

*Lampiran 1*

```

{*****}
NAMA PROGRAM : FRACIENA.PAS          LAST UPDATED : 1 MEI 2000
KETERANGAN   : PROGRAM UNTUK VISUALISASI TRANSFORMASI AFIN DAN
                SISTEM ITERASI FUNGSI DALAM GEOMETRI FRAKTAL
PROGRAMMER   : DIENA AMALIA - J101951190
{*****}

program fraciena;

uses crt, graph, fractal;

var k,p1,p2,p3,p4 : integer; p5 : longint; sk,p,q,r,s : real;

begin
  opengraph;
  buka;
  repeat
    latar;
    menuutama(p1);
    case p1 of
      1 : begin
          matriks(p,q,r,s); polos;
          layar(4);
          transformasi(p,q,r,s); readkey;
        end;
      2 : begin
          repeat
            latar; algoritma(p2);
          case p2 of
            1 : begin
                repeat
                  latar; bentukfraktal(p3);
                  if p3 in [1..6] then begin
                      macambenih(p4);
                      if p4 in [1..8] then begin
                          polos;initmatbsr;layar(12);trkx:=5;trky:=5;
                          case p3 of
                            1 : sierpin(1.5);
                            2 : carp(1.5);
                            3 : penta(1.5);
                            4 : heigh(1.5);
                            5 : vonkoch(1.5);
                            6 : lev(1.5);

```

```

end;
case p4 of
  1 : gasket;
  2 : kotak;
  3 : segilima;
  4 : naga;
  5 : garis;
  6 : kurva;
  7 : segitiga;
  8 : segiempat;
end;
for k:=2 to 12 do determ(k, trkx, trky);
readkey; freematbsr;
end;
end;
until p3 = 0;
end;
2 : begin
  repeat
    latar; bentukfraktal(p3);
    if p3 in [1..6] then begin
      banyakitr(p5);
      if p5 < 0 then begin
        polos; layar(1);
        case p3 of
          1 : sierpin1(p5);
          2 : carpl(p5);
          3 : pental(p5);
          4 : heigh1(p5);
          5 : vonkoch1(p5);
          6 : lev1(p5);
        end;
        readkey;
      end;
    end;
  until p3 = 0;
end;
3 : begin
  repeat
    latar; bentukfraktal(p3);
    if p3 in [1..6] then begin
      banyaktkk(p4);
      if p4 in [1..3] then begin
        trky := 5; statisoff;
        case p4 of
          1 : begin

```

```

    latar; banyakitr(p5); trkx := 65;
    polos; statison; layar(1); k:= p5;
    if p3 in [2,4] then sk := 4.25
    else
        if p3 in [1,5] then sk := 5
        else sk := 3;
    end;
    2 : begin
        polos; layar(4); trkx:=15; k := 4;
        if p3 = 6 then sk := 1.2
        else
            if p3 in [3,4] then sk := 1.5
            else
                if p3 in [1,5] then sk := 2.5
                else sk := 2;
            end;
        end;
    3 : begin
        polos; layar(12); trkx:=5; k:= 12;
        if p3 in [3..4,6] then sk := 1
        else sk := 1.5;
        end;
    end;
    case p3 of
        1 : sierpin2(k, sk);
        2 : carp2(k, sk);
        3 : penta2(k, sk);
        4 : heigh2(k, sk);
        5 : vonkoch2(k, sk);
        6 : lev2(k, sk);
    end;
    readkey;
    end;
    until p3 = 0;
end;
    end;
    until p2 = 0;
end;
    end;
    until p1 = 3;
tutup;
closegraph;
end.

```

*Lampiran 2*

```

{*****}
NAMA PROGRAM : FRACTAL.PAS          LAST UPDATED : 1 MEI 2000
KETERANGAN   : UNIT PROGRAM FRACIENA.PAS
PROGRAMMER   : DIENA AMALIA - J101951190
{*****}

```

```
unit fractal;
```

```
interface
```

```
uses crt, graph;
```

```
type
```

```
  matbsr = array [0..150, 0..150] of byte;
```

```
  node   = record
```

```
    x, y : integer;
```

```
  end;
```

```
const
```

```
  m = 100;  n = 150;
```

```
  sierpinski : array [1..3, 1..6] of real =
```

```
    (( 0.500, 0.000, 0.000, 0.500, 0.000, 0.000 ),
```

```
     ( 0.500, 0.000, 0.000, 0.500, 0.500, 0.000 ),
```

```
     ( 0.500, 0.000, 0.000, 0.500, 0.250, 0.430 ));
```

```
  carpet     : array [1..8, 1..6] of real =
```

```
    (( 0.333, 0.000, 0.000, 0.333, 0.000, 0.000 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.000, 0.333 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.000, 0.667 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.333, 0.000 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.333, 0.667 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.667, 0.000 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.667, 0.333 ),
```

```
     ( 0.333, 0.000, 0.000, 0.333, 0.667, 0.667 ));
```

```
  pentagon   : array [1..5, 1..6] of real =
```

```
    (( 0.382, 0.000, 0.000, 0.382, 0.119, 0.000 ),
```

```
     ( 0.382, 0.000, 0.000, 0.382, 0.498, 0.000 ),
```

```
     ( 0.382, 0.000, 0.000, 0.382, 0.614, 0.363 ),
```

```
     ( 0.382, 0.000, 0.000, 0.382, 0.309, 0.588 ),
```

```
     ( 0.382, 0.000, 0.000, 0.382, 0.000, 0.363 ));
```

```
  heighway   : array [1..2, 1..6] of real =
```

```
    (( 0.500, -0.500, 0.500, 0.500, 0.310, 0.064 ),
```

```
     (-0.500, -0.500, 0.500, -0.500, 1.064, 0.437 ));
```

```
  koch       : array [1..4, 1..6] of real =
```

```
    (( 0.333, 0.000, 0.000, 0.333, 0.000, 0.000 ),
```

```

    ( 0.333, 0.000, 0.000, 0.333, 0.667, 0.000 ),
    ( 0.167,-0.289, 0.289, 0.167, 0.333, 0.000 ),
    ( 0.167, 0.289,-0.289, 0.167, 0.500, 0.289 ));
levy      : array [1..2, 1..6] of real =
  (( 0.500,-0.500, 0.500, 0.500, 0.310, 0.065 ),
   ( 0.500, 0.500,-0.500, 0.500, 0.190, 0.565 ));
nd2a : array [1..2] of node =
  ((x:0; y:0), (x:100; y:0));
nd2b : array [1..2] of node =
  ((x:0; y:40), (x:150; y:40));
nd3a : array [1..3] of node =
  ((x:0; y:0), (x:100; y:0), (x:50; y:86));
nd3b : array [1..3] of node =
  ((x:37; y:56), (x:113; y:56), (x:75; y:94));
nd4 : array [1..4] of node =
  ((x:0; y:0), (x:100; y:0),
   (x:100; y:100), (x:0; y:100));
nd5 : array [1..5] of node =
  ((x:0; y:88), (x:29; y:0),
   (x:121; y:0), (x:149; y:88), (x:75; y:143));

var
  s, t : ^matbsr; statisf : boolean; x, y, tx : real;
  cx, cy : array [1..12] of integer; nd : array [1..20] of node;
  a, b, c, d, e, f : array [0..15] of real;
  i, j, k, jtr, jnd, jktk, trkx, trky : integer;

procedure opengraph;
procedure initmatbsr;
procedure freematbsr;
procedure latar;
procedure polos;
procedure layar(p: integer);
procedure isi(x1,y1,x2,y2: integer; warna: byte);
procedure paku(x,y: integer);
procedure icon(x1,y1,x2,y2,c: integer; warnaicon: byte;
  nama:string; warnanama:byte; shadow:boolean; mor:boolean);
procedure isian(x1,y1,x2,y2: integer);
function input(temx,temy: integer; max: real; latar: byte): real;
procedure buka;
procedure tutup;
procedure menuutama(var pilih: integer);
procedure matriks(var p,q,r,s: real);
procedure macambenh(var pilih: integer);
procedure algoritma(var pilih: integer);
procedure bentukfraktal(var pilih: integer);

```

```

procedure banyakitr(var pilih: longint);
procedure banyaktk(var pilih: integer);
procedure transformasi(p,q,r,s: real);
procedure gasket;
procedure kotak;
procedure segilima;
procedure naga;
procedure garis;
procedure kurva;
procedure segitiga;
procedure segiempat;
procedure sierpin(kk: real);
procedure carp(kk: real);
procedure penta(kk: real);
procedure heigh(kk: real);
procedure vonkoch(kk: real);
procedure lev(kk: real);
procedure sierpin1(itr: longint);
procedure carp1(itr: longint);
procedure pental(itr: longint);
procedure heigh1(itr: longint);
procedure vonkoch1(itr: longint);
procedure lev1(itr: longint);
procedure sierpin2(jktk: integer; k: real);
procedure carp2(jktk: integer; k: real);
procedure penta2(jktk: integer; k: real);
procedure heigh2(jktk: integer; k: real);
procedure vonkoch2(jktk: integer; k: real);
procedure lev2(jktk: integer; k: real);
procedure determ(itr, trkx, trky: integer);
procedure stasion;
procedure statisoff;
procedure turut(a0, b0, c0, d0, e0, f0: real; btkk: integer);
procedure iterasiacak(trkx, trky: integer; jमित: longint);

```

#### implementation

```

procedure.opengraph;
var grdriver, grmode, errcode : integer;
begin
  grdriver := detect;
  initgraph(grdriver, grmode, 'c:\mathem-1\progra-1\tp7');
  errcode := graphresult;
  if errcode <> grok then begin
    writeln('Graphics error:', grapherrormsg(errcode));
    halt(1); readln;
  end;

```

```

end;
end;

procedure initmatbsr;
begin
  new(s); new(t);
  for i:=0 to n do begin
    for j:=0 to n do begin
      t^[i,j] := 0; s^[i,j] := 0;
    end;
  end;
end;

procedure freematbsr;
begin
  dispose(s); dispose(t);
end;

procedure latar;
begin
  setfillstyle(1,0); bar(0,0,getmaxx,getmaxy);
  for i:=1 to 300 do
    putpixel(random(640),random(480),random(16));
  end;

procedure polos;
begin
  setfillstyle(1,15); bar(0,0,getmaxx,getmaxy);
  setlinestyle(0,0,0); setcolor(blue);
end;

procedure layar(p: integer);
begin
  rectangle(0, 0, getmaxx, getmaxy);
  case p of
    12 : begin
      line(0, 1*round(getmaxy/3), getmaxx, 1*round(getmaxy/3));
      line(0, 2*round(getmaxy/3), getmaxx, 2*round(getmaxy/3));
      line(1*round(getmaxx/4), 0, 1*round(getmaxx/4), getmaxy);
      line(2*round(getmaxx/4), 0, 2*round(getmaxx/4), getmaxy);
      line(3*round(getmaxx/4), 0, 3*round(getmaxx/4), getmaxy);
      for i:=1 to 3 do
        for j:=1 to 4 do begin
          cx[j+4*i-4] := (j-1) * round(getmaxx/4);
          cy[j+4*i-4] := i * round(getmaxy/3);
        end;
      end;
    end;
  end;
end;

```

```

4 : begin
    line(0, round(getmaxy/2), getmaxx, round(getmaxy/2));
    line(round(getmaxx/2), 0, round(getmaxx/2), getmaxy);
    cx[1] := 0;                cy[1] := round(getmaxy/2);
    cx[2] := round(getmaxx/2); cy[2] := round(getmaxy/2);
    cx[3] := 0;                cy[3] := getmaxy;
    cx[4] := round(getmaxx/2); cy[4] := getmaxy;
end;
1 : begin
    cx[1] := 0;                cy[1] := getmaxy;
end;
end;
end;

procedure isi(x1,y1,x2,y2: integer; warna: byte);
begin
    setfillstyle(1,warna); bar(x1,y1,x2,y2); setfillstyle(1,white);
end;

procedure paku(x, y: integer);
var a: byte;
begin
    setcolor(white);
    for a:=1 to 4 do circle(x,y,a);
    setlinestyle(0,0,3); setcolor(darkgray); line(x-3,y+3,x+3,y-3);
end;

procedure icon(x1,y1,x2,y2,c: integer; warnaicon: byte;
    nama:string; warnanama:byte; shadow:boolean; mor:boolean);
begin
    if (shadow = true) then begin
        setlinestyle(0,0,3); setcolor(white);
        line((x1-2),(y1-2),(x2+2),(y1-2));
        line((x1-2),(y1-2),(x1-2),(y2+2));
        setcolor(darkgray);
        line((x2+2),(y1-2),(x2+2),(y2+2));
        line((x1-2),(y2+2),(x2+2),(y2+2));
    end;
    isi(x1,y1,x2,y2,warnaicon);
    setttextjustify(1,1); setcolor(warnanama);
    outtextxy((x1+((x2-x1)div 2)),(y2-((y2-y1)div 2)-(C+1)),nama);
    if (mor = true) then begin
        paku(x1+10,y1+10); paku(x2-10,y1+10);
        paku(x1+10,y2-10); paku(x2-10,y2-10);
    end;
    setttextjustify(lefttext,centertext);
end;

```



```

end;

procedure isian(x1,y1,x2,y2: integer);
begin
  setcolor(darkgray);
  line((x1-2),(y1-2),(x2+2),(y1-2));
  line((x1-2),(y1-2),(x1-2),(y2+2));
  setcolor(white);
  line((x2+2),(y1),(x2+2),(y2+2));
  line((x1),(y2+2),(x2+2),(y2+2));
end;

function input(temx,temy:integer;max:real;latar:byte) : real;
var n, lebar : byte; ceck,x : integer; masukan : real;
    keluar,lanjut : boolean; key:char; str10,strmax : string;
begin
  x := temx; str(max,strmax);
  setfillstyle(1,latar);
  bar(temx,temy-10,temx+2*length(strmax)+length('0'),temy+10);
  n:=0; key := ' '; str10 := ''; lebar := length('_'); masukan := 0;
  setcolor(darkgray); outtextxy(x,temy,'_');
  repeat
    setcolor(darkgray); setttextjustify(0,1);
    key := readkey;
    outtextxy(x,temy,'_');
    if (upcase(key) in['0'..'9','.',',']) and (masukan <= max) then
      begin
        if key = ',' then key := '.';
        setcolor(latar); outtextxy(x,temy,'_');
        setcolor(darkgray); outtextxy(x,temy,key);
        inc(n);
        x := x+7+length(key);
        outtextxy(x,temy,'_');
        str10 := str10+key;
        if key <> '-' then val(str10,masukan,ceck);
      end;
    if key = #27 then begin
      input := masukan; key := #0;
      exit;
    end;
    if key = #08 then begin
      if n > 0 then begin
        dec(n,1);
        setcolor(latar); outtextxy(x,temy,'0');
        x := x-7-length('H');
        outtextxy(x,temy,'0');
      end;
    end;
  until key = #0;
  input := masukan;
end;

```

```

        setcolor(darkgray); outtextxy(x,temy,'_');
    end;
    str10 := copy(str10,1,n);
    val(str10,masukan,ceck);
end;
if (masukan > max) then begin
    x := x-7-length('H');
    dec(n,1);
    bar(temx,temy-10,temx+2*length(strmax)+length('0'),temy+10);
    str10 := copy(str10,1,n);
    val(str10,masukan,ceck);
    outtextxy(temx,temy,str10+'_');
end
until key = #13;
setcolor(latar); outtextxy(x,temy,'0');
input := masukan; setttextjustify(lefttext,centertext);
end;

```

```

procedure buka;

```

```

begin

```

```

    latar; setcolor(13);
    setttextjustify(lefttext, toptext);
    setttextstyle(gothicfont, vertdir, 7);
    outtextxy(0, 0, 'VISUALISASI');
    setcolor(9); setttextstyle(2, horizdir, 16);
    outtextxy(100,25, 'Transformasi Afin');
    outtextxy(100,95, 'dan');
    outtextxy(100,165, 'Sistem Iterasi Fungsi');
    setcolor(5); setttextjustify(righttext,toptext);
    setttextstyle(3, horizdir, 4);
    outtextxy(590,280, 'dalam');
    setttextstyle(1, horizdir, 6);
    outtextxy(590,330, 'GEOMETRI FRAKTAL');
    setcolor(1); setttextjustify(centertext,toptext);
    setttextstyle(0,horizdir,0);
    outtextxy(355,450, '(C)opyright 2000 by Diena Amalia');readkey;
end;

```

```

procedure menuutama(var pilih: integer);

```

```

begin

```

```

    icon(160,110,460,350,90,7,'M E N U   U T A M A',11,true,true);
    outtextxy(185,190,'1. VISUALISASI TRANSFORMASI');
    outtextxy(185,215,'2. VISUALISASI FRAKTAL');
    outtextxy(185,240,'3. K e l u a r');
    outtextxy(185,305,'Pilih :');
    isian(250,295,295,315); pilih := round(input(260,305,100,lightgray));

```

```

    setfillstyle(0,white); bar(155,105,465,355);
end;

procedure tutup;
begin
    latar;
    icon(160,160,460,320,0,7,'S E E Y A L A T E R',11,true,true);
    readkey;
end;

procedure matriks(var p,q,r,s: real);
begin
    icon(190,160,430,320,55,7,'MARIKS TRANSFORMASI :',5,true,true);
    isian(245,220,290,240); isian(325,220,370,240);
    isian(245,260,290,280); isian(325,260,370,280);
    setcolor(blue);
    line(225,210,225,290); line(390,210,390,290);
    line(225,210,230,210); line(225,290,230,290);
    line(385,210,390,210); line(385,290,390,290);
    p := input(255,230,100,lightgray); r := input(255,270,100,lightgray);
    q := input(335,230,100,lightgray); s := input(335,270,100,lightgray);
    setfillstyle(0,white); bar(185,155,435,325);
end;

procedure macambenih(var pilih: integer);
begin
    icon(190,125,430,355,95,lightgray,'MACAM BENIH',5,true,true);
    Outtextxy(230,175,'1. Segitiga Solid');
    Outtextxy(230,190,'2. Persegi Solid');
    Outtextxy(230,205,'3. Segilima Solid');
    Outtextxy(230,220,'4. Segitiga Naga');
    Outtextxy(230,235,'5. Garis Tengah');
    Outtextxy(230,250,'6. Garis Bawah');
    Outtextxy(230,265,'7. Segitiga Kosong');
    Outtextxy(230,280,'8. Persegi Kosong');
    Outtextxy(210,325,'Pilih :');
    isian(285,315,330,335); pilih := round(input(295,325,100,lightgray));
    setfillstyle(0,white); bar(185,120,435,360);
end;

procedure bentukfraktal(var pilih: integer);
begin
    icon(190,125,430,325,80,lightgray,'BENTUK FRAKTAL',5,true,true);
    Outtextxy(220,175,'1. Gasket Sierpinski');
    Outtextxy(220,190,'2. Karpet Sierpinski');
    Outtextxy(220,205,'3. Segilima Sierpinski');

```

```

Outtextxy(220,220,'4. Naga Highway');
Outtextxy(220,235,'5. Kurva Von Koch');
Outtextxy(220,250,'6. Naga Levy '); Outtextxy(220,295,'Pilih :');
isian(285,285,330,305); pilih := round(input(295,295,100,lightgray));
setfillstyle(0,white); bar(185,120,435,330);
end;

```

```

procedure algoritma(var pilih: integer);
begin
  icon(140,160,480,320,60,7,'VISUALISASI FRAKTAL',5,true,true);
  Outtextxy(170,210,'1. Algoritma Deterministik');
  Outtextxy(170,225,'2. Algoritma Iterasi Acak');
  Outtextxy(170,240,'3. Algoritma Pendekatan Berturutan');
  Outtextxy(170,280,'Pilih :');
  isian(235,270,280,290); pilih := round(input(245,280,100,lightgray));
  setfillstyle(0,white); bar(135,155,485,325);
end;

```

```

procedure banyakitr(var pilih: longint);
begin
  icon(180,180,440,300,35,lightgray,'BANYAK ITERASI',5,true,true);
  isian(250,250,370,270);
  pilih := round(input(260,260,100000000,lightgray));
  setfillstyle(0,white); bar(155,155,465,325);
end;

```

```

procedure banyaktkk(var pilih: integer);
begin
  icon(190,160,430,320,60,7,'BANYAK BAGIAN LAYAR',5,true,true);
  Outtextxy(215,210,'1. 1 (Satu) Bagian');
  Outtextxy(215,225,'2. 4 (Empat) Bagian');
  Outtextxy(215,240,'3. 12 (Dua Belas) Bagian');
  Outtextxy(215,280,'Pilih :');
  isian(280,270,325,290); pilih := round(input(290,280,100,lightgray));
  setfillstyle(0,white); bar(185,155,435,325);
end;

```

```

procedure transformasi(p,q,r,s: real);
const tt : array [1..4, 1..4] of integer =
  ((0,0,1,0), (1,0,1,1), (1,1,0,1), (0,1,0,0));
begin
  setcolor(red);
  for i := 1 to 4 do begin
    x := tt[i,1]; y := tt[i,2];
    moveto(cx[2] + round(x*n/2), cy[2] - round(y*n/2));
    x := tt[i,3]; y := tt[i,4];
  end;
end;

```

```

    lineto(cx[2] + round(x*n/2), cy[2] - round(y*n/2));
end;
setcolor(green);
for i := 1 to 4 do begin
    x := tt[i,1]; y := tt[i,2];
    tx:= p*x + q*y; y := r*x + s*y; x := tx;
    moveto(cx[2] + round(x*n/2), cy[2] - round(y*n/2));
    x := tt[i,3]; y := tt[i,4];
    tx:= p*x + q*y; y := r*x + s*y; x := tx;
    lineto(cx[2] + round(x*n/2), cy[2] - round(y*n/2));
end;
end;

procedure gasket;
begin
    for j:=0 to round(1.732*n/2) do begin
        for i:=round(j/1.732) to round((1.732*n-j)/1.732) do begin
            t^[i,j] := 1; putpixel(i+cx[1]+trkx, cy[1]-j-trky, blue);
        end;
    end;
end;

procedure kotak;
begin
    for j:=0 to n do begin
        for i:=0 to n do begin
            t^[i,j] := 1; putpixel(i+cx[1]+trkx, cy[1]-j-trky, blue);
        end;
    end;
end;

procedure segilima;
begin
    for j:=0 to round(0.586*n) do begin
        for i:=round((0.586*n-j)/3.078) to round((j+2.485*n)/3.078) do
            begin
                t^[i,j] := 1; putpixel(i+cx[1]+trkx, cy[1]-j-trky, blue);
            end;
    end;
    for j:= round(0.586*n) to n do begin
        for i:=round((j-0.586*n)/0.727) to round((1.312*n-j)/0.727) do
            begin
                t^[i,j] := 1; putpixel(i+cx[1]+trkx, cy[1]-j-trky, blue);
            end;
    end;
end;
end;

```

```

procedure naga;
begin
  for j:=round(3*n/8) to round(5*n/8) do begin
    for i:=round(j-0.125*n) to round(1.125*n-j) do begin
      t^[i,j] := 1; putpixel(i+cx[1]+trkx, cy[1]-j-trky, blue);
    end;
  end;
end;

procedure garis;
begin
  for i:=0 to n do begin
    t^[i,round(4*n/15)] := 1;
    putpixel(i+cx[1]+trkx, cy[1]-round(4*n/15)-trky, blue);
  end;
end;

procedure kurva;
begin
  for i:=0 to n do begin
    t^[i,0] := 1; putpixel(i+cx[1]+trkx, cy[1]-0-trky, blue);
  end;
end;

procedure segitiga;
begin
  for i:=0 to n do begin
    t^[i,0] := 1; putpixel(i+cx[1]+trkx, cy[1]-0-trky, blue);
  end;
  for i:=0 to round(n/2) do begin
    t^[i,round(1.732*i)] := 1;
    putpixel(i+cx[1]+trkx, cy[1]-round(1.732*i)-trky, blue);
  end;
  for k:= round(n/2) to n do begin
    t^[k,round(1.732*n-1.732*k)] := 1;
    putpixel(k+cx[1]+trkx, cy[1]-round(1.732*n-1.732*k)-trky, blue);
  end;
end;

procedure segiempat;
begin
  for i:=0 to round(1.732*n/2) do begin
    t^[0,i] := 1; putpixel(0+cx[1]+trkx, cy[1]-i-trky, blue);
    t^[n,i] := 1; putpixel(n+cx[1]+trkx, cy[1]-i-trky, blue);
  end;
  for j:=0 to n do begin
    t^[j,0] := 1; putpixel(j+cx[1]+trkx, cy[1]-0-trky, blue);
  end;
end;

```

```

    t^[j,round(1.732*n/2)] := 1;
    putpixel(j+cx[1]+trkx, cy[1]-round(1.732*n/2)-trky, blue);
end;
end;

procedure sierpin(kk: real);
begin
    jtr := 3;
    for i := 1 to jtr do begin
        a[i] := sierpinski[i,1];    d[i] := sierpinski[i,4];
        b[i] := sierpinski[i,2];    e[i] := sierpinski[i,5]*m*kk;
        c[i] := sierpinski[i,3];    f[i] := sierpinski[i,6]*m*kk;
    end;
end;

procedure carp(kk: real);
begin
    jtr := 8;
    for i := 1 to jtr do begin
        a[i] := carpet[i,1];        d[i] := carpet[i,4];
        b[i] := carpet[i,2];        e[i] := carpet[i,5]*m*kk;
        c[i] := carpet[i,3];        f[i] := carpet[i,6]*m*kk;
    end;
end;

procedure penta(kk: real);
begin
    jtr := 5;
    for i := 1 to jtr do begin
        a[i] := pentagon[i,1];      d[i] := pentagon[i,4];
        b[i] := pentagon[i,2];      e[i] := pentagon[i,5]*m*kk;
        c[i] := pentagon[i,3];      f[i] := pentagon[i,6]*m*kk;
    end;
end;

procedure heigh(kk: real);
begin
    jtr := 2;
    for i := 1 to jtr do begin
        a[i] := heighway[i,1];      d[i] := heighway[i,4];
        b[i] := heighway[i,2];      e[i] := heighway[i,5]*m*kk;
        c[i] := heighway[i,3];      f[i] := heighway[i,6]*m*kk;
    end;
end;

procedure vonkoch(kk: real);
begin

```

```

jtr := 4;
for i := 1 to jtr do begin
  a[i] := koch[i,1];      d[i] := koch[i,4];
  b[i] := koch[i,2];      e[i] := koch[i,5]*m*kk;
  c[i] := koch[i,3];      f[i] := koch[i,6]*m*kk;
end;
end;

procedure lev(kk: real);
begin
  jtr := 2;
  for i := 1 to jtr do begin
    a[i] := levy[i,1];      d[i] := levy[i,4];
    b[i] := levy[i,2];      e[i] := levy[i,5]*m*kk;
    c[i] := levy[i,3];      f[i] := levy[i,6]*m*kk;
  end;
end;

procedure statisoft;
begin
  statif := false;
end;

procedure statison;
begin
  cx[1] := cx[4] + trkx; cy[1] := cy[4] - trky; statif := true;
end;

procedure turut(a0, b0, c0, d0, e0, f0: real; btkk: integer);
var i : integer; as, bs, cs, ds, es, fs, xs, ys, x0, y0 : real;
begin
  x0 := nd[1].x * a0 + nd[1].y * b0 + e0;
  y0 := nd[1].x * c0 + nd[1].y * d0 + f0;
  if statif then
    moveto(cx[1]+trkx+round(x0), cy[1]-trky-round(y0))
  else
    moveto(cx[btkk]+trkx+round(x0), cy[btkk]-trky-round(y0));
  for i:= 1 to jnd do begin
    xs := nd[i].x * a0 + nd[i].y * b0 + e0;
    ys := nd[i].x * c0 + nd[i].y * d0 + f0;
    if statif then
      lineto(cx[1]+trkx+round(xs),cy[1]-trky-round(ys))
    else
      lineto(cx[btkk]+trkx+round(xs),cy[btkk]-trky-round(ys));
  end;
  If statif then lineto(cx[1]+trkx+round(x0), cy[1]-trky-round(y0))

```



```

else
  lineto(cx[bktk]+trkx+round(x0), cy[bktk]-trky-round(y0));
dec(bktk);
if bktk > 0 then begin
  for i:= 1 to jtr do begin
    setcolor(i);
    as := a[i] * a0 + c[i] * b0; ds := b[i] * c0 + d[i] * d0;
    bs := b[i] * a0 + d[i] * b0; es := e[i] * a0 + f[i] * b0 + e0;
    cs := a[i] * c0 + c[i] * d0; fs := e[i] * c0 + f[i] * d0 + f0;
    turut(as, bs, cs, ds, es, fs, bktk);
  end;
end;
end;

procedure iterasiacak(trkx, trky: integer; jumit: longint);
var xbaru, ybaru : real; i : longint;
begin
  x := 0; y := 0; randomize;
  for i := 1 to jumit do begin
    j := round(random(jtr))+1;
    xbaru := a[j]*x + b[j]*y + e[j]; ybaru := c[j]*x + d[j]*y + f[j];
    x := xbaru; y := ybaru;
    putpixel(round(x) + cx[1] + trkx, cy[1]-trky-round(y), blue);
  end;
end;

procedure determ(itr, trkx, trky : integer);
begin
  for i:=0 to n do
    for j:=0 to n do
      if t^[i,j] = 1 then
        for k:=1 to jtr do
          s^[round(a[k]*i+b[k]*j+e[k]),round(c[k]*i+d[k]*j+f[k])]:=1;
    for i:=0 to n do
      for j:= 0 to n do begin
        t^[i,j] := s^[i,j]; s^[i,j] := 0;
        if t^[i,j] = 1 then putpixel(i+cx[itr]+trkx,cy[itr]-j-trky,blue);
      end;
    end;
end;
procedure sierpin1(itr: longint);
begin
  sierpin(5.3); iterasiacak(50, 10, itr);
end;
procedure carp1(itr: longint);
begin
  carp(4.5); iterasiacak(90, 15, itr);
end;

```

```

end;

procedure penta1(itr: longint);
begin
  penta(4.5); iterasiacak(90, 25, itr);
end;

procedure heigh1(itr: longint);
begin
  heigh(6); iterasiacak(40, -40, itr);
end;

procedure vonkoch1(itr: longint);
begin
  vonkoch(6); iterasiacak(20, 60, itr);
end;

procedure lev1(itr: longint);
begin
  lev(5); iterasiacak(65, -50, itr);
end;

procedure sierpin2(jktk: integer; k: real);
begin
  jnd := 3;
  for i := 1 to jnd do begin
    nd[i].x := nd3a[i].x; nd[i].y := nd3a[i].y;
  end;
  sierpin(1); turut(k, 0, 0, k, 0, 0, jktk);
end;

procedure carp2(jktk: integer; k: real);
begin
  jnd := 4;
  for i := 1 to jnd do begin
    nd[i].x := nd4[i].x; nd[i].y := nd4[i].y;
  end;
  carp(1); turut(k, 0, 0, k, 0, 0, jktk);
end;

procedure penta2(jktk: integer; k: real);
begin
  jnd := 5;
  for i := 1 to jnd do begin
    nd[i].x := nd5[i].x; nd[i].y := nd5[i].y;
  end;
  penta(1.5); turut(k, 0, 0, k, 0, 0, jktk);
end;

```

```

procedure heigh2(jktk: integer; k: real);
begin
  jnd := 3;
  for i := 1 to jnd do begin
    nd[i].x := nd3b[i].x; nd[i].y := nd3b[i].y;
  end;
  heigh(1.5); turut(k, 0, 0, k, 0, 0, jktk);
end;

```

```

procedure vonkoch2(jktk: integer; k: real);
begin
  jnd := 2;
  for i := 1 to jnd do begin
    nd[i].x := nd2a[i].x; nd[i].y := nd2a[i].y;
  end;
  vonkoch(1); turut(k, 0, 0, k, 0, 0, jktk);
end;

```

```

procedure lev2(jktk: integer; k: real);
begin
  jnd := 3;
  for i := 1 to jnd do begin
    nd[i].x := nd3b[i].x; nd[i].y := nd3b[i].y;
  end;
  lev(1.5); turut(k, 0, 0, k, 0, 0, jktk);
end;
end.

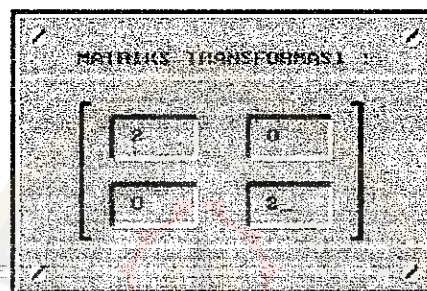
```

### Lampiran 3

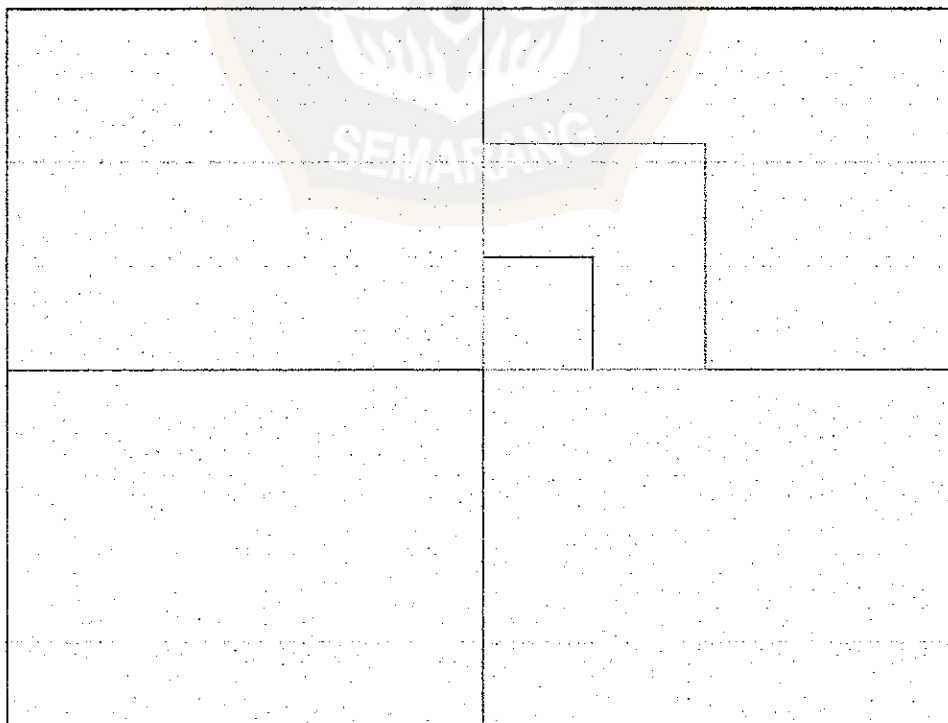
{\*\*\*\*\*}

BEBERAPA CONTOH HASIL PROGRAM FRACIENA.PAS

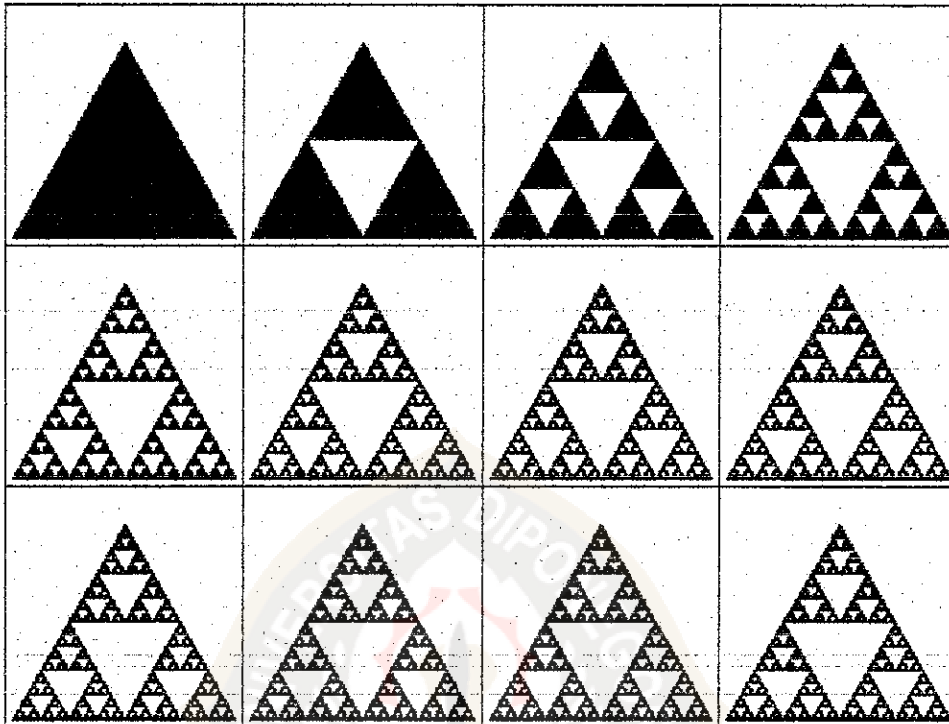
\*\*\*\*\*}



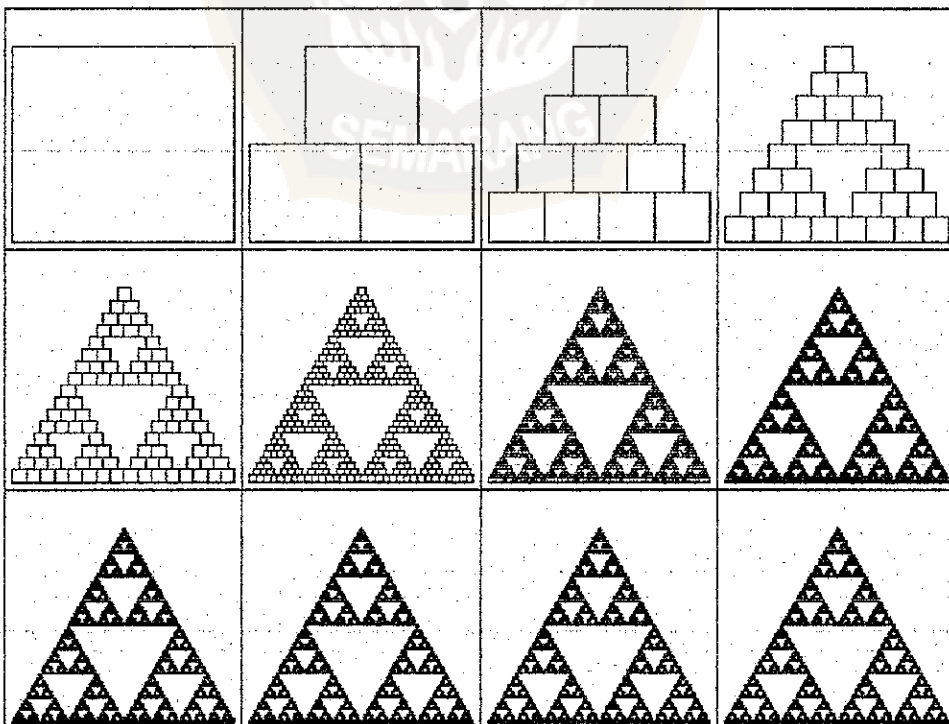
- Matriks transformasi penskalaan -



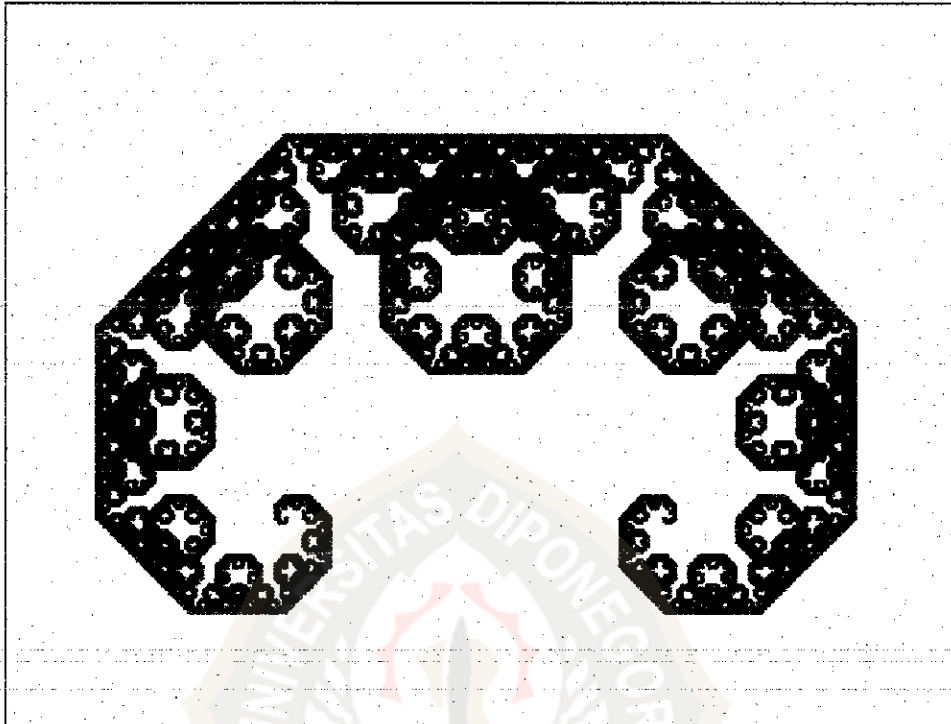
- Transformasi penskalaan -



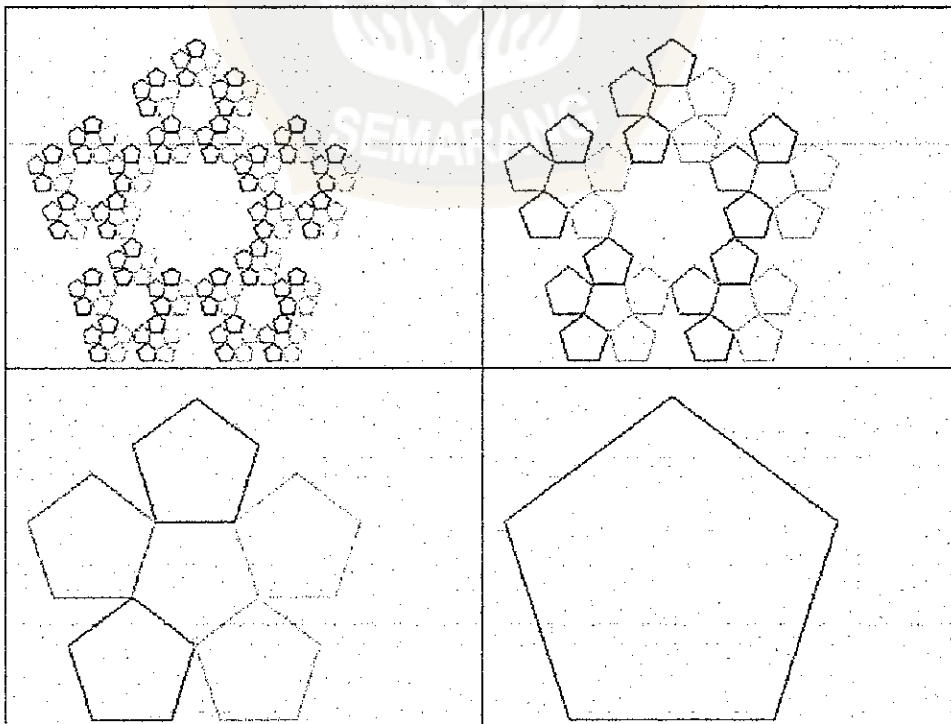
- Segitiga Sierpinski dengan algoritma deterministik, dengan benih segitiga -



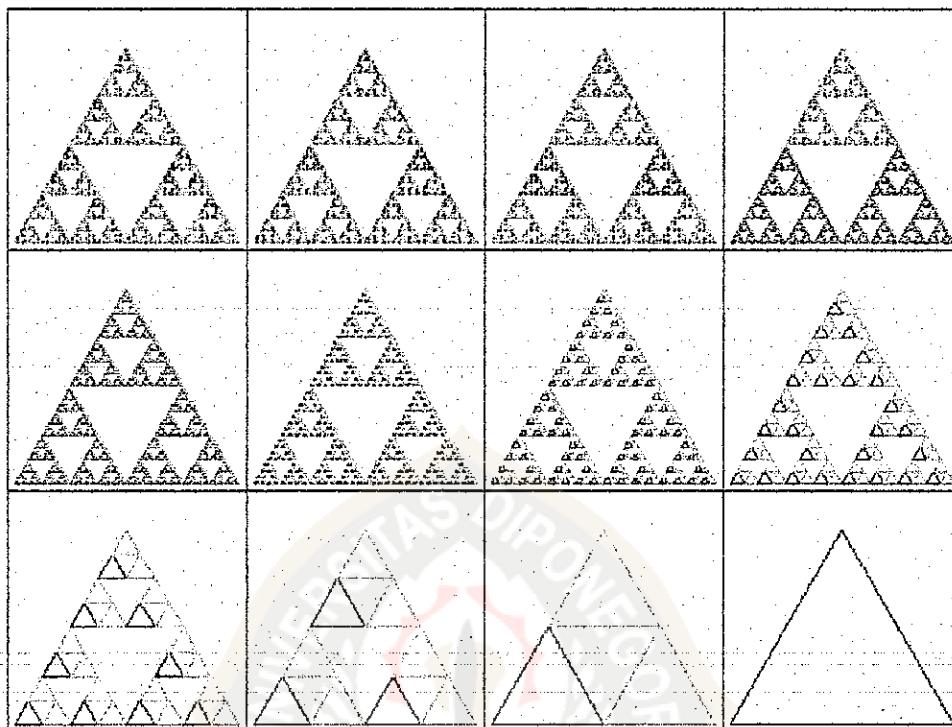
- Segitiga Sierpinski dengan algoritma deterministik, dengan benih persegi -



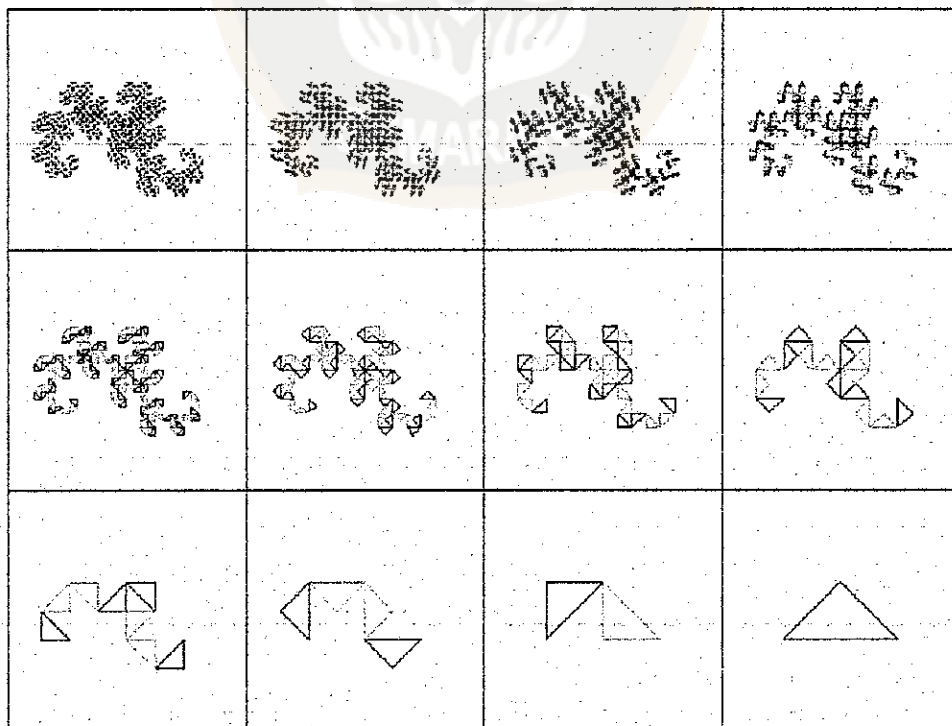
*- Naga Levy dengan algoritma iterasi acak -*



*- Segilima Sierpinski dengan algoritma pendekatan berturutan -*

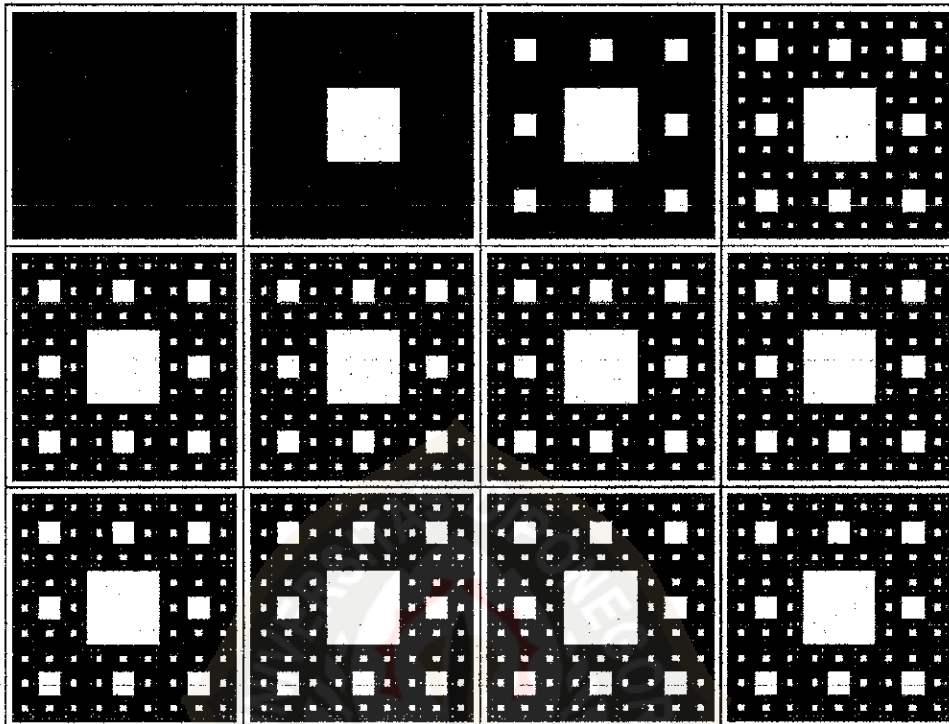


- Segitiga Sierpinski dengan algoritma pendekatan berturutan -

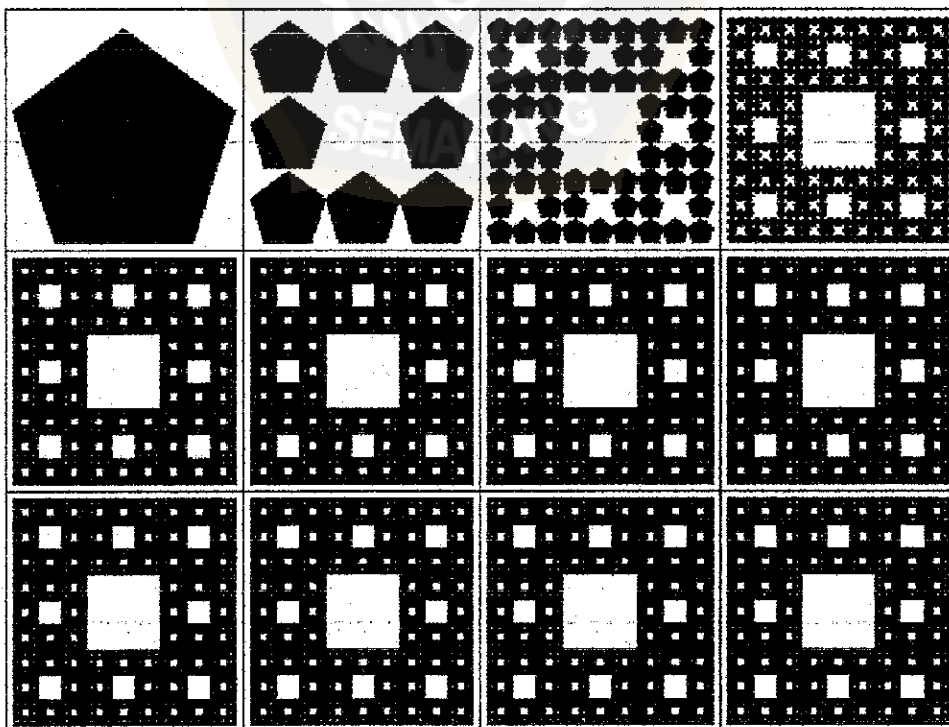


- Naga Highway Sierpinski dengan algoritma pendekatan berturutan -





- Karpets Sierpinski dengan algoritma deterministik, dengan benih persegi -



- Karpets Sierpinski dengan algoritma deterministik, dengan benih segilima-