

Menggambar Garis dan Lingkaran dengan Algoritma Bresenham Teguh Susyanto²⁾

Abstract

In computer graphics, graphic depiction of the complex forms is actually composed of a group of primitive objects in an image area, so that the whole will form a particular object. Depictions of these objects are actually a primitive generating clusters of adjacent pixel color according to the form object to be depicted. The forms of the simplest object that can underlie complex graphical form is an object lines and circles. There are several famous primitive methods for drawing lines and circles that is using either the Digital Differential Analyzer (DDA) or Bresenham algorithms. In this paper the author tries to present the implementation of Bresenham's method to draw the line and circle with Delphi.

Key words: line, circle, Bresenham's Algorithms

I. Pendahuluan

Dari sudut pandang aplikasi telah disediakan fasilitas-fasilitas untuk menggambar obyek-obyek grafis secara praktis, namun kadang fasilitas-fasilitas tersebut tidak memberikan penjelasan detail proses penggambaran obyek-obyek grafis yang dimaksudkan.

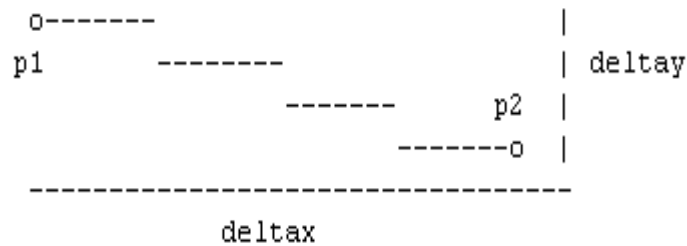
Bentuk-bentuk grafis yang rumit sebenarnya tersusun dari pembangkitan kumpulan-kumpulan pixel yang berdekatan sesuai bentuk obyek yang akan digambarkan. Adapun bentuk-bentuk obyek yang paling sederhana yang dapat mendasari bentuk grafis yang kompleks adalah obyek garis dan lingkaran. Ada beberapa metode primitive untuk menggambar garis dan lingkaran meliputi metode *Digital Differential Analyzer* (DDA) ataupun metode Bresenham's ataupun dengan metode-metode lainnya.

Dalam tulisan ini mencoba mengimplementasikan algoritma Bresenham untuk menggambar garis dan lingkaran dalam bahasa pemrograman Delphi.

²⁾ Staf Pengajar STMIK Sinar Nusantara Surakarta

II. Metode Bresenham

II.1. Menggambar Garis



Gambar 1. Menggambar Garis

Algoritma pembangkit garis bresenham dapat diuraikan sebagai berikut :

1. Masukkan dua titik akhir p1 dan p2 dan simpan titik akhir sebelah kiri dalam (x_0, y_0)
2. Isikan (x_0, y_0) ke dalam frame buffer; yang akan digambar pada titik pertama
3. Hitung konstanta $deltax$, $deltay$, $2deltay$ dan $2deltay - 2deltax$, dan dapatkan nilai awal untuk parameter keputusan sebagai :

$$p_0 = 2deltay - deltax$$

4. Pada masing-masing x_k sepanjang garis, dimulai dengan $k=0$, lakukan tes berikut: jika $p_k < 0$, dan titik berikutnya yang akan digambar adalah $(x_k + 1, y_k)$ dan

$$p_{k+1} = p_k + 2deltay$$

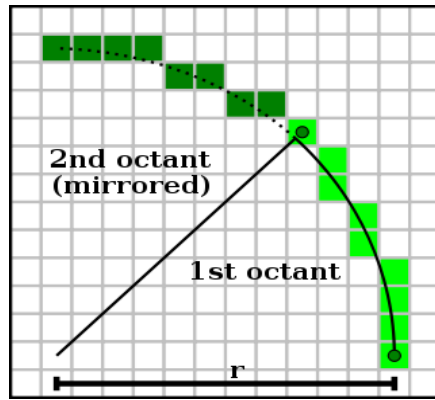
selain itu, titik berikutnya yang digambar adalah (x_k+1, y_k+1) dan

$$p_{k+1} = p_k + 2deltay - 2deltax$$

5. Ulangi langka 4 sebanyak $deltax$ kali

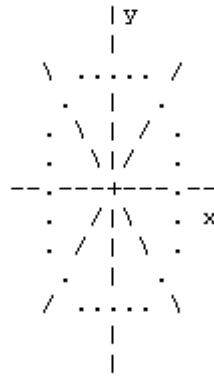
II.2. Menggambar Lingkaran

Algoritma lingkaran Bresenham's menghitung lokasi dari pixel dalam 45 derajat pertama. Hal ini diasumsikan bahwa lingkaran berpusat pada koordinat tengah.



Gambar 2. Menggambar Lingkaran

Jadi untuk setiap pixel (x, y) yang dijadikan acuan menggambar sebuah pixel di masing-masing dari 8 oktan lingkaran:



```

PutPixel (CenterX + X, CenterY + Y)
PutPixel (CenterX + X, CenterY - Y)
PutPixel (CenterX - X, CenterY + Y)
PutPixel (CenterX - X, CenterY - Y)
PutPixel (CenterX + Y, CenterY + X)
PutPixel (CenterX + Y, CenterY - X)
PutPixel (CenterX - Y, CenterY + X)
PutPixel (CenterX - Y, CenterY - X)

```

Berdasarkan algoritma bresenham ini, pada mulanya diberikan inisialisasi :

```

d := 3 - (2 * RADIUS)
x := 0

```

```
y := RADIUS
```

Untuk setiap pixel akan dilakukan operasi penggambaran 8 pixel lingkaran:

```
if d < 0 then
    d := d + (4 * x) + 6
else
    begin
        d := d + 4 * (x - y) + 10
        y := y - 1;
    end;
```

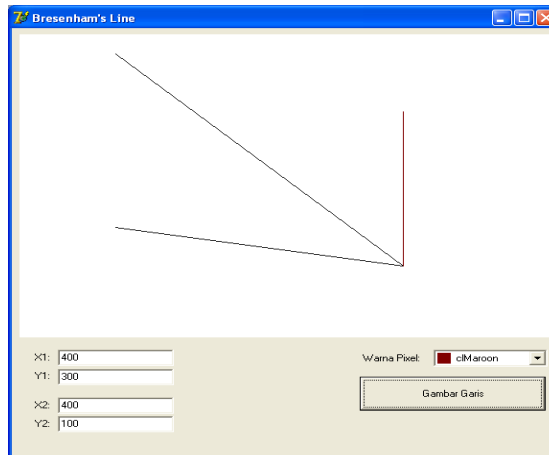
III. Implementasi program dan pembahasan

Penerapan algoritma bresenham untuk pembangkitan garis dan lingkaran ini digunakan bahasa pemrograman Delphi yang disusun dalam bentuk aplikasi visual sederhana. Untuk penggambaran titik (pixel) digunakan Obyek TCanvas pada komponen TImage.

```
procedure TForm1.PutPixel(x, y:integer);
begin
    Image1.Canvas.Pixels[x, y]:=ColorBox1.Selected;
end;
```

III.1. Menggambar Garis

Implementasi penggambaran obyek garis dengan metode bresenham ditayangkan dalam tampilan berikut ini :



Gambar 3. Implementasi penggambaran obyek garis dengan metode bresenham

Proses penggambaran garis dengan algoritma bresenham dituangkan dalam prosedur **Line** yang memerlukan 4 parameter input yaitu koordinat Titik awal x_1, y_1 dan koordinat Titik akhir x_2, y_2 .

```
procedure Line(x1, y1, x2, y2:integer);
```

Untuk dapat menggambar garis pada awalnya dihitung nilai delta yaitu deltax dan deltay berdasarkan dari dua nilai titik akhir yang telah diinputkan

```
deltax := abs(x2 - x1);  
deltay := abs(y2 - y1);
```

kemudian dilakukan inisialisasi semua variable dengan menentukan x atau y yang bersifat independent berdasarkan besar nilai delta-nya, jika nilai deltax lebih besar maka variable x akan bersifat independent dan begitu pula sebaliknya.

```
if deltax >= deltay then  
  begin  
    { x is independent variable }  
    numpixels := deltax + 1;  
    d := (2 * deltay) - deltax;  
    dinc1 := deltay shl 1;  
    dinc2 := (deltay - deltax) shl 1;  
    xinc1 := 1;  
    xinc2 := 1;  
    yinc1 := 0;  
    yinc2 := 1;  
  end  
else  
  begin  
    { y is independent variable }  
    numpixels := deltay + 1;  
    d := (2 * deltax) - deltay;  
    dinc1 := deltax shl 1;  
    dinc2 := (deltax - deltay) shl 1;  
    xinc1 := 0;  
    xinc2 := 1;  
    yinc1 := 1;  
    yinc2 := 1;  
  end;  
end;
```

Untuk memastikan penggambaran dilakukan ke arah kanan maka dilakukan perbandingan terhadap nilai titik awal dan titik akhir, yaitu nilai titik (x_1, y_1) dengan (x_2, y_2) , jika nilai titik awal baik x_1 dan atau y_1

lebih besar terhadap nilai titik akhir x2 dan atau y2 maka nilai increment ditambah dengan minus, untuk variable xinc1, xinc2 dan yinc1, yinc2.

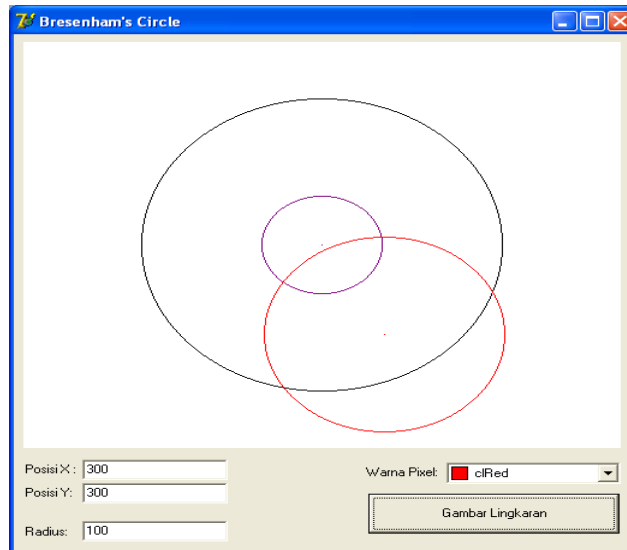
```
if x1 > x2 then
  begin
    xinc1 := - xinc1;
    xinc2 := - xinc2;
  end;
if y1 > y2 then
  begin
    yinc1 := - yinc1;
    yinc2 := - yinc2;
  end;
```

Proses pembangkitan pixel yang dinyalakan akan dimulai dari koordinat x1, y1 dan dilanjutkan secara berulang pada pixel-pixel yang terdekat. Adapun source code nya sebagai berikut :

```
x := x1;
y := y1;
for i := 1 to numpixels do
  begin
    PutPixel(x, y);
    if d < 0 then
      begin
        d := d + dincl;
        x := x + xinc1;
        y := y + yinc1;
      end
    else
      begin
        d := d + dinc2;
        x := x + xinc2;
        y := y + yinc2;
      end;
  end;
end;
```

III.2. Menggambar Lingkaran

Implementasi program untuk menggambar lingkaran ditampilkan seperti window sebagai berikut :



Gambar 4. Implementasi program untuk menggambar lingkaran

Untuk menggambar lingkaran pada program ini dituliskan metode **rasterCircle** dengan memiliki 3 parameter input yaitu koordinat titik pusat (x,y) dan radius lingkaran r.

```
procedure TForm1.rasterCircle(x0, y0, radius:integer);
```

Sebelum menggambar lingkaran inialisasi beberapa variable berikut ini :

```
f := 1 - radius;
ddF_x := 1;
ddF_y := -2 * radius;
x := 0;
y := radius;
```

dilanjutkan dengan menggambar 4 pixel arah titik koordinat pada radius sisi kanan, kiri, atas dan bawah pada dari titik pusat lingkaran.

```
PutPixel(x0, y0 + radius);
PutPixel(x0, y0 - radius);
PutPixel(x0 + radius, y0);
PutPixel(x0 - radius, y0);
```

Kemudian akan dilakukan iterasi penggambaran pixel dalam 8 titik posisi selama kondisi x lebih kecil dari pada y.

```

while (x < y) do
begin
  if(f >= 0) then
  begin
    y:=y-1;
    ddF_y :=ddF_y + 2;
    f := f + ddF_y;
  end;
  x:=x+1;
  ddF_x := ddF_x + 2;
  f := f + ddF_x;
  PutPixel(x0 + x, y0 + y);
  PutPixel(x0 - x, y0 + y);
  PutPixel(x0 + x, y0 - y);
  PutPixel(x0 - x, y0 - y);
  PutPixel(x0 + y, y0 + x);
  PutPixel(x0 - y, y0 + x);
  PutPixel(x0 + y, y0 - x);
  PutPixel(x0 - y, y0 - x);
end
end;

```

IV. Penutup

Dalam penggambaran obyek garis maupun lingkaran akan dilakukan proses pengaktifan sekumpulan pixel-pixel yang berdekatan, namun proses pengaktifan (penyalaaan) pixel- pixel ini tidak sepenuhnya presisi terhadap koordinat yang sebenarnya dilintasi. Penerapan metode Bresenham's ini akan memberikan keputusan pembangkitan pixel pada koordinat tertentu yang akan diberi warna tertentu yaitu dengan berdasarkan besar nilai delta x dan delta y.

V. Daftar Pustaka

Frank Klawon, 2008, *Introduction to Computer Graphics Using Java 2D dan 3D*, Srpinger-Verlag, London

James D. Foley, 1995, *Computer Graphics: Principles and Practice in C Second Edition*, Addison-Wesley Publishing Company.

<http://www.gamedev.net/reference/articles/article767.asp> diakses tanggal 11 Februari 2010

http://en.wikipedia.org/wiki/File:Bresenham_circle.svg diakses tanggal 11 Februari 2010