

Output Primitive : point, garis,
lingkaran

Pertemuan : 3-4

Dosen Pembina :

Sriyani Violina

Danang Junaedi

1

Nine-year-old writes hit iPhone app

Programming for the under-tens

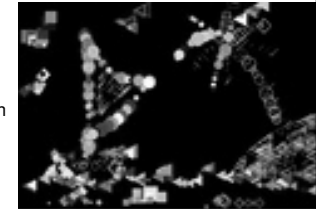
You might think you're pretty hot stuff because you've figured out how to change your Facebook status from your iPhone, but you've got nothing on nine-year-old Lim Ding Wen.

This young prodigy from Singapore is fluent in six programming languages, according to a [BBC report](#) this week, and his newest creation, an iPhone drawing game called Doodle Kids, has racked up over 4,000 downloads in just two weeks. He wrote it for his younger sisters, who love to draw.

Doodle Kids, which lets players sketch with their fingers on the iPhone's screen and shake it, Etch-A-Sketch-style, to clear, has already racked up a healthy three-and-a-half star rating on the App Store. One reviewer commented: "Awesome app!...Amazing that something like this was made by a 9 year old".

<http://videogames.yahoo.com/feature/nine-year-old-writes-hit-iphone-app/1287368> Tanggal Akses : 10 Februari 2009 14:28

Next Question is : "Kitaa khaapaaaannnn...???"



2

Output Primitif : Garis

3

Overview

- Tujuan
- Konsep Output Primitif
- Titik
- Konsep penggambaran Garis
- Algoritma Pembuatan Garis : DDA
- Algoritma Pembuatan Garis : Bresenham
- Studi Kasus

4

Tujuan Instruksional

- Memahami primitif geometri titik dan garis
- memahami dan membedakan algoritma pembuatan garis
- Menganalisis algoritma pembuatan garis baik DDA maupun bresenham
- Mengimplementasikan algoritma pembuatan garis dalam program

5

Konsep Output Primitives

- Graphics output primitives
 - Functions used to describe the various picture components
 - Examples: car, house, flower, ...
- Geometric primitives
 - Functions used to describe points, lines, triangles, circles, ...

6

Titik

- Titik digambar dengan cara mengkonversi posisi koordinat tunggal dari program aplikasi ke operasi/fungsi alat output yang digunakan.
- Contoh pada monitor CRT electron beam diset on untuk memberi cahaya pada layar fosfor pada lokasi tertentu.

7

Fungsi Pada Titik

- Koordinat
- Warna
- Ukuran (Size)

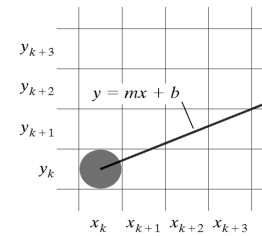
8

Garis

- Garis digambarkan dengan cara menghitung posisi intermediate sepanjang jalur garis antar dua posisi titik. (titik awal dan titik akhir)
- Alat output kemudian mengisi posisi antara titik tersebut.
- sistem analog seperti vector pen plotter atau random-scan display, garis lurus dapat digambar secara halus dari titik awal ke titik akhir.
- Sistem Digital menampilkan garis dengan mem-plot titik-titik antara titik awal dan titik akhir. Koordinat titik-titik tersebut didapat dari perhitungan persamaan garis.

9

Algoritma Penggambaran Garis



Persamaan garis :

$$y = m \cdot x + b$$

Jika diketahui dua titik (x_1, y_1) dan (x_2, y_2) maka dari persamaan garis dapat dihitung

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$b = y_1 - m \cdot x_1$$

10

Contoh

- Gambarkan garis dengan titik awal $(0,0)$ dan titik akhir $(4,3)$

Jawab:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3 - 0}{4 - 0} = \frac{3}{4}$$

$$y_1 = m \cdot x_1 + b$$

$$b = 0 - 0 = 0$$

Maka persamaan garis : $y = 3/4x$

x	Y
0	0
1	0.75
2	1.5
3	2.25
4	3

11

Algoritma DDA

- DDA atau Digital Differential Analyzer adalah scan conversion algorithm yang didasari oleh perhitungan berikut :

$$\Delta y = m \Delta x$$

$$\Delta x = \frac{\Delta y}{m}$$

12

DDA (Digital Differential Analyser) Algorithm

$$y = m \cdot x + c$$

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{\Delta y}{\Delta x}$$

• Jika $0 < m < 1 \rightarrow$

$$y_{k+1} = y_k + m$$
$$x_{k+1} = x_k + 1$$

• Jika $m > 1 \rightarrow$

$$x_{k+1} = x_k + 1/m$$
$$y_{k+1} = y_k + 1$$

13

DDA Algorithm

```
void lineDDA (int x0, int y0, int xEnd, int yEnd)
{
    int dx = xEnd - x0, dy = yEnd - y0, steps, k;
    float xIncrement, yIncrement, x = x0, y = y0;
    if (fabs (dx) > fabs (dy))
        steps = fabs (dx);
    else
        steps = fabs (dy);
    xIncrement = float (dx) / float (steps);
    yIncrement = float (dy) / float (steps);
    setPixel (round (x), round (y));
    for (k = 0; k < steps; k++) {
        x += xIncrement;
        y += yIncrement;
        setPixel (round (x), round (y));
    }
}
```

14

Line Drawing Algorithms

- Advantage
 - Does not calculate coordinates based on the complete equation (uses offset method)
- Disadvantage
 - Round-off errors are accumulated, thus line diverges more and more from straight line
 - Round-off operations take time
 - Perform integer arithmetic by storing float as integers in numerator and denominator and performing integer arithmetic.

15

Bresenham Algorithm

- Sumbu vertikal memperlihatkan posisi scan line.
- Sumbu horizontal memperlihatkan kolom pixel
- Pada tiap langkah, penentuan pixel selanjutnya didasari oleh parameter integer yang nilainya proporsional dengan pengurangan antara vertical separations dari dua posisi piksel dari nilai actual.

16

Bresenham Algorithm

1. Input the two line endpoints and store the left endpoint in (x_0, y_0) .
2. Set the color for the frame-buffer position (x_0, y_0) – i.e. plot the first point.
3. Calculate the constant $2\Delta y - \Delta x$, and set the starting value for the decision parameter as $p_0 = 2\Delta y - \Delta x$.
4. At each x_k along the line, from $k=0$, perform the following test:
if $p_k < 0$, next point to plot is $(x_k + 1, y_k)$ and $p_{k+1} = p_k + 2\Delta y$
otherwise, next point to plot is $(x_k + 1, y_k + 1)$
and $p_{k+1} = p_k + 2\Delta y - 2\Delta x$
5. Repeat step 4 $\Delta x - 1$ times.

17

Bresenham's Algorithm (1)

```

P0 = 2Δy - Δx
Plot (X0, Y0)
K = 0
Repeat
  If Pk < 0 then
    Plot (Xk+1, Yk)
    Pk+1 = Pk + 2Δy
  Otherwise
    Plot (Xk+1, Yk+1)
    Pk+1 = Pk + 2Δy - 2Δx
Until (K = (Δx - 1))
    
```

18

Bresenham's Algorithm (2)

```

void lineBresenham(int x0, int y0, int x1,
int y1)
{
  dy = y1 - y0
  dx = x1 - x0
  if (dy < 0) then
    dy = -dy; stepy = -1
  else stepy = 1
  if (dx < 0)
    dx = -dx; stepx = -1;
  else stepx = 1
  dy = 2*dy
  dx = 2*dx
  Plot (x0, y0)

  if (dx > dy)
    P = 2*dy - dx
    while (x0 != x1)
      if (P >= 0) then y0 = y0 +
stepy; P = P - 2*dx
      x0 = x0 + stepx; P = P - 2*dy
      Plot(x0, y0)
    else
      P = 2 * dx - dy
      While (Y <> Yend)
        If P >= 0 Then X = X + stepX; P
= P - dy
        Y = Y + stepY; P = P + dx
        Plot (X, Y)
  }
    
```

<http://www.cs.unc.edu/~mcmillan/comp136/Lecture6/Lines.html>, 20 Februari 2010

19

Contoh Soal

```

garis (20,10,30,18)
ΔX = 10, ΔY = 8
P0 = 2ΔY - ΔX = 6
2.ΔY = 16
2ΔY - 2ΔX = -4
    
```

K	P _k	(X _{k+1} , Y _{k+1})
0	6	(20,10)
1	2	(21,11)
2	-2	(22,12)
3	14	(23,12)
4	10	(24,13)
5	6	(25,14)
6	2	(26,15)
7	-2	(27,16)
8	14	(28,16)
9	14	(29,17)
10	10	(30,18)

20

Studi Kasus

- Hitung proses pembentukan garis dengan menggunakan persamaan garis, algoritma DDA dan algoritma Bresenham
 - garis (0,0,4,3)
 - garis (1,3,8,5)

21

Output Primitif : Lingkaran

22

Overview

- Tujuan
- Konsep penggambaran Lingkaran
- Algoritma Midpoint Circle

23

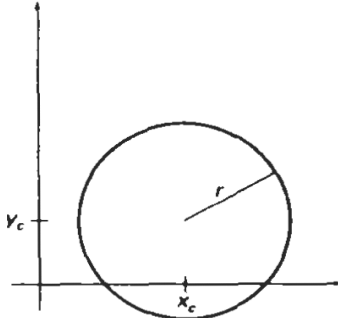
Tujuan Instruksional

- Memahami primitif geometri lingkaran
- Menganalisis algoritma midpoint circle
- Mengimplementasikan algoritma midpoint circle dalam program

24

Lingkaran

- Lingkaran didefinisikan sebagai sekumpulan garis yang memiliki jarak yang sama dari titik pusat yaitu r .



25

Persamaan Lingkaran

- Persamaan lingkaran dengan titik pusat (x_c, y_c) dan jari-jari r adalah sebagai berikut :

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

- Dari persamaan tersebut dapat dihitung posisi titik pada lingkaran dengan menentukan nilai tertentu pada sumbu x dan menghitung y

$$y = y_c \pm \sqrt{r^2 - (x_c - x)^2}$$

26

Persamaan Lingkaran

- Lingkaran dapat digambarkan dengan menggunakan persamaan koordinat rectangular tsb akan tetapi pendekatan ini menimbulkan dua masalah yaitu :
 - Persamaan tersebut mengandung perhitungan akar yang operasinya memakan waktu.
 - Timbul gap yang cukup signifikan pada lingkaran ketika digambarkan.

27

Koordinat Polar

- Cara lain untuk menggambarkan lingkaran adalah dengan menggunakan koordinat polar yaitu :

$$x = x_c + r \cdot \cos \theta$$

$$y = y_c + r \cdot \sin \theta$$

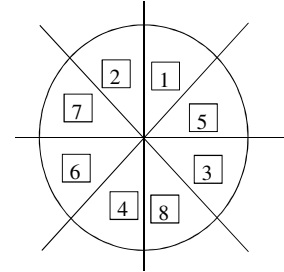
28

Midpoint Circle Algorithm

- Untuk mengatasi masalah yang timbul dari penerapan koordinat polar maupun rectangular, Bresenham menyusun suatu algoritma pembentukan lingkaran yang hanya menggunakan aritmetika integer. Secara prinsip algoritma ini sejenis dengan algoritma penggambaran garis yang disusun oleh orang yang sama.
- Lingkaran merupakan objek yang simetris sehingga karakteristik ini dapat dimanfaatkan untuk mengurangi pekerjaan pada saat menggambar lingkaran. Lingkaran dibagi menjadi 8 oktan (lihat gambar 3.x), misalkan kita menyusun algoritma untuk menggambarkan lingkaran di oktan pertama, maka koordinat untuk 7 oktan selanjutnya dapat ditentukan.

29

Tabel Pencermiran titik



Oktan	x	y
1	x	y
2	-x	y
3	x	-y
4	-x	-y
5	y	x
6	-y	x
7	y	-x
8	-y	-x

30

Midpoint Circle

1. Input radius r and circle center (x_c, y_c) , then set the coordinates for the first point on the circumference of a circle centered on the origin as $(x_0, y_0) = (0, r)$.
2. Calculate the initial value of the decision parameter as $p_0 = 1 - r$
3. At each x_k , from $k=0$, perform the following test:
if $p_k < 0$, next point to plot along the circle centered on $(0,0)$ is $(x_k + 1, y_k)$ and $p_{k+1} = p_k + 2x_{k+1} + 1$
otherwise, next point to plot is $(x_k + 1, y_k - 1)$ and $p_{k+1} = p_k + 2x_{k+1} + 1 - 2y_{k+1}$
where $2x_{k+1} = 2x_k + 2$, and $2y_{k+1} = 2y_k - 2$
4. Determine symmetry points in the other seven octants.
5. Move each calculated pixel position (x, y) onto the circular path centered at (x_c, y_c) and plot the coordinate values:
 $x = x + x_c, y = y + y_c$
6. Repeat steps 3 through 5 until $x \geq y$.

31

Midpoint Circle

$(X_0, Y_0) = (0, r)$; Plot (X_0, Y_0)

$P_0 = 1 - r$

Repeat

If $P_k < 0$ then

$X = X + 1$

Plot (X, Y)

$P_{k+1} = P_k + 2X + 1$

Otherwise

$X = X + 1; Y = Y - 1$

Plot (X, Y)

$P_{k+1} = P_k + 2X - 2Y + 1$

Until $X \geq Y$

32

Midpoint Circle Algorithm – cont.

- Example : $x^2 + y^2 = 100$
- Penentuan koordinat di oktan 1

k	2X	2Y	P _k	(X,Y)
-	0	20	-9	(0,10)
0	2	20	-6	(1,10)
1	4	20	-1	(2,10)
2	6	20	6	(3,10)
3	8	18	-3	(4,9)
4	10	18	8	(5,9)
5	12	16	5	(6,8)
6	14	14	6	(7,7)

33

Studi Kasus

- Dengan menggunakan midpoint circle algorithm, buat lingkaran berdasarkan persamaan berikut ini :
 - $x^2 + y^2 = 64$
 - $x^2 + y^2 = 49$

34

Output Primitif : Ellipse

35

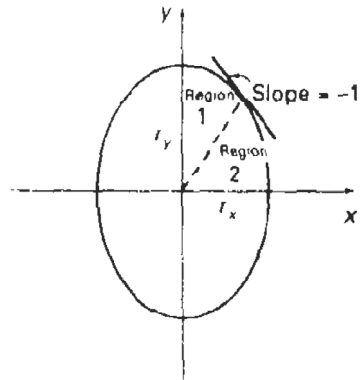
Ellipse

- Ellips didefinisikan sebagai himpunan titik yang jumlah jarak dari 2 posisi yang tetap adalah sama untuk setiap titik.
- Persamaan Ellips dengan pusat 0,0 dan jari-jari Rx dan Ry adalah :

$$\frac{(x - x_c)^2}{R_x} + \frac{(y - y_c)^2}{R_y} = 1$$

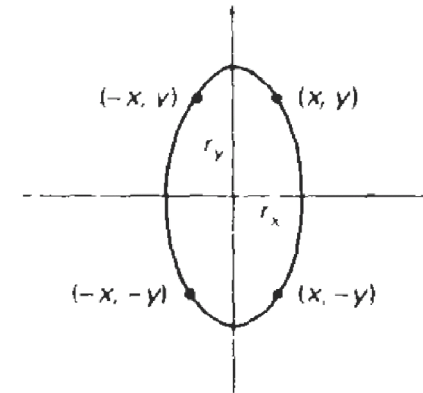
36

- Elips merupakan objek yang memiliki empat bagian yang simetris, dari karakteristik tsb dapat disusun suatu algoritma yang memplot pixel di kuadran pertama dan menentukan titik di tiga kuadran lainnya.



37

Pencerminan titik



38

Algoritma Pembentukan elips

- Kuadran pertama dibagi menjadi 2 (dua) region dan dengan menggunakan algoritma midpoint ellipse, plot titik untuk region pertama, kemudian koordinat akhir pada region I menjadi koordinat awal untuk region II.
- Region 1 dan 2 dapat digunakan dengan berbagai macam cara. Pertama dimulai dari posisi (0,r) dan melangkah searah jarum jam sepanjang jalur ellips pada kuadran pertama. Pergeseran dengan unit step dalam x pada saat slope lebih besar dari 1.

39

Midpoint Ellipse Algorithm

Region I :

```

(x0,y0)=(0,ry); Plot (x0,y0)
P0 = ry2 - rx2ry+1/4rx2
Px = 0; Py=2rx2y
While Px<Py
  x=x+1
  Px = Px + 2ry2
  If Pk ≥ 0
    y=y-1
    Py = Py - 2rx2
  Pk+1 = Pk+ry2+Px·Py
  Otherwise
    Pk+1 = Pk+ry2+Px
  Plot(x,y)
End While

```

Region II :

```

Plot (x0,y0)
P0 = ry2 (x+1/2)2+rx2(y-1)- rx2ry2
While y>0
  y=y-1
  Py = Py - 2rx2
  If Pk ≤ 0
    x=x+1
    Px = Px + 2ry2
  Pk+1 = Pk+rx2+Px·Py
  Otherwise
    Pk+1 = Pk+rx2·Py
  Plot(x,y)
End While
Ket : titik akhir di region I menjadi titik awal region II

```

40

Contoh soal

- Sebuah ellipse dengan $R_x = 8$ dan $R_y = 5$, pusat (0,0)
Region 1

k	Px	Py	Pk	x	y
0	0	640	-279	0	5
1	50	640	-204	1	5
2	100	640	-79	2	5
3	150	640	96	3	5
4	200	512	-191	4	4
5	250	512	84	5	4
6	300	384	25	6	3
7	350	256	144	7	2

41

- Region 2

k	Px	Py	Pk	x	y
-	350	256	-129.75	7	2
0	400	128	206.25	8	1
1	400	0	270.25	8	0

42

Studi Kasus

Gambarkan Elips dengan spesifikasi berikut :

1. Pusat (0,0), $R_x=15$ dan $R_y=2$
2. Pusat (0,0), $R_x=6$ dan $R_y=2$
3. Pusat (0,0), $R_x=8$ dan $R_y=6$

$$\frac{x^2}{15} + \frac{y^2}{2} = 1$$

43