```
Program Running_Time_Marking;{Program grafik Running-time dengan
                              metode  garbage collection}
{$D+}
{$M 65000,0,655360}
uses crt,graph,dos;
const MaxNode = 10000;
type  Pointer  =^alist;
      alist = record
            Data: integer;
            tandaKiri,tandaKanan: boolean;
            end;

var   s:array[1..MaxNode] of Pointer;randm: integer;
    jml_dt,wkt_max,y,z,n,b,J,I,jml,awl,bts_bwh,step,waktu : integer;
    jam,menit,sec,secper100:word;
    sbx,sby : real;

procedure inisialisasi;{Menginisialisasi komputer ke mode grafik}
var gd,gm : integer;
begin
   gd:= detect;
   gm:= VGAHi;
   initgraph(gd,gm,'');
   if graphresult <> grok then
   halt(1);
end;

procedure batas;{Menentukan batas bawah dari grafik}
begin
bts_bwh := round(getmaxy*0.65);
end;
procedure layar;{Menentukan latar belakang dan warna garis
                 tepi layar pada grafik}
begin
   setlinestyle(0,0,3);
   setcolor(blue);
   setBkcolor(white);
   rectangle(3,3,getmaxx-3,getmaxy-3);
end;

procedure sumbuxy;{Membuat sumbu x dan y serta meletakkan
                   suatu pointer titik pusat sb. xy}
begin
   setlinestyle(0,0,3);
   setcolor(red);
   line(80,5,80,bts_bwh+10);
```

```
    moveto(80,5);
    lineto(75,10);
    lineto(85,10);
    lineto(80,5);
    line(80,bts_bwh+10,getmaxx-20,bts_bwh+10);
    moveto(getmaxx-20,bts_bwh+10);
    lineto(getmaxx-25,bts_bwh+5);
    lineto(getmaxx-25,bts_bwh+15);
    lineto(getmaxx-2,bts_bwh+10);
    moveto(80,bts_bwh+10);
end;

procedure skalasbx(mulai,usai:integer);{Memberi skala pada
sb. x sekaligus bilangannya sesuai dengan awal pengeplotan
grafik dan jumlah data seluruhnya}
var s: integer; noskx,judul1,judul2,judul3 :string;
begin
    for s:= 1 to 11 do
    begin
    setcolor(red);
    outtextxy(77+(s-1)*52,bts_bwh+8,'I');
    str(mulai+((usai-mulai)div 10)*(s-1),noskx);
    outtextxy(60+(s-1)*52,bts_bwh+20,noskx);
    end;
    setcolor(red);
    judul1 :='Jumlah Data';
    judul2 :='Gambar : Grafik Running Time Marking';
    judul3 :='       Menggunakan Garbage Collection';
    settextstyle(2,0,5);
    outtextxy(300,bts_bwh+40,judul1);
    settextstyle(2,0,6);
    outtextxy(150,bts_bwh+70,judul2);
    outtextxy(150,bts_bwh+90,judul3);
    settextstyle(0,0,1);
end;

procedure skalasby;{Memberi skala sb. y dan juga ditampilkan
                    judul sumbu y}
var s:integer; tot:real;
   nosky,judul3:string;
begin
    wkt_max:= bts_bwh+10-10;
    step:=bts_bwh div 6;
    for s:= 1 to 6 do
    begin
        outtextxy(80,bts_bwh+10-s*step,'-');
```

```
        str(s*500,nosky);
        outtextxy(40,bts_bwh+10-s*step,nosky);
        setcolor(red);
    end;
    setcolor(red);
    judul3:='Dalam 1/100 detik';
    settextstyle(2,1,4);
    outtextxy(15,bts_bwh-200,judul3);
    settextstyle(0,0,1);
    moveto(80,bts_bwh+10);
end;

{procedure Tanda;{Memberikan tanda pada setiap node yang
                  digunakan}
begin
    for I:= 1 to MaxNode do
    begin
    s[I]^.tandaKiri := false;
    end;
    if s[I]^.tandaKiri = false then
    begin
    s[I]^.tandaKiri := true;
    end;
    if s[I]^.tandakanan = false then
    s[I]^.Data := I
    else
    begin
    I := I + 1;
    end;
    if s[I]^.Data = MaxNode then
    begin
    for I:= MaxNode downto 1 do
    begin
    s[I]^.tandaKiri := false;
    end;
    if s[I]^.tandaKiri = false then
    begin
    s[I]^.tandaKiri := true;
    end;
    if s[I]^.tandakanan = false then
    s[I]^.Data := I
    else
    begin
    I:=I+1;
    end;
end;
```

```
end;}

procedure Tandai;
begin
begin
    for I:= MaxNode downto 1 do
    begin
    s[I]^.tandaKiri := false;
    end;
    if  s[I]^.tandaKiri = false then
    s[I]^.tandaKiri := true;
    if s[I]^.tandakanan = false then
    s[I]^.Data := I
    else
    I := I +1;
    end;
end;



procedure tutup;{Menutup mode grafik dan merubah ke mode
                 layar dengan prosedur close graph}
begin
    writeln('');
    settextstyle(2,0,7);
    settextjustify(1,2);
    outtextxy(getmaxx div 2,19*getmaxy div 20,'Tekan <esc>');
    repeat until readkey = #27;
    closegraph;
end;

{program utama};{Pengeplotan grafik diawali dengan pengini-
              lisasian pembangkit bil. random dengan
                 prosedur randomize}
begin
textbackground(0);clrscr;
write('Random angka 0..n      =');readln(randm);
write('Banyaknya data (maximal 6000) =');readln(jml_dt);
write('Awal ploting data pada grafik =');readln(awl);
write('step ploting data             =');readln(step);
y:=awl;
repeat
    randomize;
    for z:= 1 to y  do s[z]^.Data:= random(randm);
    settime(0,0,0,0);
    for z := 1 to y do tandai;
```

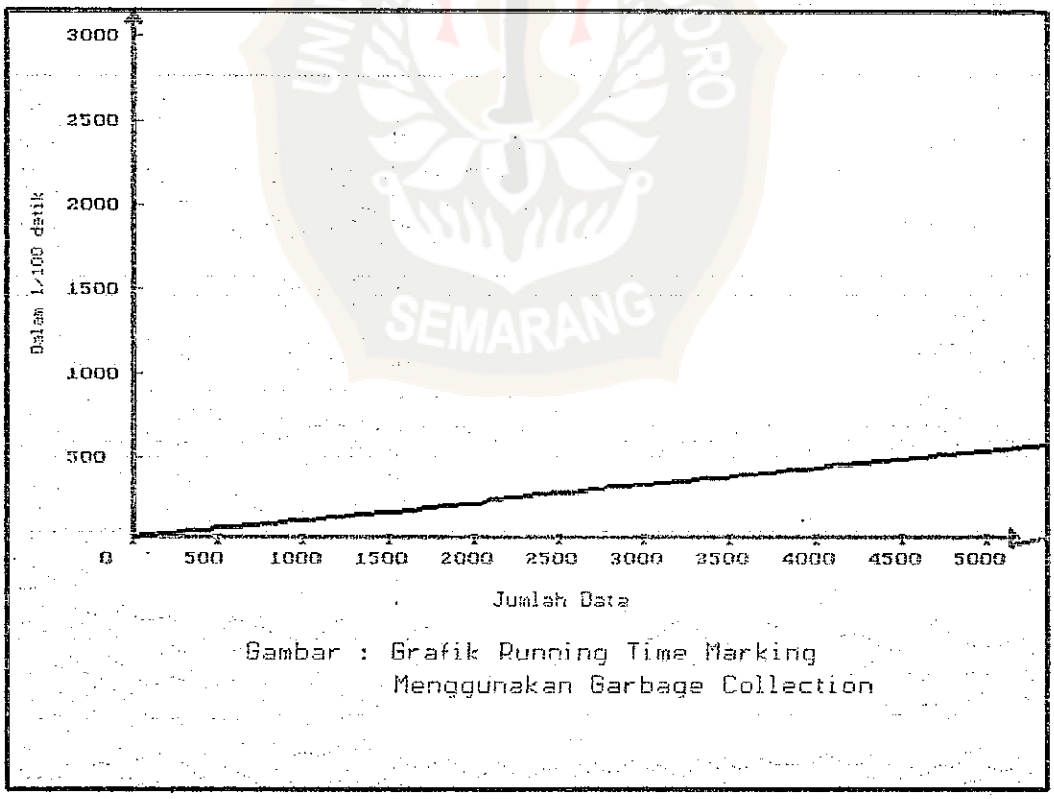
```
    gettime(jam,menit,sec,secper100);
    waktu := jam*60*60*100+menit*60*100+sec*100+secper100;
    writeln('Waktu dengan jumlah data',y:8,   'adalah',waktu:4,  .'per 100 detik');
    y := y+ step;
until y > jml_dt;writeln;writeln;writeln;
writeln('UNTUK MENAMPILKAN GRAFIK <<< Tekan sebarang tombol >>>');
repeat until keypressed;
inisialisasi;batas;layar;sumbuxy;
skalasbx(awl,jml_dt);
skalasby;
y:= awl;
repeat
if y >= awl then
begin
   randomize;
   for z:= 1 to  y do s[z]^.Data :=random(randm);
   settime(0,0,0,0);
   for z:= 1 to y do tandai;
   gettime(jam,menit,sec,secper100);
   waktu:= jam*60*60*100+menit*60*100+sec*100+secper100;
   sbx := (y-awl)*((560)/(jml_dt-awl));
   if waktu <= 3000 then
   begin
      sby :=wkt_max/3000*waktu;
      setcolor(blue);setlinestyle(0,0,3);
      lineto(round(sbx)+80,(bts_bwh+10)-round(sby));
   end
   else
   y:=jml_dt;
   y:=y+step;
end;
until y>jml_dt;
tutup;
end.
```

```
Random angka 0..n       =2000
Banyaknya data (maximal 6000) =5000
Awal ploting data pada grafik =0
step ploting data             =100
Waktu dengan jumlah data                    0adalah
0per 100 detik
Waktu dengan jumlah data                  100adalah
10per 100 detik
Waktu dengan jumlah data                  200adalah
21per 100 detik
Waktu dengan jumlah data                  300adalah
32per 100 detik
Waktu dengan jumlah data                  400adalah
38per 100 detik
Waktu dengan jumlah data                  500adalah
54per 100 detik
Waktu dengan jumlah data                  600adalah
65per 100 detik
Waktu dengan jumlah data                  700adalah
71per 100 detik
Waktu dengan jumlah data                  800adalah
87per 100 detik
Waktu dengan jumlah data                  900adalah
93per 100 detik
Waktu dengan jumlah data                 1000adalah
104per 100 detik
Waktu dengan jumlah data                 1100adalah
120per 100 detik
Waktu dengan jumlah data                 1200adalah
126per 100 detik
Waktu dengan jumlah data                 1300adalah
137per 100 detik
Waktu dengan jumlah data                 1400adalah
148per 100 detik
Waktu dengan jumlah data                 1500adalah
159per 100 detik
Waktu dengan jumlah data                 1600adalah
170per 100 detik
Waktu dengan jumlah data                 1700adalah
175per 100 detik
Waktu dengan jumlah data                 1800adalah
192per 100 detik
Waktu dengan jumlah data                 1900adalah
203per 100 detik
Waktu dengan jumlah data                 1900adalah
203per 100 detik
```

Waktu dengan jumlah data 2000adalah 14192per 100 detik
Waktu dengan jumlah data 2100adalah 263per 100 detik
Waktu dengan jumlah data 2200adalah 258per 100 detik
Waktu dengan jumlah data 2300adalah 258per 100 detik
Waktu dengan jumlah data 2400adalah 263per 100 detik
Waktu dengan jumlah data 2500adalah 280per 100 detik
Waktu dengan jumlah data 2600adalah 285per 100 detik
Waktu dengan jumlah data 2700adalah 296per 100 detik
Waktu dengan jumlah data 2800adalah 307per 100 detik
Waktu dengan jumlah data 2900adalah 318per 100 detik
Waktu dengan jumlah data 3000adalah 340per 100 detik
Waktu dengan jumlah data 3100adalah 340per 100 detik
Waktu dengan jumlah data 3200adalah 351per 100 detik
Waktu dengan jumlah data 3300adalah 1186per 100 detik
Waktu dengan jumlah data 3400adalah 400per 100 detik
Waktu dengan jumlah data 3500adalah 395per 100 detik
Waktu dengan jumlah data 3600adalah 400per 100 detik
Waktu dengan jumlah data 3700adalah 406per 100 detik
Waktu dengan jumlah data 3800adalah 417per 100 detik
Waktu dengan jumlah data 3900adalah 422per 100 detik
Waktu dengan jumlah data 4000adalah 439per 100 detik
Waktu dengan jumlah data 4100adalah 450per 100 detik
Waktu dengan jumlah data 4200adalah 461per 100 detik

```
Waktu dengan jumlah data                4300adalah
488per 100 detik
Waktu dengan jumlah data                4400adalah
1158per 100 detik
Waktu dengan jumlah data                4500adalah
499per 100 detik
Waktu dengan jumlah data                4600adalah
499per 100 detik
Waktu dengan jumlah data                4700adalah
505per 100 detik
Waktu dengan jumlah data                4800adalah
527per 100 detik
Waktu dengan jumlah data                4900adalah
538per 100 detik
Waktu dengan jumlah data                5000adalah
543per 100 detik

UNTUK MENAMPILKAN GRAFIK <<< Tekan sebarang tombol >>>
```



Gambar : Grafik Running Time Marking
Menggunakan Garbage Collection

```
Program Mencari_Node_Garbage;
uses crt;
const MaxNode = 1006;
type
    karyawan = ^catatankaryawan;
    catatankaryawan = record
                      tanda: boolean;
                      kode : string;
                      kanan: karyawan;
                      end;
var Datakaryawan,lagi : array[1..MaxNode]of karyawan;
    Bebas : array[1..MaxNode] of karyawan;
    KodeAwal : karyawan;
    I,J:integer;
     {proses penandaan}
     procedure Tandai;
     begin
     for I := 1 to  MaxNode do
     if  not Datakaryawan[I]^.tanda then
     Datakaryawan[I]^.tanda := true;
     writeln('Banyaknya Node yang ditandai ='
,MaxNode:8);
     end;
     {proses pengumpulan}
     procedure Kumpulkan;
     begin
     J:= 0;
     for I := 1 to MaxNode do
     if Datakaryawan[I]^.tanda then
     Datakaryawan[I]^.tanda := false;
         inc(J);
         Bebas[J] := Datakaryawan[I];
         inc(I);
         writeln('Banyaknya Node yang bebas    ='
,((memAvail)/66):8);
         end;

begin
     clrscr;
     writeln('heap asli =',maxAvail:8);

Writeln('=========================================');
     writeln('Kondisi heap setelah data ditandai');

writeln('=========================================');
     KodeAwal := nil;
     for I:= 1 to MaxNode do
```

```
        new(Datakaryawan[I]);
        write('Kode =',Datakaryawan[I]^.kode);
        Datakaryawan[I]^.kanan := KodeAwal;
        KodeAwal := Datakaryawan[I];
        inc(Datakaryawan[I]);
        writeln;
        writeln('Banyaknya data yang disimpan
=',sizeof(Datakaryawan):8);
        writeln('Banyaknya heap yang tersisa
=',maxAvail:8);
        Tandai;
        Kumpulkan;

writeln('=========================================');

end.
```

```
heap asli =  551584
==========================================
Kondisi heap setelah data ditandai
==========================================
Kode =
Banyaknya data yang disimpan =     4024
Banyaknya heap yang tersisa  =   286000
Banyaknya Node yang ditandai =     1006
Banyaknya Node yang bebas    = 4.3E+03
==========================================
```

A. Algoritma Marking pertama :

```
1.  Procedure Mark1(x: ListPointer);
2.  var P,Q : ListPointer; done : boolean;
3.  Begin
4.  Top := 0; add(x); {meletakkan x pada tumpukan}
5.  While top > 0 do { tumpukan tidak kosong}
6.  Begin
7.  Delete(P); {Mengambil tumpukan} done := false;
8.  Repeat
9.  Q := P^.Rlink;
10. If  Q <> nil
11.  Then begin
12. if Q^.tag andnot Q^.Mark then add(Q);
13. Q^.Mark := true;
14. End;
15. P:= P^.Dlink;
16. If P <> nil then
17. If P^.Mark or not P^.tag then done := true
18. Else P^.Mark := true
19. Else done := true;
20. Until done;
21. If P <> nil then P^.Mark := true;
22. End; {while}
23. End; {Mark1}
```

B. Algoritma Marking kedua :

```
1.  Procedure Mark2(x : ListPointer);
2.  Var P,Q,T : ListPointer; failure : boolean;
3.  Begin
4.  P := x; T := nil; {inisialisasi senarai path T-x}
5.  Repeat
6.  MoveDown;
7.  If failure then begin
8.  MoveRight;
9.  If failure then backup;
10. End;
11. Until failure;
12. End;{Mark2}
```

```
1.  Procedure MoveDown;
2.  Begin
3.  Q := P^.Dlink; failure := true;
4.  If Q <> nil
5.  Then if not Q^.Mark and Q^.tag
6.  Then begin
7.  Q^.Mark := true; P^.tag := false;
8.  P^.Dlink := T; T:= P; { menambah senarai path T-x}
9.  P:= Q; failure := false;
10. End
11. Else Q^.Mark := true;
12. End;{MoveDown}
```

```
1.  Procedure MoveRight;
2.  Begin
3.  Q:= P^.Rlink; failure := true;
4.  If Q <> nil
5.  Then if not Q^.Mark and Q^.tag
6.  Then begin
7.  Q^.Mark := true; P^.Rlink := T;
8.  T:= P; P:= Q; failure := false;
9.  End
10. Else Q^.Mark := true;
11. End; {MoveRight}
```

```
1.  Procedure backup;
2.  Begin
3.  Failure := true;
4.  While (T<> nil) and failure do
5.  Begin
6.  Q := T;
7.  If not Q^.tag
8.  Then begin{menghubungkan melalui Dlink}
9.  T := Q^.Dlink; Q^.Dlink := P;
10. Q^.tag := true; P:= Q;
11. MoveRight;
12. End
13. Else begin{menghubungkan melalui Rlink}
```

```
14. T := Q^.Rlink; Q^.Rlink := P;
15. P := Q;
16. End;
17. End;{while}
18. End;{backup}
```