);write(' Masukan nama simpul yang akan dihapus ');
  gotoxy(20,6);write('          Simpul : ');
  gotoxy(20,7);write(' ');
  gotoxy(38,6);readln(T);textbackground(1);textcolor(7);
  HapusSimpul(SS,SA,T);
end;

```

```

procedure EditHapusGaris(SS : PSimpul; var SA : PSimpul);
var T,G : string3;
begin
  petunjuk(' Tekan <enter> dan ketikkan garis yang akan dihapus');
  gotoxy(24,1);write('MASUKAN GARIS YANG AKAN DIHAPUS');
  gotoxy(3,2);write('—GARIS HAPUS—');
  DaftarSimpul(SS);
  gotoxy(14,3);readln(T);
  gotoxy(14,4);readln(G);
  HapusGaris(SS,T,G);
  KembaliLayar(2,6,77,22,LayarMasukan);
  DaftarSimpul(SS);
end;

```

```

procedure FormMenuUtama;
var i : byte;
begin
  textbackground(1);clrscr;
  Petunjuk('Gerakkan kursor '+chr(24)+' atau '+chr(25)+' atau tekan huruf
           dengan warna berbeda ');
  gotoxy(12,2);
  write('APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH');
  gotoxy(20,3);write('DENGAN ALGORITMA DEPTH FIRST SEARCH');
  gotoxy(22,7);write('MENU UTAMA');
  gotoxy(22,8);write(' ');gotoxy(53,8);write(' ');
  for i:= 1 to 7 do
  begin
    textcolor(7);
    gotoxy(22,i+8);write(' ');gotoxy(53,i+8);write(' ');
    with DataMenu do
    begin
      gotoxy(27,posisiMenu[i]);textbackground(1);
      textcolor(7);write(IsiMenu[i]);
      gotoxy(posisiKarakterMenu[i],i+8);
      textcolor(14);write(KarakterMenu[i]);
    end;
  end;
  textcolor(7);gotoxy(22,i+10);write(' ');
  gotoxy(22,i+9);write(' ');gotoxy(53,i+9);write(' ');
  move(layar,LayarMenuUtama,4000);
end;

```

```

procedure FormMenuEdit;
var i : byte;
begin
  textbackground(1);clrscr;
  Petunjuk('Gerakkan kursor '+chr(24)+' atau '+chr(25)+' atau tekan huruf
    dengan warna berbeda ');
  gotoxy(12,2);
  write('APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH');
  gotoxy(20,3);write('DENGAN ALGORITMA DEPTH FIRST SEARCH');
  gotoxy(22,7);write('MENU EDIT');
  gotoxy(22,8);write('');
  gotoxy(52,8);write('');
  for i:= 1 to 4 do
  begin
    textcolor(7);
    gotoxy(22,i+8);write('');
    gotoxy(52,i+8);write('');
    with DataMenuEdit do
    begin
      gotoxy(27,posisiMenu[i]);textbackground(1);
      textcolor(7);write(IsiMenu[i]);
      gotoxy(posisiKarakterMenu[i],i+8);
      textcolor(14);write(KarakterMenu[i]);
    end;
  end;
  textcolor(7);gotoxy(22,i+10);write('');
  gotoxy(22,i+9);write('');
  gotoxy(52,i+9);write('');
  move(Layar,LayarMenuEdit,4000);
end;

procedure FormMasukan;
var i:byte;
begin
  textbackground(1);clrscr;
  gotoxy(33,1);write('MASUKAN SIMPUL');
  gotoxy(3,2);write('');
  gotoxy(3,3);write(' Asal : ');
  gotoxy(3,4);write(' Tujuan : ');
  gotoxy(3,5);write('');
  gotoxy(3,7);write(' DAFTAR SIMPUL ');
  for i:= 8 to 21 do
  begin
    gotoxy(3,i);write('|');
    gotoxy(78,i);write('|');
  end;
  gotoxy(3,22);write('');
  move(Layar,LayarMasukan,4000);
end;

procedure Edit(SS:PSimpul;var SA:PSimpul);
var baris1,pilih,PilihPrev,k1,k2,m: byte;
    T,G : string3;
    lagi,TekanMenu : char;
begin
  m := 1;
  k1 := 18;k2 := 37;
  KembalikanLayar(1,1,80,25,LayarMenuEdit);
  petunjuk(' ');
  pilih:=1;
  baris1:=DataMenuEdit.PosisiMenu[pilih];
  ChAtributLayar(27,baris1,49,baris1,$70);

```

```

repeat
TekanMenu := ' ';
PilihPrev := Pilih;
gotoxy(27,baris1);
repeat
n := n+1;
if n >= 255 then n := 21;
KembalikanLayar(k1,4,k1,4,LayarMenuEdit);
KembalikanLayar(k2,4,k2,4,LayarMenuEdit);
if K1 > 56 then k1 := 18
else k1 := k1+1;
if K2 > 56 then k2 := 18
else k2 := k2+1;
ChAtributLayar(k1,4,k1,4,$1A);
ChAtributLayar(k2,4,k2,4,$1A);
if n >= 20 then ChrJalan(m,LayarMenuEdit)
else
begin
m := 1;
petunjuk((IsiMenuEdit[pilih]));
end;
gotoxy(27,baris1);
delay(70);
until KeyPressed;
TekanMenu := readkey;
if TekanMenu <> #0 then
begin
case upcase(TekanMenu) of
'I': pilih:=1;
'D': pilih:=2;
'S': pilih:=3;
'H': pilih:=4;
'G': pilih:=5;
'E': pilih:=6;
#13: pilih:=PilihPrev;
else
write(#7);
end;
end;
if TekanMenu = #0 then
begin
TekanMenu := readkey;
case TekanMenu of
#72 : begin
if pilih = 1 then pilih := 6
else pilih := pilih-1;
end;
#80 : begin
if pilih = 6 then pilih := 1
else pilih := pilih+1;
end;
else write(#7);
end;
end;
if Pilih <> PilihPrev then
begin
baris1 := DataMenuEdit.PosisiMenu[PilihPrev];
KembalikanLayar(27,baris1,50,baris1,LayarMenuEdit);
baris1 := DataMenuEdit.PosisiMenu[pilih];
ChAtributLayar(27,baris1,49,baris1,$70);
end;
if (upcase(TekanMenu) = 'E') or ((pilih = 6) and (TekanMenu = #13)) then
exit;

```

```

KembalikanLayar(30,20,65,20,LayarMenuEdit);
if TekanMenu = #13 then
begin
  KembalikanLayar(1,1,80,25,LayarMasukan);
  if pilih = 1 then InformasiMenuEdit;
  if pilih = 2 then DaftarSimpul(SS);
  if pilih = 3 then EditTambahGaris(SS,SA);
  if pilih = 4 then EditHapusSimpul(SS,SA);
  if pilih = 5 then EditHapusGaris(SS,SA);
  KembalikanLayar(1,1,80,25,LayarMenuEdit);
  petunjuk(' ');
end;
until false;
end;

procedure MenuUtama;
var i,j,k1,k2,m,pilih,pilihprev,
    baris1,n      :byte;
    TekanMenu,selesai,t  : char;
begin
  m := 1;k1 := 18;k2 := 37;
  KembalikanLayar(1,1,80,25,LayarMenuUtama);
  petunjuk(' ');
  pilih:=1;
  baris1:=DataMenu.PosisiMenu[pilih];
  ChAtributLayar(27,baris1,49,baris1,$70);
  repeat
    TekanMenu := ' ';
    PilihPrev := Pilih;
    gotoxy(28,baris1);
    repeat
      n := n+1;
      if n >= 255 then n := 21;
      KembalikanLayar(k1,4,k1,4,LayarMenuUtama);
      KembalikanLayar(k2,4,k2,4,LayarMenuUtama);
      if K1 > 56 then k1 := 18
      else k1 := k1+1;
      if K2 > 56 then k2 := 18
      else k2 := k2+1;
      ChAtributLayar(k1,4,k1,4,$1A);ChAtributLayar(k2,4,k2,4,$1A);
      if n >= 20 then ChrJalan(m,LayarMenuUtama)
      else
        begin
          m := 1;
          petunjuk(IsimenuUtama[pilih]);
        end;
      gotoxy(28,baris1);delay(70);
    until KeyPressed;
    TekanMenu := readkey;
    if TekanMenu <> #0 then
      begin
        case upcase(TekanMenu) of
          'I': pilih:=1;
          'M': pilih:=2;
          'B': pilih:=3;
          'D': pilih:=4;
          'S': pilih:=5;
          'C': pilih:=6;
          'E': pilih:=7;
          #13: pilih:=PilihPrev;
          else write(#7);
        end;
      end;
end;

```

```

if TekanMenu = #0 then
begin
TekanMenu := readkey;
case TekanMenu of
#72 : begin
if pilih = 1 then pilih := 7
else pilih := pilih-1;
end;
#80 : begin
if pilih = 7 then pilih := 1
else pilih := pilih+1;
end;
else write(#7);
end;
end;
if Pilih <> PilihPrev then
begin
baris1 := DataMenu.PosisiMenu[PilihPrev];
KembalikanLayar(27,baris1,50,baris1,LayarMenuUtama);
baris1 := DataMenu.PosisiMenu[pilih];
ChAtributLayar(27,baris1,49,baris1,$70);
end;
if (upcase(TekanMenu) = 'E') or ((pilih = 7) and (TekanMenu = #13)) then
begin
selesai := ' ';
while not ((selesai = 'Y') or (selesai = 'T')) do
begin
textcolor(14);
gotoxy(28,20);write(' Anda yakin (Y/T) ? ');read(selesai);
selesai := upcase(selesai);
end;
if selesai = 'Y' then halt;
KembalikanLayar(29,20,70,20,LayarMenuUtama);
end;
if TekanMenu = #13 then
begin
if pilih = 1 then InformasiMenuUtama;
if pilih = 2 then Masukan(SS,SA);
if pilih = 3 then BukaFile(SS,SA);
if pilih = 4 then Edit(SS,SA);
if pilih = 5 then SimpanKeFile(SS);
if pilih = 6 then CetakLintasan(SS);
KembalikanLayar(1,1,80,25,LayarMenuUtama);
petunjuk(' ');
end;
until false;
end;

```

```

rocedure Cover;
egin
  textbackground(0);textcolor(7);clrscr;
  gotoxy(1,5);
  writeln(' ':8, ' ');
  writeln(' ':8, ' Aplikasi untuk Mendapatkan Satu Basis Set Lintasan Flowgraph ');
  writeln(' ':8, ' ');
  writeln(' ':8, ' Disusun oleh : ');
  writeln(' ':8, ' Sulardi ( J2A 096 060 ) ');
  writeln(' ':8, ' ');
  writeln(' ':8, ' Jurusan Matematika ');
  writeln(' ':8, ' Fakultas Matematika dan Ilmu Pengetahuan Alam ');
  writeln(' ':8, ' Universitas Diponegoro ');
  writeln(' ':8, ' Semarang ');
  writeln(' ':8, ' 2001 ');
  writeln(' ':8, ' ');
  TungguTekan(10);
nd;

```

```

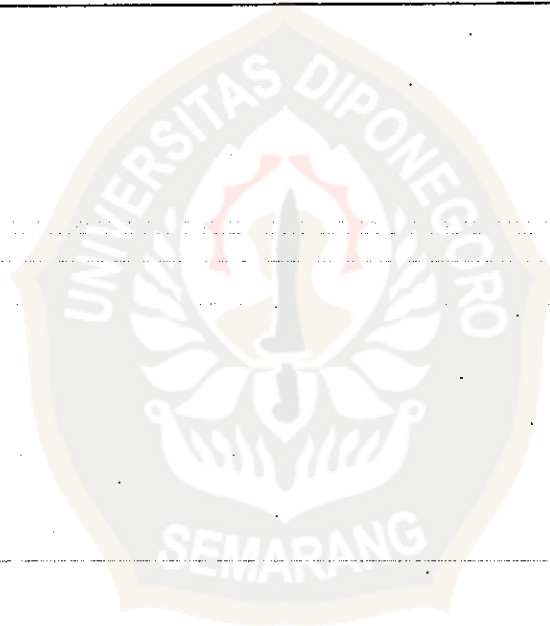
*** program utama ****

```

```

egin
  clrscr;
  Cover;
  AturanPenggunaan;
  FormMenuUtama;
  FormMenuEdit;
  FormMasukan;
  k := 1;
  menuutama;
nd.

```



Lampiran 6 : Tampilan Keluaran Aplikasi untuk Mendapatkan Satu Basis Set Lintasan Flowgraph

```
APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH
DENGAN ALGORITMA DEPTH FIRST SEARCH
COPYRIGHT © 2011, ALL RIGHTS RESERVED BY THE AUTHOR

=====
MENU UTAMA
=====
Masukan dari Keyboard
Buka File
Edit Flowgraph
Simpan ke File
Cetak Lintasan
Keluar
=====

Gerakkan kursor | atau | atau tekan huruf dengan mouse
```

Tampilan Menu Utama

```
APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH
DENGAN ALGORITMA DEPTH FIRST SEARCH
COPYRIGHT © 2011, ALL RIGHTS RESERVED BY THE AUTHOR

=====
MENU EDIT
=====
Masukan dari Keyboard
Tampilkan Daftar Simpul
Sisip Simpul / Garis
Hapus Simpul
Hapus Garis
Ke Menu Utama
=====

Gerakkan kursor | atau | atau tekan huruf dengan mouse
```

Tampilan Menu Edit

PETUNJUK PENGGUNAAN PROGRAM

Petunjuk Pemberian Masukan

1. Masukan yang diberikan berupa karakter ASCII untuk memanasi simpul flowgraph
2. Syarat-syarat masukan agar dapat diproses dan memberikan keluaran :
 - a) Terdapat sebuah simpul start dan sebuah simpul stop
 - b) dari simpul start, terdapat sebuah lintasan ke masing-masing simpul
 - c) dari masing-masing simpul, terdapat sebuah lintasan ke simpul stop
3. Masukan data dapat diberikan dengan membaca file yang menyimpan data flowgraph. Pembongkaran file dilakukan dari pilihan menu Buka File
4. Untuk memulai memasukkan data flowgraph harus dimulai dari pilihan menu Masukan dari Keyboard tidak boleh dari menu Edit Flowgraph
5. Agar data flowgraph aman, sebelum melakukan proses lebih lanjut, simpan data flowgraph dalam bentuk file dari pilihan menu Simpan ke File

Petunjuk Proses Cetak Lintasan

Program akan memberikan keluaran satu basis set lintasan flowgraph apabila masukan flowgraph telah memenuhi syarat-syarat masukan di atas. Apabila masukan flowgraph belum lengkap, maka program akan meminta pemakai untuk melengkapinya dari menu edit

Tekan sembarang tombol ...

Tampilan Aturan Pemakaian Aplikasi

INFORMASI MENU

Masukan dari Keyboard

Membaca masukan flowgraph dari papan ketik apabila terdapat data flowgraph pada memori akan dihapus

Buka File

Memberi masukan flowgraph dengan membuka file yang menyimpan data flowgraph nama file dan path penyimpanan diketikkan pada tempat yang disediakan

Edit Flowgraph

Melakukan editing terhadap masukan flowgraph berupa penambahan, dan penghapusan simpul atau garis

Simpan ke File

Menyimpan flowgraph yang ada di memori ke media penyimpanan disket atau hard disk

Cetak Lintasan

Menampilkan keluaran proses berupa satu set lintasan dan nilai kompleksitas siklikomatik dari flowgraph tersebut

Keluar

Membali ke DOS data flowgraph di memori otomatis hilang

Tekan <enter>

Tampilan Informasi Menu Utama


```

APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH
DENGAN ALGORITMA DEPTH FIRST SEARCH
COPY RIGHT (c) 2009 ERIANDI N N P UNDP

===== MENU UTAMA =====
Informasi Menu
Masukan dari Keyboard
Buka File
Edit Flowgraph

Flowgraph belum memenuhi asumsi, tekan <enter>

```

**Tampilan peringatan bahwa flowgraph masih belum memenuhi asumsi
(belum ada data flowgraph di memori)**

```

APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH
DENGAN ALGORITMA DEPTH FIRST SEARCH
COPY RIGHT (c) 2009 ERIANDI N N P UNDP

===== MENU UTAMA =====
Informasi Menu
Masukan dari Keyboard
Buka File
Edit Flowgraph

Flowgraph belum memenuhi asumsi, tekan <enter>

```

**Tampilan peringatan bahwa flowgraph masih belum memenuhi asumsi
(terdapat masalah pada simpul stop)**

APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH
 DENGAN ALGORITMA DEPTH FIRST SEARCH

```

.....
MENU UTAMA
.....
Informasi Menu
Masukan dari Keyboard
Buka File
Edit Flowgraph
.....

```

Flowgraph belum memenuhi asumsi, tekan <enter>

Tampilan peringatan bahwa flowgraph masih belum memenuhi asumsi
 (terdapat siklus pada flowgraph)

APLIKASI PENCARIAN SATU BASIS SET LINTASAN FLOWGRAPH
 DENGAN ALGORITMA DEPTH FIRST SEARCH

```

.....
MENU UTAMA
.....
Informasi Menu
Masukan dari Keyboard
Buka File
Edit Flowgraph
.....

```

Flowgraph belum memenuhi asumsi, tekan <enter>

Tampilan peringatan bahwa flowgraph masih belum memenuhi asumsi
 (terdapat masalah pada simpul start)