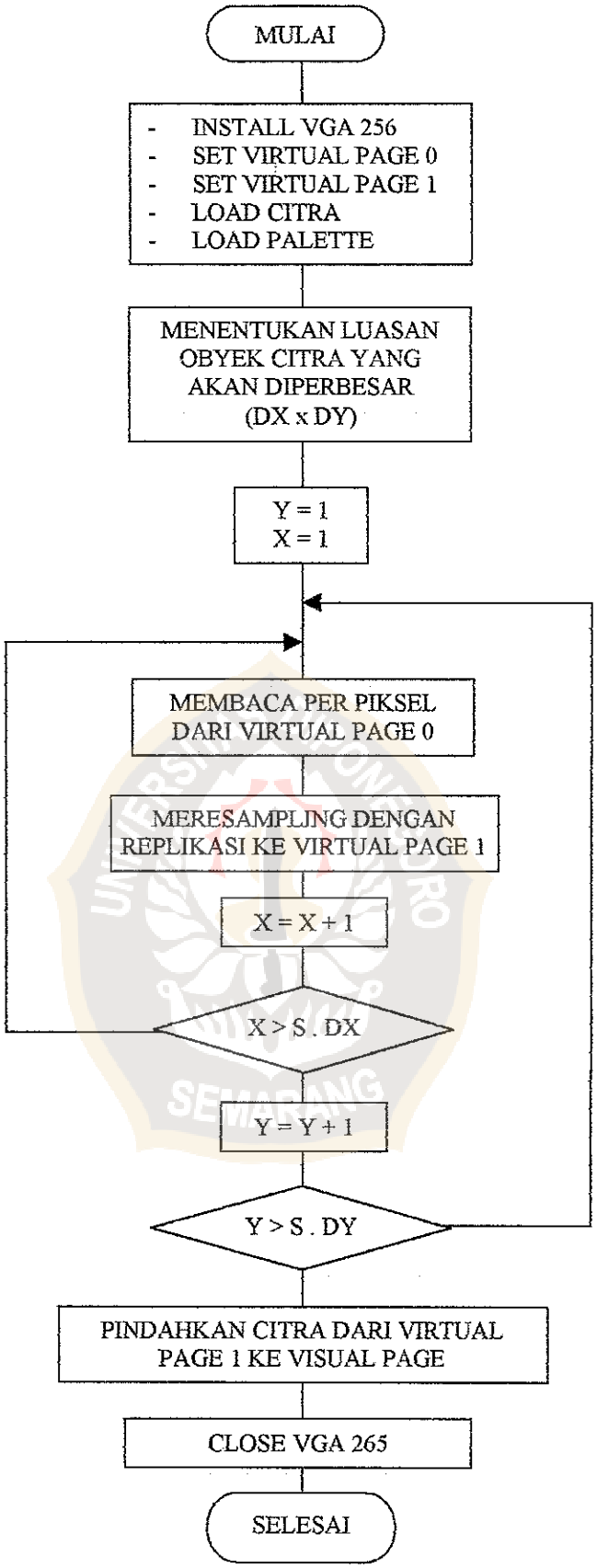
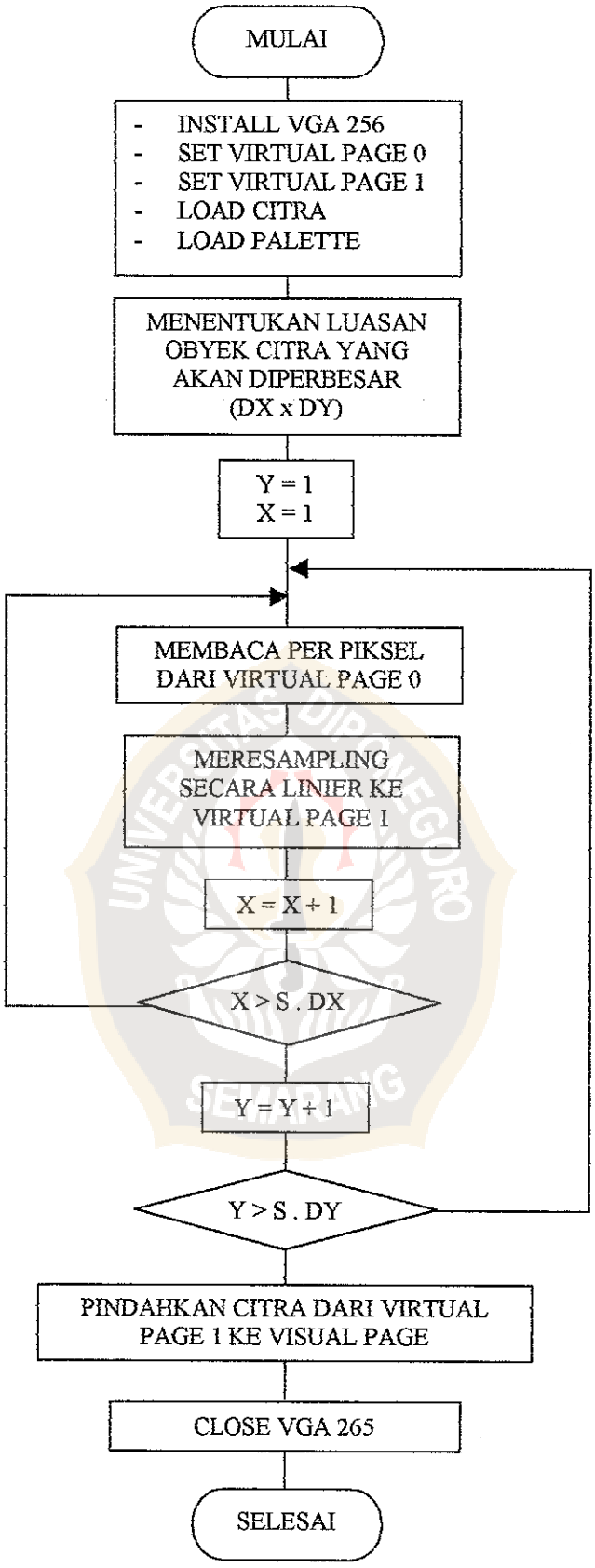
LAMPIRAN A

DIAGRAM ALIR PROGRAM PENSKALAAN

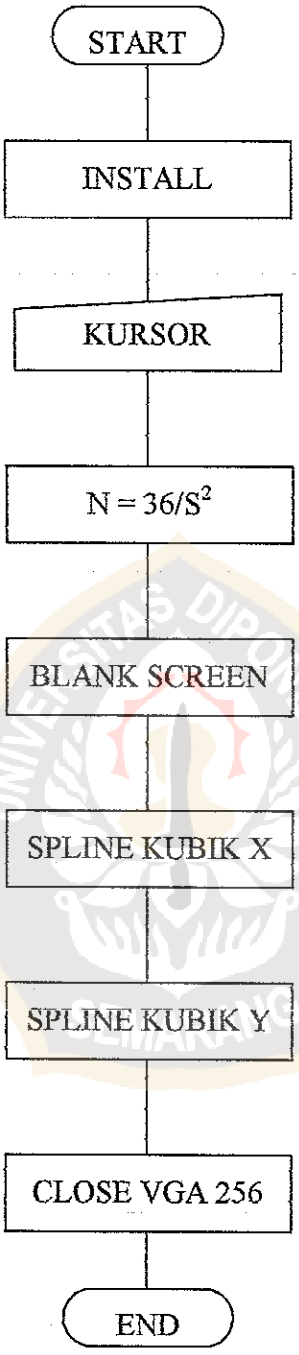
PROSEDUR INTERPOLASI TETANGGA TERDEKAT



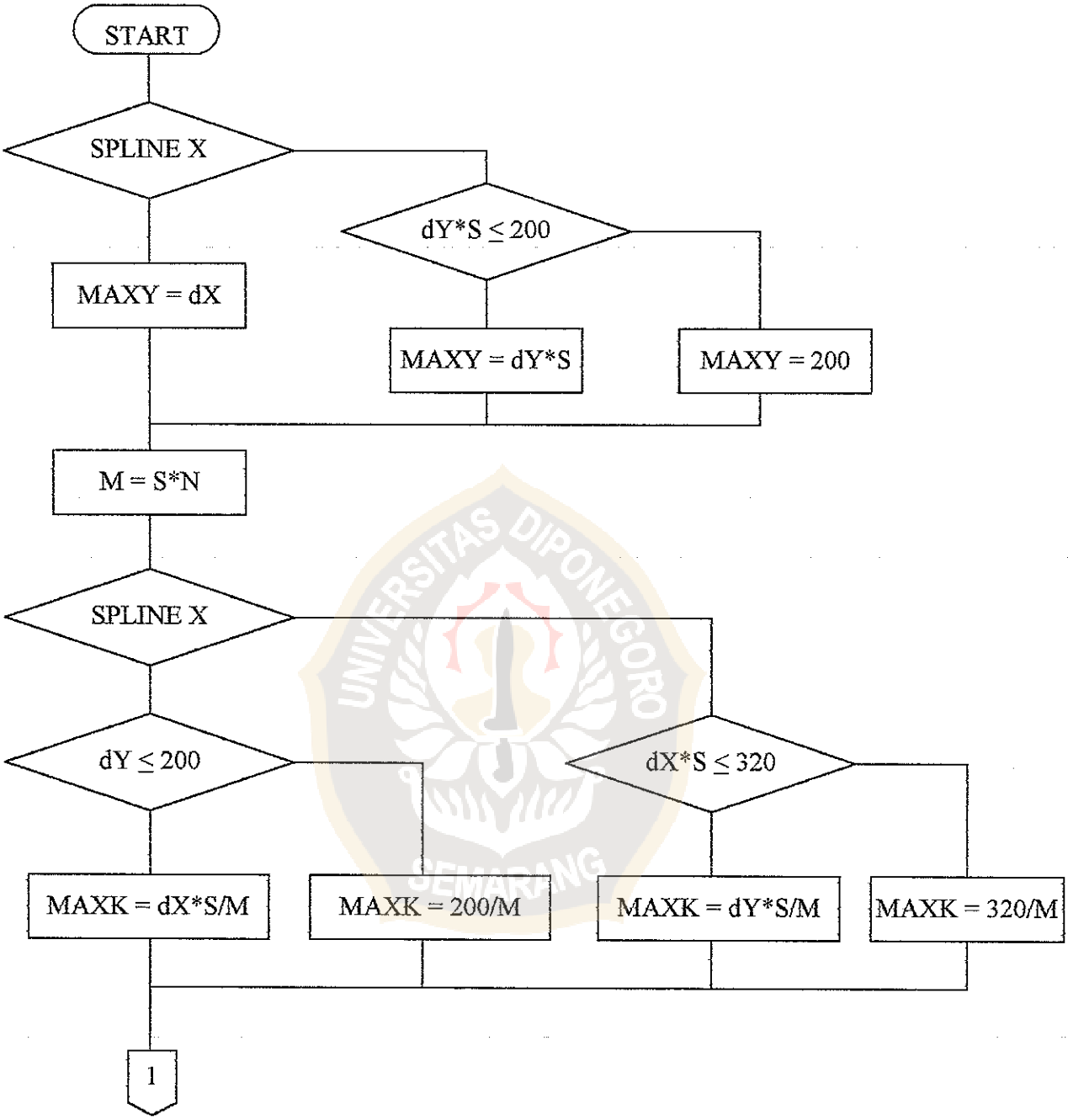
PROSEDUR INTERPOLASI BILINIER

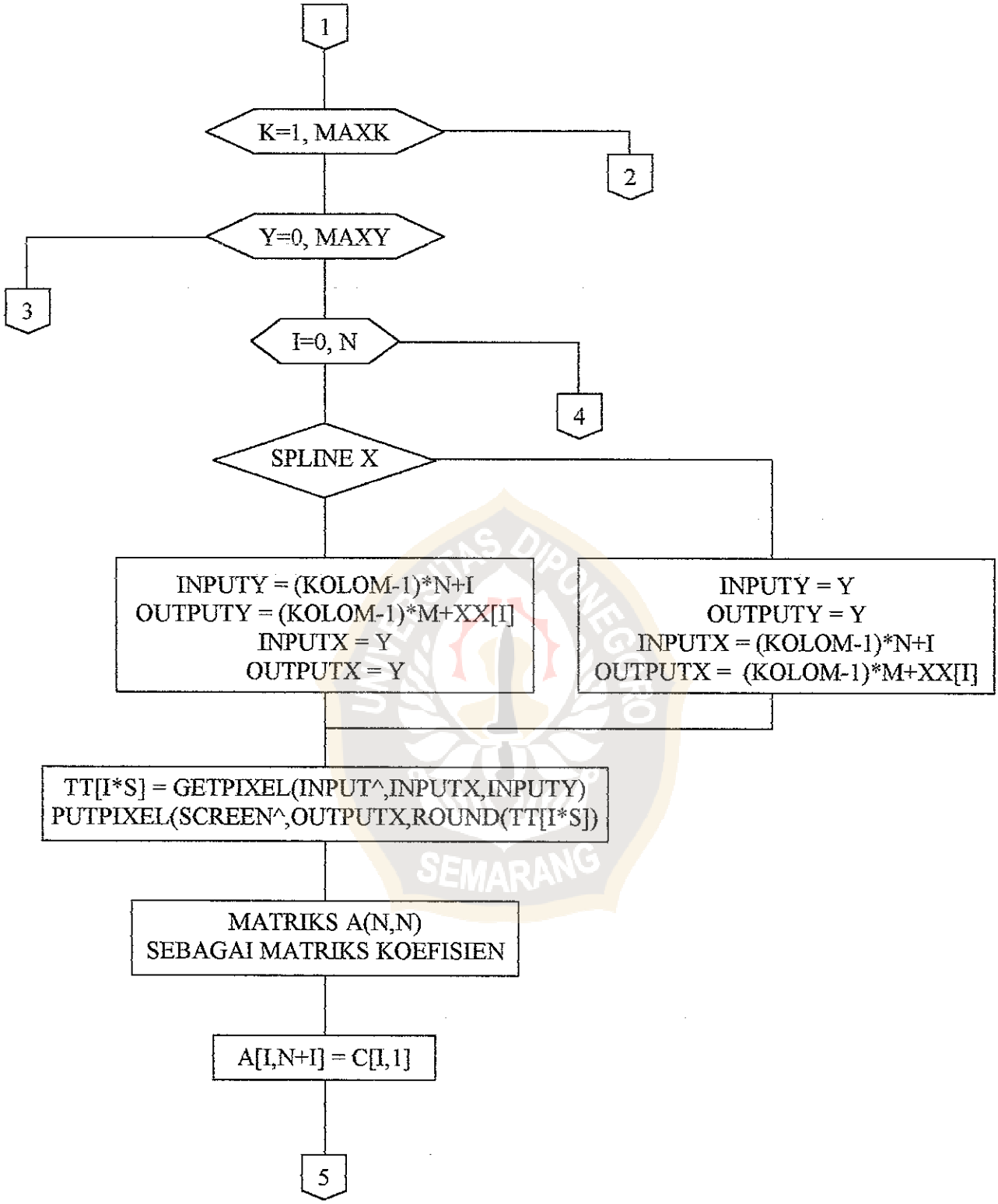


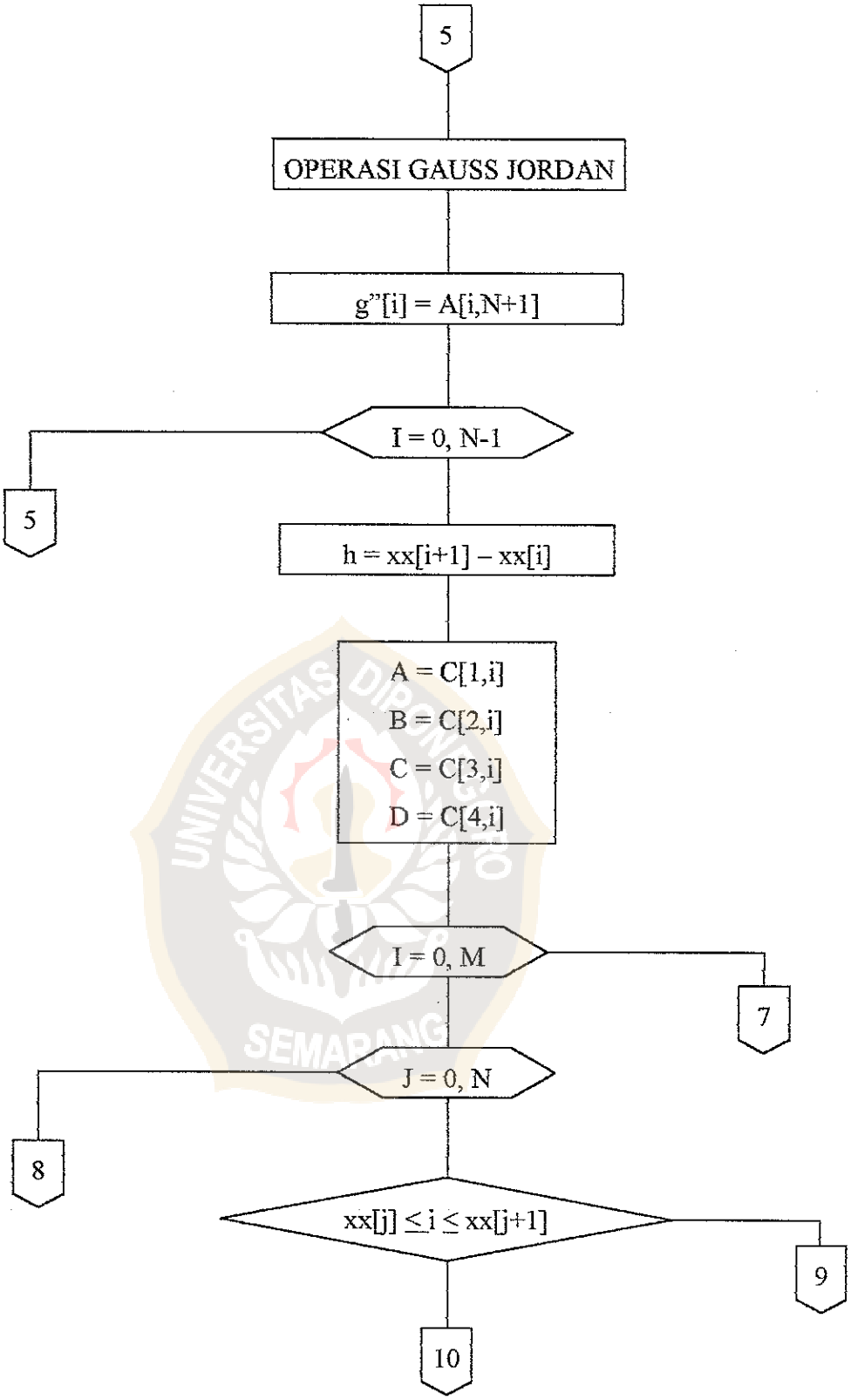
PROSEDUR BIKUBIK



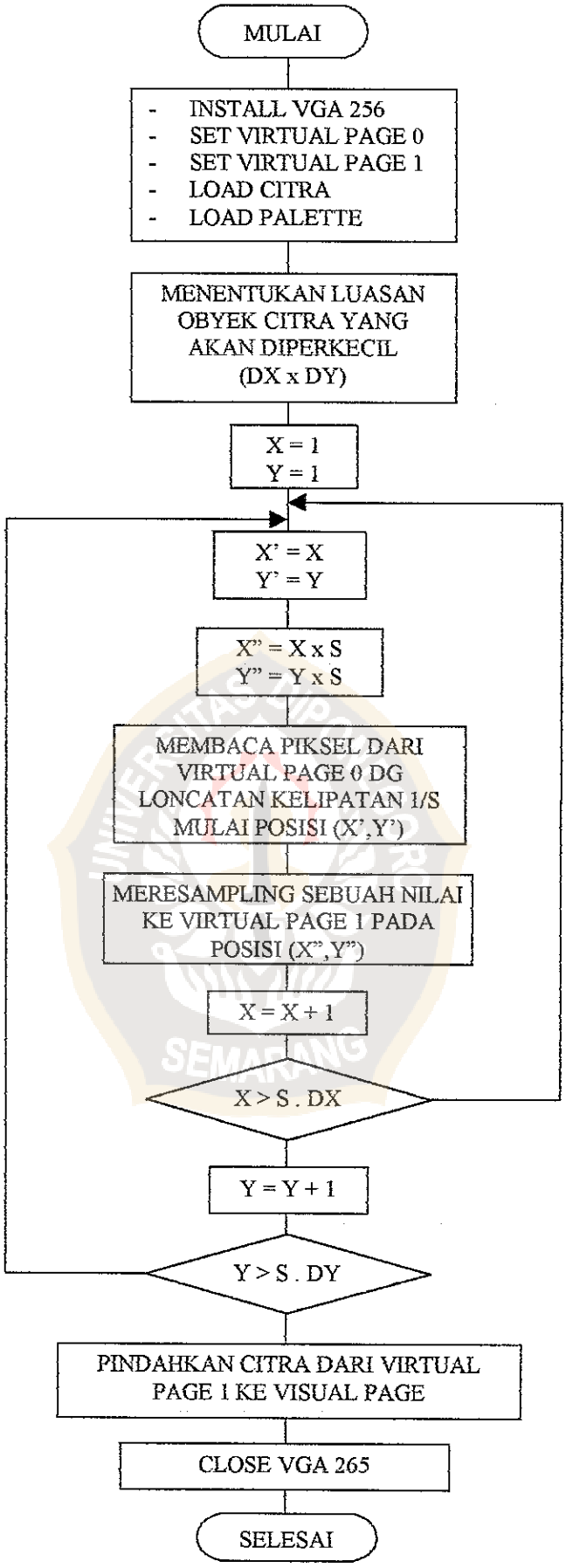
PROSEDUR
SPLINE BIKUBIK



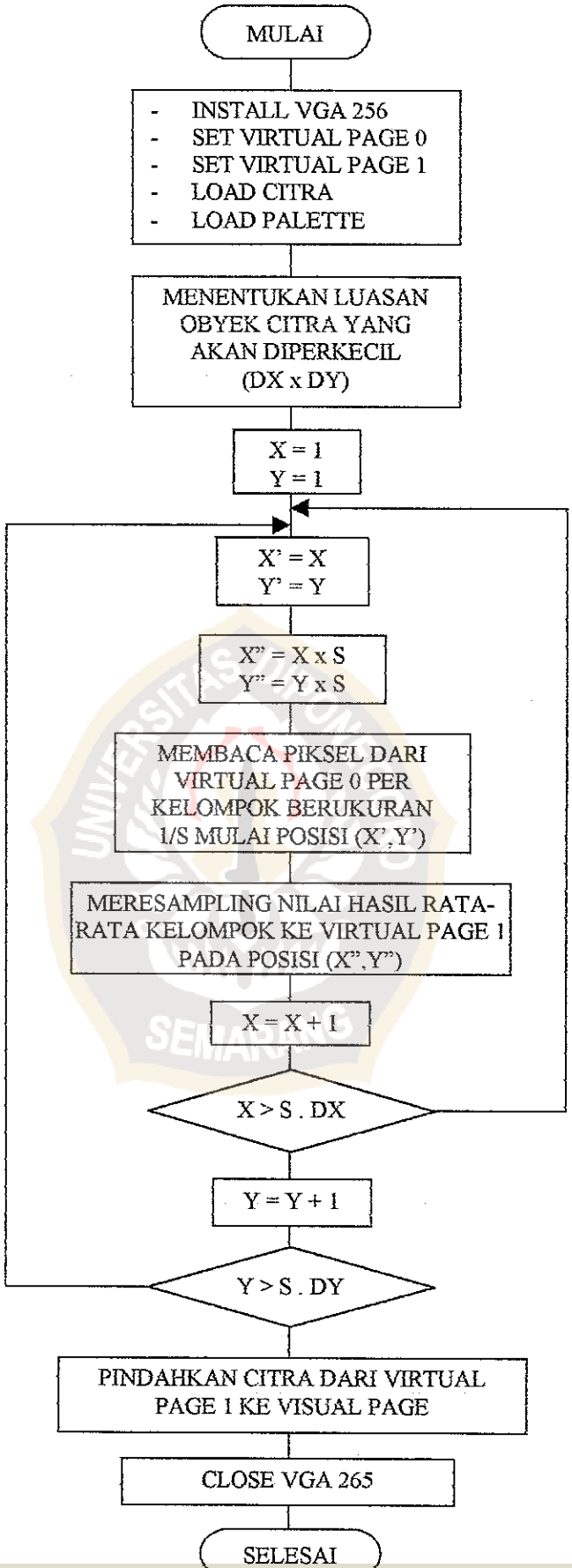




PROSEDUR REDUKSI LONCATAN



PROSEDUR REDUKSI RATA-RATA



LAMPIRAN B

LISTING PROGRAM PENSKALAAN

```
uses crt,graph,pen2,menu1,menu2,slides;

type
    txt1 = array[1..5] of string[20];
    txt2 = array[1..2] of string[20];
    txt3 = array[1..3] of string[20];
var
   drv,md  : integer;
   pilih   : array[1..5] of atrPilihan;
   pilihM2 : array[1..2] of atrPilihan;
   short   : array[1..3] of atrPilihan;

   text1 : txt1;   c:char;
   text2 : txt2;
   text3 : txt3;

   kembali : boolean;
label
     first,f_three;

Procedure justify;
var
   x,y,dx,dy : integer;
   larik,cacah : byte;
begin
     { -string map 1- }
     text1[1]:='Intisari';
     text1[2]:='Menu Proses >';
     text1[3]:='Petunjuk';
     text1[4]:='Tentang Program';
     text1[5]:='Keluar';

     { -string map 2- }
     text2[1]:='Menu Umum';
     text2[2]:='Perbandingan';

     { -string map 3- }
     text3[1]:='F1 : Petunjuk';
     text3[2]:='F2 : Slide';
     text3[3]:='F3 : Keluar';

     { -justify map 1- }
     x:=5;y:=20;dx:=120;dy:=30;

     for larik:= 1 to 5 do
         atur_pilihan(x,y+dy*(larik-1),dx,dy,pilih[larik]);

     { -justify map2- }
     for cacah:= 1 to 2 do
         atur_pilihan(x+dx+1,y+dy+dy*(cacah-1),dx-
15,dy,pilihM2[cacah]);

     { -justify short- }
     x:=50;y:=440;dx:=100;dy:=30;

     for larik:= 1 to 3 do
```

```
          atur_pilihan(x+(dx+20)*(larik-1),y,dx,dy,short[larik]);
end;

Procedure Tampilan;
var
   larik,cacah : byte;

begin
     { -background- }
     setfillstyle(1,3);
     bar(0,0,getmaxX,getmaxY);
     setfillstyle(1,7);

     { -map 1- }
     with pilih[1] do map(false,1,x,y,dx+x,(5*dy+y));
     for larik:= 1 to 4 do
         buat_pilihan(pilih[larik]);

     { -short_ }
     map(false,1,5,430,630,479);
     for larik:= 1 to 3 do
         buat_short(short[larik]);

     settextstyle(2,0,4);
     for larik:= 1 to 5 do
         tulis_pilihan(pilih[larik],text1[larik],1);

     for larik:= 1 to 3 do
         tulis_pilihan(short[larik],text3[larik],1);
end;

Procedure HidupkanM3;
var cacah: byte;
begin
     settextstyle(2,0,4);
        setfillstyle(1,7);
     with pilihM2[1] do
          map(false,1,x,y,x+dx,y+dy*2);
     buat_pilihan(pilihM2[1]);
     for cacah:= 1 to 2 do
         tulis_pilihan(pilihM2[cacah],text2[cacah],1);
end;

Procedure MatikanM3;
begin
     setfillstyle(1,3);
     with pilihM2[1] do
          bar(x,y,x+dx,y+dy*2);
end;

Procedure deteksi_Tombol(var kembali:boolean;var c:char);
const
   redup  = false;
   terang = true;
   down   = #80;
   up     = #72;
```

```
    right  = #77;
    left   = #75;
    enter  = #13;
    tab    = #9;
    f1     = #59;
    f2     = #60;
    f3     = #61;

var
    lm1,m1,
    lm2,m2,
    lm3,m3 : byte;
label
    f_one,f_two;
begin
    lm1:=1;m1:=1;lm2:=1;m2:=1;lm3:=0;m3:=0;
    repeat
      c:=readkey;
      if c=#0 then
      begin
           c:=readkey;
           case c of
             down : begin
                    if m1= 1 then
                    begin
                     if m3 <> 0 then
                     begin
                       if m3 = 2 then m3:=1
                                 else m3:=2;
                       lighting_pilihan(pilihM2[lm3],redup);
                       tulis_pilihan(pilihM2[lm3],text2[lm3],1);
                       lighting_pilihan(pilihM2[m3],terang);
                       tulis_pilihan(pilihM2[m3],text2[m3],3);
                       lm3:= m3;
                     end
                     else
                     begin
                       if m2 = 5 then m2:= 1
                                 else inc(m2);
                       lighting_pilihan(pilih[lm2],redup);
                       tulis_pilihan(pilih[lm2],text1[lm2],1);
                       lighting_pilihan(pilih[m2],terang);
                       tulis_pilihan(pilih[m2],text1[m2],3);
                       lm2:= m2;
                     end;
                    end;
                    end;

              up   : begin
                    if m1= 1 then
                    begin
                     if m3 <> 0 then
                     begin
                       if m3 = 1 then m3:=2
                                 else m3:=1;
                       lighting_pilihan(pilihM2[lm3],redup);
```

```
            tulis_pilihan(pilihM2[lm3],text2[lm3],1);
            lighting_pilihan(pilihM2[m3],terang);


            tulis_pilihan(pilihM2[m3],text2[m3],3);
            lm3:= m3;
          end
          else
          begin
            if m2 = 1 then m2:= 5
                      else dec(m2);
            lighting_pilihan(pilih[lm2],redup);
            tulis_pilihan(pilih[lm2],text1[lm2],1);
            lighting_pilihan(pilih[m2],terang);
            tulis_pilihan(pilih[m2],text1[m2],3);
            lm2:= m2;
          end;
        end;
        end;

right : begin
        if m1 = 1 then
        begin
         if m2 = 2 then
         begin
           if m3 = 0 then
           m3:= 1;
           hidupkanm3;
           lighting_pilihan(pilihM2[1],terang);
           tulis_pilihan(pilihM2[1],text2[1],3);
           lm3:= m3;
         end;
        end;
        end;

 left : begin
        if m1 = 1 then
        begin
         if m3 <> 0 then
         begin
           m3:= 0;
           matikanM3;
         end;
        end;
        end;

  f1  : begin
        f_one:
        if m3 <> 0 then
        begin
             m3 := 0;
             matikanM3;
        end;
        lighting_pilihan(pilih[m2],redup);
        tulis_pilihan(pilih[m2],text1[m2],1);
```

```
                lighting_pilihan(pilih[3],terang);
                tulis_pilihan(pilih[3],text1[3],3);
                lm2:=3;m2:=3;
                help(200,100,c);
                end;

          f2 : begin
                f_two:
                slide;
                kembali:= true;
                exit;
               end;
       end;
end
else
begin
    case c of
    enter   : begin
               if m1= 2 then goto f_one;
               if m1= 3 then goto f_two;
               if m1= 4 then c:= f3;
               if m1= 1 then
               begin
                if m3 <> 0 then
                begin
                 if m3= 1 then begin
                                menu01;
                                kembali:= true;
                                exit;
                               end;
                 if m3= 2 then begin
                                menu02;
                                kembali:= true;
                                exit;
                               end;
                end
                else
                begin
                 if m2= 1 then intisari(160,50,c);
                 if m2= 3 then help(200,100,c);
                 if m2= 4 then about(200,100,c);
                 if m2= 5 then c:= f3;
                end;
               end;
              end;
         tab : begin
               if m1= 4 then m1:= 1
                        else inc(m1);
               if lm1 > 1 then
                    tulis_pilihan(short[lm1-1],text3[lm1-1],1)
                      else
                      begin
                   lighting_pilihan(pilih[lm2],redup);
                   tulis_pilihan(pilih[lm2],text1[lm2],1);
                        m2:= 1;
```

```
                              lm2:= m2;
                            end;
                    if m3 <> 0 then
                    begin
                         m3:= 0;
                         matikanM3;
                    end;
                    if m1 <> 1 then tulis_pilihan(short[m1-
1],text3[m1-1],2)
                              else begin

lighting_pilihan(pilih[m2],terang);

tulis_pilihan(pilih[m2],text1[m2],3);
                                   end;
                    lm1:= m1;
                    end;
          end;
     end;
   until c=f3;
end;


begin
     drv:= detect;
     initgraph(drv,md,'c:\progra~2\units');
     kembali:= false;
     first:
     justify;
     tampilan;
     if kembali=false then about(200,100,c);
     deteksi_tombol(kembali,c);
     if c=#61 then goto f_three;
     if kembali then goto first;
     f_three:
     closegraph;
     textmode(80);
end.
```

```
unit menu1;
interface

procedure menu01;

implementation

uses crt,graph,pen2,map_bar,life,citra;
const
     Esc   = #27;
     Enter = #13;
     Tab   = #9;
var
   drv,md       : integer;
   map          : byte;
   sub_2,sub_3  : byte;
   skala        : real;
   text_box1,text_box2,text_box3 : text;
   c,c1                          : char;


Procedure DeteksiTombol(var map:byte;var skala:real; var
sub_2,sub_3:byte);
const
     F1    = #59;
     F2    = #60;
     Right = #77;
     Left  = #75;
     Up    = #72;
     Down  = #80;
     Metode2 : array[1..3] of string
             = ('Tetangga terdekat','Bilinier','Bikubik');
     Metode1 : array[1..2] of string
             = ('Loncatan','Rata-rata');
     Metode3 : array[1..2] of string
             = ('Linear Add Intensity','Scalling Intensity');
var
   M0_1,M0_2  : byte;
   sub_1      : byte;
   sub,csub_3 : byte;
   st         : string;
   m1         : byte;
   err        : integer;
   gambar     : string;

Procedure Reset;
var i  : byte;
    st : string;
begin
     case map of
     1 : begin
          for i:=1 to 2 do
           text_box3[i]:=metode1[i];
          for i:=1 to 7 do
          begin
           str(skala1[i],st);
```

```
            text_box1[i] := st+' %';
           end;
           for i:=1 to maxi do
           begin
            text_box2[i] := pcitra[i];
           end;
           skala:=skala1[1];
          end;
     2 : begin
           for i:=1 to 3 do
            text_box3[i]:=metode2[i];
           for i:=1 to 7 do
           begin
            str(skala2[i],st);
            text_box1[i] := st+' %';
           end;
           for i:=1 to maxi do
           begin
            text_box2[i] := pcitra[i];
           end;
           skala:=skala2[1];
          end;
     3 : begin
           for i:=1 to 2 do
            text_box3[i]:=metode3[i];
           for i:=1 to 7 do
           begin
            str(skala3[i],st);
            text_box1[i] := st;
           end;
           for i:=1 to maxi do
           begin
            text_box2[i] := pcitra[i];
           end;
           skala:=skala3[1];
          end;
     end;
     sub_1:=1; M0_1:=1;M0_2:=1;
     m1 :=1;
     sub_2:=1;
     sub_3:=1;
end;

begin
     map:=1;
     reset;
     { The first condition of variable }
     { ------------------------------ }
     repeat
           c:=readkey;
           if c=#0 then
           begin
                c:=readkey;
                case c of
                     Up : begin
                            if (m1=2) or (m1=3) then
```

```
                    begin
                         if m1=2 then
                         begin
                            aktif_menu0(false,1,(sub_1-
                            M0_1)+1,
□
                                                text_box1[sub_1]);
                            if sub_1 > 1 then dec(sub_1);
                            if sub_1 < M0_1 then
                            begin
                              dec(M0_1);
                              Showsub(7,2,M0_1,
                              170,310,182,text_box1);
                              scroll_bar(308,184,234,
          round((3/7)*100),round(((M0_1-1)/7)*100));
                            end;
                        aktif_menu0(true,1,(sub_1-M0_1)+1,
                                                text_box1[sub_1]);
                            st:=text_box1[sub_1];
                            if map<>3 then
                               delete(st,ord(st[0])-1,2);
                            show_scala(st);
                            val(st,skala,err);
                         end
                         else
                         begin
                        aktif_menu0(false,2,(sub_2-M0_2)+1,
                                                text_box2[sub_2]);
                            if sub_2 > 1 then dec(sub_2);
                            if sub_2 < M0_2 then
                            begin
                              dec(M0_2);
                              showsub(maxi,4,M0_2,
                                 315,450,150,text_box2);
                              scroll_bar(448,152,234,
      round((5/maxi)*100),round(((M0_2-1)/maxi)*100));
                            end;
                         aktif_menu0(true,2,(sub_2-M0_2)+1,
                                                text_box2[sub_2]);
                         end; {3}
                    end; {begin-m1}
               end;{Up}

            Down: begin
                    if (m1=2) or (m1=3) then
                    begin
                       if m1=2 then
                       begin
                       aktif_menu0(false,1,(sub_1-M0_1)+1,
                                   text_box1[sub_1]);
                        if sub_1 < 7 then inc(sub_1);
                        if sub_1 > (M0_1+2) then
                        begin
                         inc(M0_1);
                         Showsub(7,2,M0_1,
                                 170,310,182,text_box1);
```

```
                scroll_bar(308,184,234,
      round((3/7)*100),round(((M0_1-1)/7)*100));
               end;
               aktif_menu0(true,1,(sub_1-M0_1)+1,
                           text_box1[sub_1]);
               st:=text_box1[sub_1];
               if map<>3 then
                  delete(st,ord(st[0])-1,2);
               show_scala(st);
               val(st,skala,err);
              end
              else
              begin
              aktif_menu0(false,2,(sub_2-M0_2)+1,
                           text_box2[sub_2]);
               if sub_2 < maxi then inc(sub_2);
               if sub_2 > (M0_2+4) then
               begin
                 inc(M0_2);
                 showsub(maxi,4,M0_2,
                         315,450,150,text_box2);
                 scroll_bar(448,152,234,
round((5/maxi)*100),round(((M0_2-1)/maxi)*100));
               end;
               aktif_menu0(true,2,(sub_2-M0_2)+1,
                           text_box2[sub_2]);
               end;
              end;
            end;{Down}

  Right : begin
            if (m1=1) then
            begin
              aktif_menu1(false,map,m1);
              if (map=3) then map:=1
                         else inc(map);
              aktif_map(map);
              reset;
              aktif_menu1(true,map,m1);
            end;
          end; {right}

  Left  : begin
            if (m1=1) then
            begin
              aktif_menu1(false,map,m1);
              if (map=1) then map:=3
                         else dec(map);
              aktif_map(map);
              reset;
              aktif_menu1(true,map,m1);
            end;
          end;

  F2 : begin
         if m1=2 then
```

```
                              begin
                                   aktif_menu1(false,map,m1);
                                   case map of
                                   1 : skala := valuekia(170,210,150,
                                               false,97);
                                   2 : skala := valuekia(170,210,150,
                                               true,1274);
                                   3 : skala := valuekia(170,210,150,
                                               false,200);
                                   end;
                                   inc(m1);
                                   aktif_menu1(true,map,m1);
                              end;
                            end;

                       F1 : begin
                                 Help;
                            end;
                  end; {case c}
             end{begin-c}
             else
             begin
                  case c of
                  enter : if m1=4 then
                          begin
                               sub:=0;
                               if map=2 then csub_3:=3
                                    else csub_3:=2;
                               sub := sub_3;
                               if sub_3=csub_3 then sub_3:=1
                                    else inc(sub_3);
                               demo_tombol(map,4,text_box3[sub],
                                    text_box3[sub_3]);
                          end;
                  tab   : begin
                               aktif_menu1(false,map,m1);
                               if m1=5 then m1:=1
                                    else inc(m1);
                               aktif_menu1(true,map,m1);
                          end;
                  end;
             end;
        until ((m1=5)and(c=enter)) or (c=Esc);
        demo_tombol(map,m1,'kacang','garing');
end;

procedure menu01;
label menu;
begin
        menu:
        daftarkanpic;
        drv:=detect;
        initgraph(drv,md,'c:\progra~2\units');
        justify;
        setfillstyle(1,3);
```

```
      bar(0,0,639,479);
      {First state}
      latar;
      for drv:=3 downto 1 do
          aktif_map(drv);
      aktif_menu1(true,1,1);
      {*}
      DeteksiTombol(map,skala,sub_2,sub_3);
      closegraph;
      if c=Enter then
      begin
        Tampil(map,skala,sub_2,sub_3);
        repeat
          c1:=readkey;
        until (c1=Enter) or (c1=Esc);
        textmode(80);
        if c1=Enter then goto menu
                    else c:=Esc;
      end
      else
      begin
        textcolor(yellow);
        writeln('Thank''s for trying !');
        write('Programmed by ');
        textcolor(yellow+blink);
        writeln('Saefudin');
      end;
      drv:=detect;
      initgraph(drv,md,'c:\progra~2\units');
end;

end.
```

```
unit menu2;
interface

procedure menu02;

implementation

uses crt,graph,pen2,life,citra;

var
   drv,md : integer;
   form1  : atrtombol;
   button : array[1..3] of atrtombol;
   x,y    : word;
   m1,m3 : byte;
   m2 : real;
   c  : char;

const pesan : array[1..2] of string
                        = ('Perbesaran','Pengecilan');

procedure tampilan;
begin
      setfillstyle(1,7);
      atur_tombol(x,y,170,245,1,false,form1);
   atur_tombol(15+x,80+y,138,25,1,false,button[1]);
   atur_tombol(15+x,160+y,138,25,1,false,button[2]);
   atur_tombol(15+x,210+y,138,25,1,false,button[3]);
   buat_tombol(form1,false);
   buat_tombol(button[1],false);
   buat_tombol(button[2],false);
   buat_tombol(button[3],false);
   tulis_tombol(button[2],pcitra[1]);
   tulis_tombol(button[3],'OK');
   garis(5+x,198+y,x+165,198+y);
   setcolor(yellow);
   tulis_tombol(button[1],pesan[1]);
   {simbol}
   settextstyle(10,0,5);
   setusercharsize(2,1,1,1);
   catatv1(91+x,25+y,1,chr(232),3);
   kotak3dr(x+15,y+10,140,60);
   {empty block}
   emp_bar(15+x,120+y,153+x,142+y,white);
end;

procedure Input_keyboard(var m1:byte;var m2:real;var m3:byte);
const
     tab   = #9;
   enter = #13;
   esc   = #27;
var
      lm1,lm3 : byte;      warna: word;
    lmenu,menu    : byte;
begin
     lm1:=1;lm3:=1;
```

```
menu:=1;lmenu:= 1;
repeat
   c:=readkey;
   case c of
         tab    : begin
                       if menu=4 then menu:=1
                      else inc(menu);
                  case menu of
                  1 : begin
                       warna:= getcolor;
                       setcolor(yellow);
                       tulis_tombol(button[1],pesan[m1]);
                       setcolor(warna);
                       tulis_tombol(button[3],'OK');
                      end;
                  2 : begin
                       setcolor(0);
                       tulis_tombol(button[1],pesan[m1]);
           if m1=1 then m2:=valuekia(15+x,153+x,120+y,true,1250)
                       else
                       m2:=valuekia(15+x,153+x,120+y,false,100);
                      end;
                  3 : begin
                       warna:= getcolor;
                       setcolor(yellow);
                       tulis_tombol(button[2],pcitra[m3]);
                       setcolor(warna);
                       tulis_tombol(button[1],pesan[m1]);
                      end;
                  4 : begin
                       warna:= getcolor;
                       setcolor(yellow);
                       tulis_tombol(button[3],'OK');
                       setcolor(warna);
                       tulis_tombol(button[2],pcitra[m3]);
                      end;
                  end;
                  lmenu:= menu;
                   end;
                Enter : begin
                       if menu=1 then
                  begin
                       if m1=2 then dec(m1)
                       else inc(m1);
             simulasi_ditekan2(button[1],pesan[lm1],pesan[m1]);
                       lm1:=m1;
                  end;
                  if menu=3 then
                  begin
                       if m3=maxi then m3:=1
                       else inc(m3);
            simulasi_ditekan2(button[2],pcitra[lm3],pcitra[m3]);
                       lm3:=m3;
                  end;
                  setcolor(0);
               end;
```

```
      end;
   until (c=Esc) or ((c=Enter) and (menu=4));
end;

Procedure menu02;
label menu;
var ff : fillsettingstype;
begin
     drv := detect;
     menu:
     initgraph(drv,md,'c:\progra~2\units');
     daftarkanpic;
     getfillsettings(FF);
     setfillstyle(1,3);
     bar(0,0,639,479);
     with ff do
     setfillstyle(pattern,color);
     x:=230;y:=120;
     tampilan;
     m1:=1;m3:=1;
     Input_keyboard(m1,m2,m3);
     closegraph;
     if c=#13then show_compare(c,m1,m2,m3);
     if c=#13 then goto menu;
     drv := detect;
     initgraph(drv,md,'c:\progra~2\units');
end;

end.
```

```
unit Life;

interface

uses dos,crt,graph,pen2;

type text = array[1..30] of string;

Const
     Skala1 : array[1..7] of word
            = (05,10,15,25,50,75,90);
     Skala2 : array[1..7] of word
            = (125,150,200,250,500,1000,1250);
     Skala3 : array[1..7] of word
            = (01,02,03,05,10,20,30);

Var
   pcitra    : text;
   maxi      : word;

Procedure aktif_menu1(toggle:boolean;map,m1:word);
Procedure aktif_menu0(toggle:boolean;box,show:byte;text:string);
Procedure demo_tombol(map,m1:word;txt1,txt2:string);
Procedure Justify;
procedure daftarkanpic;

implementation

type
    pilihan1 = record
                x,y    : integer;
                tx,ty  : word;
    end;

    pilihan2 = record
                x,y    : integer;
                pesan  : string;
    end;
    menumap  = array[1..3] of pilihan2;
    menuM1   = array[1..3,1..5] of Pilihan2;
    MenuSub  = array[1..3,1..5,1..5] of Pilihan2;
    Menu0    = array[1..3,1..5] of Pilihan1;

const
     MPengecilan : array[1..5] of string
                 = ('Pengecilan','Diperkecil menjadi :','Citra
:','Metode :','O K');
     MPembesaran : array[1..5] of string
                 = ('Pembesaran','Diperbesar menjadi :','Citra
:','Metode :','O K');
     MVariasiK : array[1..5] of string
                   = ('Variasi Kontras','Faktor :','Citra
:','Metode :','O K');

     Posx_m1 : array[1..3,1..5] of integer
             = ((170,175,320,170,310),
```

```
{ Unit dipergunakan dalam penggambaran
  tool-tool dalam GUI
  ===================================
}
unit pen2;
interface
uses crt,graph;

type AtrTombol=record
     xx,yy,txx,tyy : word;
                  tebal      : word;
                  status       : boolean;
                               end;
type Atrpilihan= record
                  x,y,dx,dy : word;
                 end;

procedure kotak3d(x1,y1,x2,y2 : integer);
procedure kotak3dr(x,y,tx,ty : integer);
procedure kotak3dc(xc,yc,tx,ty : integer);
procedure catat0(x,y:integer;z:byte;text:string);
procedure catat1(x,y:integer;z:byte;text:string);
procedure catat2(x,y:integer;z:byte;text:string);
procedure catatv0(x,y:integer;z:byte;text:string;mode:byte);
procedure catatv1(x,y:integer;z:byte;text:string;mode:byte);
procedure catatv2(x,y:integer;z:byte;text:string;mode:byte);
procedure Teks0(x,y:integer;z:byte;text:string;mode:byte);
procedure Teks1(x,y:integer;z:byte;text:string;mode:byte);
procedure Teks2(x,y:integer;z:byte;text:string;mode:byte);
procedure header0(x,y:integer;z:byte;text:string;col:word);
procedure header1(x,y:integer;z:byte;text:string;col:word);
procedure header2(x,y:integer;z:byte;text:string;col:word);
procedure tombolkia(tertekan:boolean;tebal:byte;
                    x1,y1,x2,y2:integer);
procedure rtombolkia(tertekan:boolean;tebal:byte;
                    x,y,tx,ty:integer);
procedure tombolkib(tertekan:boolean;tebal:byte;
                    x1,y1,x2,y2:integer);
procedure rtombolkib(tertekan:boolean;tebal:byte;
                    x,y,tx,ty:integer);
procedure tombolcc(tertekan:boolean;tebal:byte;
                    x,y,tx,ty:integer);
procedure map(tertekan:boolean;tebal:byte;
                    x1,y1,x2,y2:integer);
procedure emp_bar(x1,y1,x2,y2 : integer;col:word);
procedure remp_bar(x,y,tx,ty : integer;col:word);
procedure choice_barkia(x1,y1,x2,y2:integer;col:word);
procedure rchoice_barkia(x,y,tx,ty:integer;col:word);
procedure choice_barkib(x1,y1,x2,y2:integer;col:word);
procedure rchoice_barkib(x,y,tx,ty:integer;col:word);
procedure scroll_bar(x,y1,y2:integer;percent1,percent2:word);
procedure sc_bar(x,y1,y2:integer;percent1,percent2:word);
Procedure garis(x1,y1,x2,y2:integer);
function valuekib(x1,x2,y:integer;lbh_bsr:boolean;nilai:word) :
real;
```

```
function valuekia(x1,x2,y:integer;Lbh_bsr:boolean;nilai:word) :
real;
{prosedur baru dari buku}
procedure Atur_Tombol(x,y,tx,ty:word;tbl:byte;
                                          stat:boolean;var
box:atrtombol);
procedure buat_tombol(box:atrtombol;stat:boolean);
procedure tulis_tombol(box:atrtombol;pesan:string);
procedure Simulasi_Ditekan(box:atrtombol;pesan:string);
procedure Simulasi_Ditekan2(box:atrtombol;pesan1,pesan2:string);

procedure atur_pilihan(xa,ya,dxa,dya:word;var pilihan:Atrpilihan);
procedure buat_pilihan(pilihan:AtrPilihan);
procedure lighting_pilihan(pilihan:atrPilihan;col:boolean);
procedure tulis_pilihan(pilihan:AtrPilihan;text:string;col:byte);
procedure buat_short(pilihan:AtrPilihan);

procedure help(x,y:word;var c:char);
procedure intisari(x,y:word;var c:char);
procedure about(x,y:word;var c:char);

implementation

procedure kotak3d(x1,y1,x2,y2 : integer);
begin
     setcolor(8);
     rectangle(x1,y1,x2,y2);
     setcolor(15);
     line(x1,y1+1,x2,y1+1);
     line(x1,y2+1,x2,y2+1);
     line(x1+1,y1+1,x1+1,y2-1);
     line(x2+1,y1,x2+1,y2);
end;

procedure kotak3dr(x,y,tx,ty : integer);
begin
     setcolor(8);
     rectangle(x,y,x+tx,y+ty);
     setcolor(15);
     line(x,y+1,x+tx,y+1);
     line(x,y+ty+1,x+tx,y+ty+1);
     line(x+1,y+1,x+1,y+ty-1);
     line(x+tx+1,y,x+tx+1,y+ty);
end;

procedure kotak3dc(xc,yc,tx,ty : integer);
var x1,y1,x2,y2 : integer;
begin
     setcolor(8);
     x1:= xc-tx div 2;  y1:= yc-ty div 2;
     x2:= xc+tx div 2;  y2:= yc+ty div 2;
     rectangle(x1,y1,x2,y2);
     setcolor(15);
     line(x1,y1+1,x2,y1+1);
     line(x1,y2+1,x2,y2+1);
     line(x1+1,y1+1,x1+1,y2-1);
```

```
Unit Graf0;
interface
uses crt,vgacad;
Var
   awal, input, output   : pointer;

Procedure Kursor(var dx,dy:word;image:string);
Procedure Install;
Procedure go_graph;

implementation

Procedure Install;
begin
  OpenVga256;
  mark(awal);
  Getmem(input,64000);
  getmem(output,64000);
  Install_palette('c:\progra~2\units\b&w.pal');
end;

Procedure go_graph;
begin
  freemem(input,64000);
  freemem(output,64000);
  release(awal);
end;

Procedure Kursor(var dx,dy:word;image:string);
const
     up    = 1;
     down  = 2;
     left  = 3;
     right= 4;
     PgDn  = 5;
     Edd   = 6;
     PgUp  = 7;
     Home  = 8;
var
   udani,panah,gbr    : pointer;
   aa,bb,x1,y1,x2,y2  : integer;
   dir,stand          : byte;
   ch                 : char;
   luar               : boolean;
   label              kembali;
begin
     install;
     kembali:
     stand:=0;
     x1:=0; y1:=0;
     aa:=10; bb:=10;
     luar := false;
     Load_Picture(image,screen^);
     moveto(screen^,input^);
     readln;
     Load_Picture('c:\progra~2\tools\arrow.pic',output^);
```

```
mark(udani);
getmem(panah,imagesize(0,0,11,10));
getimage(output^,0,0,11,10,panah^);
fillword(screen^,32000,0);
{===}
repeat
      moveto(input^,output^);
      case dir of
           up    : if (bb>0) and (bb>y1) then dec(bb,1);
           down  : if bb<199 then inc(bb,1);
           left  : if (aa>0) and (aa>x1) then dec(aa,1);
           right : if aa<319 then inc(aa,1);
           PgUp  : if (bb>0) and (bb>(y1+11)) then
                   dec(bb,10);
           PgDn  : if bb<188 then inc(bb,10);
           Home  : if (aa>0) and (aa>(x1+10)) then
                   dec(aa,10);
           Edd   : if aa<308 then inc(aa,10);
      end;
      if stand=1 then rectangle(output^,x1,y1,aa,bb,55);
      putimage(output^,aa,bb,panah^);
      moveto(output^,screen^);

      ch := readkey;
      case ch of
           #80  : dir := down;
           #72  : dir := up;
           #75  : dir := left;
           #77  : dir := right;
           #81  : dir := PgDn;
           #79  : dir := Edd;
           #73  : dir := PgUp;
           #71  : dir := Home;
           #27  : luar := true;
           ' '  : dir := 0;
           #13  : begin
                       if stand = 1 then
                       begin
                            x2 := aa;
                            y2 := bb;
                            getmem(gbr,imagesize(x1+1,y1+1,
                               x2-1,y2-1));
                            getimage(screen^,x1+1,y1+1,
                               x2-1,y2-1,gbr^);
                       end
                       else
                       begin
                            x1 :=aa;
                            y1 :=bb;
                       end;
                       inc(stand,1)
                  end;
      end;
until (luar) or (stand=2);
if luar then goto kembali;
{====}
```

```
        fillword(screen^,32000,0);
        putimage(screen^,0,0,gbr^);
        freemem(panah,imagesize(50,50,60,60));
        freemem(gbr,imagesize(x1+1,y1+1,x2-1,y2-1));
        release(udani);
        moveto(screen^,input^);
        dx:=x2-x1;dy:=y2-y1;
        readln;
end;

end.
```

```
Unit Graf00;
interface

Var
   start,page01,page02,page03 : pointer;

Procedure Install_pages;
Procedure go_graph2;
Procedure garis(x,y:byte;yy,dyy:real);

implementation

uses crt,vgacad;

procedure install_pages;
begin
   mark(start);
      getmem(page01,64000);
   getmem(page02,64000);
   getmem(page03,64000);
end;

procedure go_graph2;
begin
      freemem(page01,64000);
   freemem(page02,64000);
   freemem(page03,64000);
   release(start);
end;

procedure garis(x,y:byte;yy,dyy:real);
var
   fakt:byte;
begin
     rectangle(screen^,x,y,x+150,y+4,150);
     fakt:=round(yy*150/dyy);
     line(screen^,x+1,y+1,x+fakt,y+1,200);
     line(screen^,x+1,y+2,x+fakt,y+2,200);
     line(screen^,x+1,y+3,x+fakt,y+3,200);
end;

end.
```

```
unit citra;
interface

uses crt,life,vgacad,graf0,graf00,zoom,graf2;
Procedure Tampil(map:byte;skala:real;sub_2,sub_3:word);
Procedure Show_Compare(var c:char;m1:byte;m2:real;m3:byte);

implementation

Procedure Tampil(map:byte;skala:real;sub_2,sub_3:word);
var gambar : string;
begin
     case map of
          2    : begin
                   skala:=(skala/100);
                   gambar := 'c:\progra~2\citra\'+
                             pcitra[sub_2]+'.pic';
                   case sub_3  of
                          1    : Replikasi(skala,gambar);
                          2    : Bilinier(skala,gambar);
                          3    : Bikubik(skala,gambar);
                   end;
                  end;
          1    : begin
                   skala:=(skala/100);
                   gambar := 'c:\progra~2\citra\'+
                             pcitra[sub_2]+'.pic';
                   case sub_3  of
                          1    : Replikasi(skala,gambar);
                          2    : scala1(skala,gambar);
                   end;
                  end;
          3    : begin
                   gambar := 'c:\progra~2\citra\'+
                             pcitra[sub_2]+'.pic';
                   case sub_3 of
                          1 :  ladd1(round(skala),gambar);
                          2 :  ladd2(round(skala),gambar);
                   end;
                  end;
     end;
end;


Procedure Show_Compare(var c:char;m1:byte;m2:real;m3:byte);
var gambar : string;
begin
   case m1 of
          1    : begin
                   m2:=(m2/100);
                   gambar := 'c:\progra~2\citra\'+
                             pcitra[m3]+'.pic';
                   compare1(m2,gambar);
                  end;
          2    : begin
                   m2:=(m2/100);
```

```
                    gambar := 'c:\progra~2\citra\'+
                              pcitra[m3]+'.pic';
                    compare2(m2,gambar);
                  end;
    end;
    repeat
       c:=readkey;
       case c of
            '1'  : moveto(page01^,screen^);
            '2'  : moveto(page02^,screen^);
            '3'  : if m1=1 then moveto(page03^,screen^);
       end;
    until (c=#27) or (c=#13);
    readln;
    go_graph2;
end;

end.
```

```
Unit zoom;
interface

Procedure Replikasi(scala:real;citra:string);
Procedure Bilinier(scala:real;citra:string);
Procedure Bikubik(scala:real;citra:string);
Procedure Compare1(scala:real;citra:string);
Procedure Compare2(scala:real;citra:string);
procedure scala1(scala: real;citra: string);

implementation

uses crt,vgacad,gerbang1,tipe2,graf0,graf00;

Procedure zoom1(dx,dy:word;scala:real;citra:string);
var
    Ym,Xm,i,j        : integer;
    F                : integer;
    Xb,Yb,Xbn,Ybn    : real;

Begin
  fillword(screen^,32000,0);
  {*}
  if dy*scala < 200 then Ym := dy
          else Ym := round(200/scala)-1;
  if dx*scala < 320 then Xm := dx
          else Xm := round(320/scala)-1;

  for y:=0 to Ym do
  begin
       for x:=0 to Xm do
       begin
            F := getpixel(input^,x,y);

            Xb:= scala*x;
            Yb:= scala*y;
            Xbn:= scala*(x+1);
            Ybn:= scala*(y+1);

            for j:= round(Yb) to round(Ybn)-1 do
                 for i:= round(Xb) to round(Xbn)-1 do
                      Putpixel(screen^,i,j,F);
       end;
  end;
end;

Procedure zoom2(dx,dy:word;scala:real;citra:string);
var
    F1,F2,F3,F4      : integer;
    Ym,Xm,i,j        : integer;
    Xb,Yb,Xbn,Ybn    : real;
    SA,NA,SZ         : real;
    ii,jj            : byte;
Begin
  fillword(screen^,32000,0);
  {*}
```

```
  if scala > 1 then
  begin
       if dy*scala < 200 then Ym := dy
          else Ym := round(200/scala)-1;
       if dx*scala < 320 then Xm := dx
          else Xm := round(320/scala)-1;
  end;
  for y:=0 to Ym do
  begin
       for x:=0 to Xm do
       begin
            F1:= getpixel(input^,x,y);
            F2:= getpixel(input^,x+1,y);
            F3:= getpixel(input^,x,y+1);
            F4:= getpixel(input^,x+1,y+1);

            Xb:= scala*x;
            Yb:= scala*y;
            Xbn:= scala*(x+1);
            Ybn:= scala*(y+1);

            jj:=1;
            for j:= round(Yb) to round(Ybn)-1 do
            begin
                 SA:= ((F3-F1)/scala)*(jj-1)+F1;
                 SZ:= ((F4-F2)/scala)*(jj-1)+F2;

                 ii:=1;
                 for i:= round(Xb) to round(Xbn)-1 do
                 begin
                      NA:= ((SZ-SA)/scala)*(ii-1)+SA;
                      Putpixel(screen^,i,j,round(NA));
                      inc(ii);
                 end;
                 inc(jj);
            end;
       end;
  end;
end;

Procedure Try(dx,dy:integer;scala:real;
n:integer;spline_y:boolean;
              Var page0,page1 : pointer);
Var
   x1,y1,x2,y2,max_b,max_k :integer;
   i,j,k,m       : integer;
   f             : datas;
   xx,zz         : data;
   kolom,baris   : integer;
   a,c           : matriks;
   b,d           : real;
   cacah         : integer;
   more,less     : real;
```

```
function terbesar(a,b:real) : real;
begin
      if a > b then terbesar:=a
                  else terbesar:=b;
end;

function terkecil(a,b:real) : real;
begin
      if a < b then terkecil:=a
                  else terkecil:=b;
end;

procedure Compare(less:real;var Yu:real;more:real);
begin
      if Yu > more then Yu:=more;
      if Yu < less then Yu:=less;
end;

Begin
     m:=round(scala*n);
     if spline_y then
     begin
          max_b:=dx;
          if (dy*scala)<= 199 then max_k:=trunc(dy*scala/m)
             else max_k:=trunc(199/m)
     end
     else
     begin
          if (dy*scala)<= 199 then max_b := trunc(dy*scala)
             else max_b := 199;
          if (dx*scala)<= 319 then max_k := trunc(dx*scala/m)
             else max_k := trunc(319/m)
     end;

     for Kolom := 1 to max_k do
     begin
          for baris:=0 to max_b do
          begin
               for i:=0 to n do
               begin
                    xx[i]:=round(i*scala);
                    if spline_y=true then
                    begin
                         x1:=baris;
                         y1:=(kolom-1)*n+i;
                         x2:=baris;
                         y2:=(kolom-1)*round(scala*n)+xx[i]
                    end
                    else
                    begin
                         y1:=baris;
                         x1:=(kolom-1)*n+i;
                         y2:=baris;
                         x2:=(kolom-1)*round(scala*n)+xx[i]
                    end;
                    f[round(i*scala)] :=getpixel(page0^,x1,y1);
```

```
   putpixel(page1^,x2,y2,round(f[round(i*scala)]));
end;
more:=terbesar(f[0],f[round(scala)]);
for cacah:=2 to n do
   more:=terbesar(more,f[round(cacah*scala)]);
less:=terkecil(f[0],f[round(scala)]);
for cacah:=2 to n do
less:=terkecil(less,f[round(cacah*scala)]);
for i:=0 to n do
   for j:=0 to n+1 do
       A[i,j]:=0;

{ Matriks Koefisien }
A[1,2]:=xx[2]-xx[1];
A[1,1]:=xx[1]-xx[0];
A[1,1]:=2*(A[1,1]+A[1,2]);

A[n-1,n-2]:=xx[n-1]-xx[n-2];
A[n-1,n-1]:=xx[n]-xx[n-1];
A[n-1,n-1]:=2*(A[n-1,n-2]+A[n-1,n-1]);

for i:=2 to n-2 do
begin
     A[i,i-1]:=xx[i]-xx[i-1];
     A[i,i+1]:=xx[i+1]-xx[i];
     A[i,i]:=2*(A[i,i-1]+A[i,i+1]);
end;

for i:=1 to n-1 do
begin
     A[i,n]:=(f[round((i+1)*scala)]-
      f[round(i*scala)])/(xx[i+1]-xx[i]);
     A[i,n]:=A[i,n]-(f[round(i*scala)]-f[round((i-
      1)*scala)])/(xx[i]-xx[i-1]);
     A[i,n]:=6*A[i,n];
end;

{Gauss Jordan}
for i:=1 to n-1 do
begin
     if A[i,i] = 0 then
     begin
          for j:=i+1 to n-1 do
          begin
               if A[j,i] <> 0 then
               begin
                    for k:=i to n do
                    begin
                         b:=A[i,k];
                         A[i,k]:=A[j,k];
                         A[j,k]:=b;
                    end;
               end;
          end;
     end;
     b:=A[i,i];
```

```
                    for j:=i to n do
                        A[i,j] := A[i,j]/b;
                    for j:=1 to n-1 do
                    begin
                        if j<>i then
                        begin
                        for k:=i to n do
                            c[j,k]:=A[j,k]-A[j,i]*A[i,k];
                        for k:=i to n do
                            A[j,k]:=c[j,k];
                        end;
                    end;
                end;

                {End of Gauss Jordan}

                for i:=0 to n-1 do
                begin
                    d:=xx[i+1]-xx[i];
                    c[1,i]:=(A[i+1,n]-A[i,n])/(6*d);
                    c[2,i]:=A[i,n]/2;
                    c[3,i]:=(f[round((i+1)*scala)]-
                            f[round(i*scala)])/d;
                    c[3,i]:=c[3,i]-((2*d*A[i,n]+d*A[i+1,n])/6);
                    c[4,i]:=f[round(i*scala)];
                end;

                FOR i:=0 to m do
                begin
                    for j:=0 to n do
                    if (i >= xx[j]) and (i <= xx[j+1]) then
                    begin
                        d:=i-xx[j];
                        f[i]:=C[1,j]*d*sqr(d)+C[2,j]*sqr(d)+
                              C[3,j]*d+C[4,j];
                        if spline_y=true then
                        begin
                            x2:=baris;
                            y2:=(kolom-1)*m+i
                        end
                        else
                        begin
                            y2:=baris;
                            x2:=(kolom-1)*m+i
                        end;
                        compare(less,f[i],more);
                        putpixel(page1^,x2,y2,round(f[i]));
                    end;
                end;
            end;
        if spline_y then
           garis(80,180,kolom,max_k);
        end;
end;
```

```
Procedure zoom3(dx,dy:word;scala:real;citra:string);
const n=2;
var f: integer;
begin
     fillword(screen^,32000,0);
     fillword(output^,32000,0);
     Load_Picture('C:\PROGRA~2\tools\2.pic',screen^);
     Try(dx,dy,scala,n,true,input,output);
     fillword(input^,32000,0);
     moveto(output^,input^);
     fillword(screen^,32000,0);
     Try(dx,dy,scala,n,false,input,screen);
end;

Procedure Compare1(scala:real;citra:string);
var
    dx,dy:word;
    Ym,Xm,i,j        : integer;
    F,n              : integer;
    Xb,Yb,Xbn,Ybn    : real;
Begin
  install_pages;
  dx:=1;dy:=1;
  kursor(dx,dy,citra);
  {}
  zoom1(dx,dy,scala,citra);
  readln;
  moveto(screen^,page01^);

  zoom2(dx,dy,scala,citra);
  readln;
  moveto(screen^,page02^);

  zoom3(dx,dy,scala,citra);
  moveto(screen^,page03^);
  { menghilangkan input dan output page saja
    ---------------------------------------- }
  go_graph;
end;

{   metode pengecilan reduksi rata-rata   }
{   ------------------------------------   }

procedure reduksi1(xxx : boolean; Var page0,page1 : pointer;
                   dx,y : integer;ak_m:real);
var
   realpos, realpointer : real;
   xpos, xpointer2      : integer;
   xpointer, xscreen    : integer;
   f                    : integer;
begin
     realpointer := 0; xscreen :=0;
     while realpointer < (dx+1) do
     begin
          realpos := realpointer;
          realpointer := realpointer + 1/ak_m;
```

```
            if (realpos = 0) then xpos := 0
               else xpos := trunc(realpos) + 1;
            xpointer2 := round(realpointer);

            f:=0;
            for xpointer := xpos to xpointer2 do
                if xxx then f := getpixel(page0^, xpointer, y) + f
                   else f := getpixel(page0^, y,xpointer) + f;

            if frac(f/(xpointer2 - xpos + 1)) > 0.5 then
               f := round(f/(xpointer2 - xpos + 1)) else
               f := trunc(f/(xpointer2 - xpos + 1));

            if xxx then putpixel(page1^,xscreen,y,f)
               else putpixel(page1^,y,xscreen,f);
            inc(xscreen);
      end;
end;

procedure Compare2(scala:real;citra:string);
var
    dx,dy:word;
    Ym,Xm,i,j        : integer;
    F                : integer;
    Xb,Yb,Xbn,Ybn    : real;
Begin
  install_pages;
  dx:=1;dy:=1;
  kursor(dx,dy,citra);
  {}
  zoom1(dx,dy,scala,citra);
  readln;
  moveto(screen^,page01^);

  fillword(screen^,32000,0);
  Load_Picture('C:\PROGRA~2\tools\2.pic',screen^);
  fillword(output^,32000,0);
  for y:=1 to dy do
  begin
      reduksi1(true,input,output,dx,y,scala);
        garis(80,180,y,dy);
  end;
  moveto(output^,input^);
  fillword(screen^,32000,0);
  for x:=1 to round(dx*scala) do
      reduksi1(false,input,screen,dy,x,scala);
  moveto(screen^,page02^);
  go_graph;
end;

Procedure Replikasi(scala:real;citra:string);
var dx,dy : word;
begin
  dx:=1;dy:=1;
  kursor(dx,dy,citra);
```

```
  zoom1(dx,dy,scala,citra);
  readln;
  go_graph;
end;

Procedure Bilinier(scala:real;citra:string);
var dx,dy : word;
begin
  dx:=1;dy:=1;
  kursor(dx,dy,citra);
  zoom2(dx,dy,scala,citra);
  readln;
  go_graph;
end;

Procedure Bikubik(scala:real;citra:string);
var dx,dy : word;
begin
  dx:=1;dy:=1;
  kursor(dx,dy,citra);
  zoom3(dx,dy,scala,citra);
  readln;
  go_graph;
end;

procedure scala1(scala: real;citra: string);
var dx, dy : word;
begin
  dx:=1;dy:=1;
  kursor(dx,dy,citra);
  fillword(screen^,32000,0);
  Load_Picture('C:\PROGRA~2\tools\2.pic',screen^);
  fillword(output^,32000,0);
  for y:=1 to dy do
  begin
      reduksi1(true,input,output,dx,y,scala);
      garis(80,180,y,dy);
  end;
  moveto(output^,input^);
  fillword(screen^,32000,0);
  for x:=1 to round(dx*scala) do
      reduksi1(false,input,screen,dy,x,scala);
  go_graph;
end;
end.
```