KURVA BEZIER DAN BRESENHAM UNTUK PEMBUATAN LINGKARAN

Djoni Haryadi Setiabudi
Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra
e-mail: djonihs@petra.ac.id

ABSTRAK: Salah satu primitif yang penting di komputer grafik adalah pembuatan lingkaran. Untuk menggambar bentuk lingkaran diperlukan suatu metode tertentu seperti metode Bezier dan algoritma Bresenham. Pada metoda Bezier menggunakan titik-titik kontrol poligon untuk membuat lingkaran. Sedangkan pada algoritma Bresenham menggunakan translasi titik koordinat.

Pada penelitian ini dibandingkan kedua metode yaitu metode Bezier dan algoritma Bresenham. Kedua metode ini akan dibandingkan berdasarkan kecepatan proses dalam pembuatan lingkaran dan akurasi hasil penggambaran lingkaran oleh masing-masing metode. Tujuan penelitian ini untuk mengetahui metode pembuatan lingkaran yang paling baik. Untuk penelitian ini digunakan bahasa pemrograman Borland Delphi.

Dari hasil penelitian, didapatkan bahwa algoritma Bresenham memiliki kecepatan proses 1.44 kali lebih cepat dari Bezier untuk 70 titik penggambaran, sedangkan akurasi dalam pembuatan lingkaran metode Bezier lebih baik, dimana error untuk koordinat X Bezier lebih kecil 0,038379671 dari koordinat X Bresenham sedangkan error untuk koordinat Y Bezier lebih kecil 0,026411257dari koordinat Y Bresenham.

Kata kunci: Bezier, Bresenham, lingkaran, komputer grafik.

ABSTRACT: *One of the primitive in computer graphics is a circle. It needs a special method to draw a circle like Bezier method and Bresenham algorithm. According to Bezier method, it use polygon control points to draw circle but it use coordinate points of translation on Bresenham algorithm.*

In this research, there are two methods were compared namely Bezier's method and Bresenham algorithm. Both of them were comparing according to speed and accuration in drawing a circle. The purpose of this research is to know which one is better to draw a good circle. It is used of Borland Delphi programming language for implementation.

The result of this research shows Bresenham algorithm had 1.44 times faster than Bezier for 70 drawing points, however for accuration, the Bezier's method is better. The error of Bezier X coordinate is 0,038379671 smaller than that of Bresenham X coordinate, and the error of Bezier Y coordinate is 0,026411257 less than Bresenham Y coordinate.

Keywords: Bezier, Bresenham, circle, computer graphics.

1. PENDAHULUAN

Pembuatan suatu lingkaran merupakan salah satu primitive yang berperan dalam grafika komputer. Prinsip yang dipakai pada penelitian ini dalam pembuatan lingkaran adalah menterjemahkan dari bahasa matematika ke dalam bahasa pemrograman untuk metode Bezier dan algoritma Bresenham. Untuk program-program terapan sederhana, lingkaran yang diperoleh dari bentuk dasar kurva cukup memadai, tetapi seringkali diinginginkan bentuk lingkaran yang jauh lebih rumit dan tidak teratur. Bentuk fungsi matematis yang paling utama untuk menggambar kurva sebagai dasar pembuatan bentuk lingkaran adalah fungsi parametrik atau vector-valued function yaitu untuk sembarang titik pada suatu permukaan, lokasinya ditentukan oleh dua buah parameter u dan v biasanya bernilai antara 0 dan 1, dan fungsi parametrik x,y, dan z merupakan lokasi-lokasi titik pada kurva atau permukaan.

Algoritma de Casteljau [2] merupakan algoritma untuk membuat kurva menggunakan sejumlah titik kontrol, dan menggunakan teknik *in-betweening* untuk mendapatkan kurva yang diinginkan. Algoritma ini dikembangkan oleh P. de Casteljau, dan merupakan cikal bakal kurva Bezier, yang secara terpisah dikembangkan lebih lanjut oleh P. Bezier. Algoritma de Casteljau untuk membuat kurva Bezier cukup ampuh secara algoritmik, tetapi tidak secara eksplisit menyatakan bentuk fungsionalnya. Karena alasan ini, kemudian dikembangkan persamaan lain untuk membuat kurva Bezier, yang sangat berguna untuk tujuan analisis.

Algoritma Bresenham dipilih karena merupakan metode dasar grafis yang sangat populer dan terkenal efisiensinya, Bresenham memakai dasar *integer arithmetic* yang jauh lebih cepat daripada *floating-point arithmetic* yang biasa dipakai dan menggunakan suatu persamaan matematis untuk mengetahui adanya baris atau kolom baru yang akan dibuat dalam proses pembuatan suatu garis. Tetapi Bresenham juga memiliki kekurangan yaitu timbulnya error jika dua segmen garis yang overlap dibuat, error juga akan timbul jika sebuah segmen garis yang intensitasnya berbeda overlap terhadap suatu segment garis yang sudah ada.

2. ALGORITMA

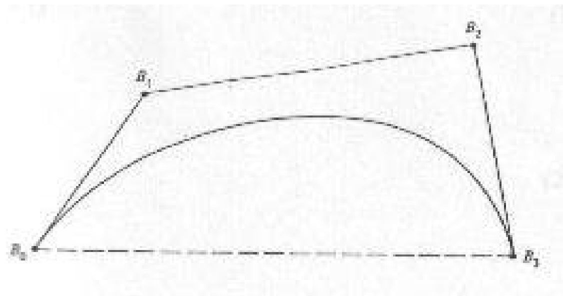
2.1 Algoritma BEZIER

Metode kurva Bezier didefinisikan oleh titik-titik kontrol poligon seperti ditunjukkan pada gambar 1. Kurva Bezier menggunakan fungsi blending yang juga adalah basis Bernstein sehingga beberapa macam dari kurva Bezier dapat diketahui. Beberapa definisi dari kurva Bezier yaitu :

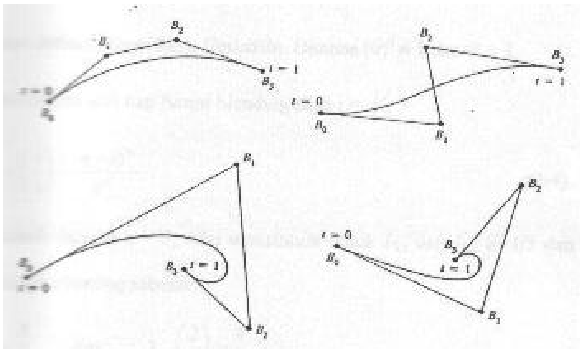
- Fungsi basis adalah real.
- Tingkat definisi polinomial segmen kurva adalah satu lebih kecil dari jumlah definisi titik-titik poligon.
- Kurva mengikuti bentuk dari definisi poligon.
- Titik-titik awal dan akhir dari kurva tepat sama dengan titik-titik awal dan akhir dari definisi poligon.
- Arah vector di ujung-ujung dari kurva mempunyai arah yang sama dengan awal dan akhir dari bentuk poligon.
- Kurva didalam convex hull dari definisi poligon.

Pada gambar 2 ditunjukkan contoh untuk empat titik poligon Bezier dan kurva yang dihasilkan. Sehingga cepat dipelajari dan diperkirakan bentuk dari kurva yang dihasilkan oleh suatu poligon Bezier. Kurva Bezier dengan suatu parameter t memiliki persamaan matematika yang didefinisikan sebagai berikut :

$$P(t) = \sum_{i=0}^{n} B_i J_{n,i}(t) \quad 0 \le t \le 1$$



Gambar 1. Kurva Bezier Dengan Definisi Poligon



Gambar 2. Beberapa Kurva Bezier dengan Empat titik Kontrol Poligon

dengan fungsi blending atau basis Bernstein adalah :

$$J_{n,1}(t) = \binom{n}{i} t^i (1-t)^{n-1}$$

dengan

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

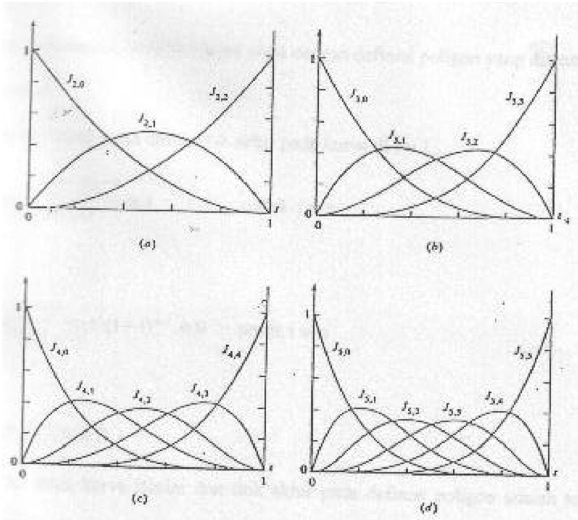
keterangan :

- $J_{n,i}$ adalah fungsi factor blending atau basis Bernstein
- $\binom{n}{i}$ adalah koefisien binomial[3]
- B_i adalah titik kontrol

$J_{n,i}$(t) adalah tingkat ke-n yang ke-I dari fungsi basis Bernstein. Disini n adalah derajat dari definisi fungsi basis Bernstein. Dimana $(0)^0 \equiv 1$ dan $0! \equiv 1$.

Harga maksimum dari tiap fungsi blending pada t = i/n

$$J_{n,i}\left(\frac{i}{n}\right)=\binom{n}{i}\frac{i^i(n-i)^{n-i}}{n^n}$$



Gambar 3. Kurva Fungsi Blending Bernstein

Titik akhir pada kurva Bezier dan titik akhir pada definisi poligon adalah tepat sama. Hasil fungsi blending ini ditunjukkan pada gambar 3. Selanjutnya ditunjukkan bahwa untuk memberikan nilai pada parameter t, hasil penjumlahan terakhir dari fungsi basis adalah tepat sama dengan satu, yaitu:

$$\sum_{i-0}^{n} B_i J_{n,i}(t)=1$$

2.2 Algoritma BRESENHAM

Algoritma Bresenham menggunakan aritmatika integer yang tidak memerlukan perkalian dan pembagian dalam proses perhitungannya didalam seluruh implementasi, yang mana aritmatika integer ini memiliki kecepatan perhitungan yang lebih tinggi daripada aritmatika floating point.

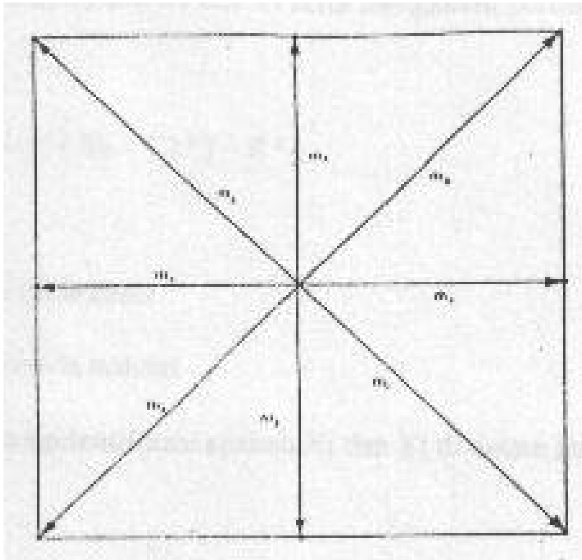
Algoritma Bresenham memberikan persamaan umum untuk lingkaran sebagai berikut:

$$(X-a)^2+(Y-b)^2=R^2$$

Dengan (Xa,Ya) sebagai koordinat awal dan (Zt,Yt) sebagai koordinator akhir. Persamaan umum lingkaran yang diberikan oleh Algoritma Bresenham di atas diturunkan dari persamaan umum lingkaran :

$X^2 + Y^2 = R^2$
Keterangan:
X = absis ≥ 0
Y = ordinat ≥ 0
R = bilangan integer ≥ 1.

Pada algoritma pembuatan lingkaran dengan arah penggambaran searah jarum arah jarum jam untuk lintasan yang besarnya seperempat lingkaran atau 90^o memiliki tiga titik perhitungan yaitu m1, m2, dan m3 seperti yang digambarkan pada gambar 2.5 di bawah ini:



Gambar 4. Titik Penggambaran Lintasan Lingkaran Algoritma Bresenham

Seperti terlihat pada gambar 4, pada suatu titik Pi dengan koordinat (Xi , Yi) akan bergerak diantara m1 ke (Xi + 1 , Yi) pada sudut 0^o, m2 ke (Xi + 1 , Yi – 1) pada sudut 315^o, dan m3 ke (Xi , Yi – 1) pada sudut 270^o.

Dimana ketiga lintasan tersebut memiliki besar sebagai berikut:

- Untuk m1 ke (Xi + 1 , Yi) besarnya $\left|\{(Xi+1)^2+Yi^2\}-R^2\right|$
- Untuk m2 ke (Xi + 1 , Yi – 1) besarnya $\left|\{(Xi+1)^2+(Yi-1)^2\}-R^2\right|$
- Untuk m3 ke (Xi , Yi – 1) besarnya $\left|\{Xi\ \ +(Yi-1)^2\}-R^2\right|$

jadi pada algoritma ini hanya dilakukan evaluasi terhadap dua buah titik saja pada setiap langkahnya yaitu Xi dan Yi serta mengamati perubahan harga Δi yang besarnya:

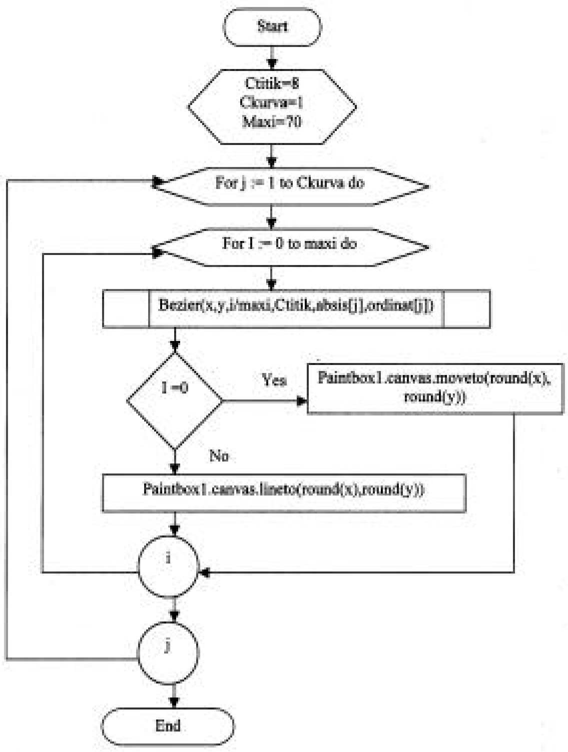
$$\Delta i = \{[(Xi+1)^2 + (Yi-1)^2] - R^2\}$$

Keterangan:
Xi adalah translasi pada absis.
Yi adalah translasi pada ordinat.

Δi adalah untuk mengidentifikasi apakah Xi dan Yi di dalam atau di luar lingkaran.

3. FLOWCHART

3.1 Bezier

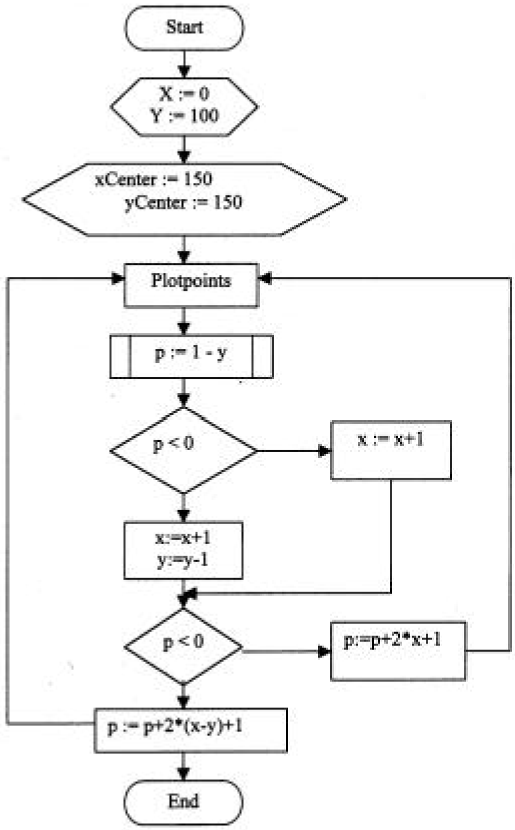


Gambar 5. Flowchart Program Lingkaran Bezier

Fungsi procedure dibawah ini sebagai algoritma utama metode Bezier.

```
procedure bezier(var x,y:real; t:real;
n:integer; absis,ordinat:larik1d);
var i: integer;
    b:real;
begin
  x:=0; y:=0;
  for I:=0 to n do
  begin
     b:=basis(I,n,t);
     x:=x+absis[I]*b;
     y:=y+ordinat[I]*b;
  ed;
end;
```

3.2 Bresenham



Gambar 6. Flowchart Procedure Bresenham

Fungsi procedure dibawah ini sebagai algoritma utama Bresenham untuk menggambar lingkaran.

```
begin
x:=0;
y:=100;
xCenter:=150;
yCenter:=150;
plotpoints;
p:=1-y;
while x < y do
        begin
            if p<0 then
                x:=x+1
            else
                begin
                    x:=x+1;
                    y:=y-1;
                end;
             if p<0 then
                 p:=p+2*x+1;
             else
                  p:=p+2*(x-y)+1;
              plotpoints
              end;
end;
```

4. PENGUJIAN DAN ANALISIS

Hasil percobaan penghitungan waktu yang dilakukan sebanyak 25 kali, untuk penggambaran lingkaran berjari-jari 100 sebanyak 100 kali. Untuk nilai rata-rata 1 kali penggambaran lingkaran didapatkan hasil seperti pada tabel 1.

Tabel 1. Rata-rata Hasil Penghitungan Waktu

	Bezier (ms)	Bresenham (ms)
1	8,8	6,6
2	8,3	6,6
3	8,8	5,5
4	8,8	6,0
5	8,3	6,0
6	8,8	6,0
7	8,8	6,1
8	8,3	6,6
9	8,8	6,0
10	8,8	6,0
11	8,8	5,5
12	8,8	6,1
13	8,8	6,0
14	8,8	6,0
15	8,8	5,5
16	8,8	6,1
17	8,8	6,0
18	8,8	6,0
19	8,8	6,0
20	8,3	6,0
21	8,8	5,5
22	8,3	6,1
23	8,8	6,0
24	8,8	6,0
25	8,8	6,0

Hasil perhitungan rata-rata kecepatan proses diatas diperoleh dari pengujian terhadap penggambaran satu lingkaran penuh dengan metode Bezier dan seperdelapan bagian lingkaran dengan algoritma Bresenham tetapi dengan jumlah titik potong yang sama sebesar 70 titik, sehingga akan didapatkan bahwa algoritma Bresenham lebih cepat 2,692 ms dari metode Bezier.

Nilai titik-titik koordinat sepanjang penggambaran lingkaran dari masing-masing metode dapat dilihat pada tabel 2 kolom 1 dan 3 untuk koordinat X dan tabel 3 kolom 1 dan 3 untuk koordinat Y. Pada tabel 2 kolom 2 dan 4, kemudian tabel 3 kolom 2 dan 4 memperlihatkan hasil perhitungan koordinat real untuk X dan Y pada setiap metode. Perhitungan koordinat real didapatkan dengan substitusi koordinat-koordinat X dan Y yang bersesuaian ke dalam persamaan umum lingkaran $X^2 + Y^2 = R^2$.

Persamaan untuk menghitung koordinat Y lingkaran real adalah sebagai berikut:

$X1 = X - 150$

$Y1 = \sqrt{R^2 - X1^2}$

$Y = 150 - Y1$

Sedangkan persamaan untuk menghitung koordinat X lingkaran real adalah sebagai berikut:

$Y1 = 150 - Y$

$X1 = \sqrt{R^2 - Y1^2}$

$X = 150 + X1$

Tabel 2. Nilai Real Y

Bezier x	Real y Bezier	Bresenham x	Real y Bresenham
150	50	150	50
163,5656542	50,9244075	164	50,9848496
175,9622953	53,4289938	176	53,4391383
187,2476206	57,1958257	187	57,0968246
197,4720415	61,9863347	197	61,7333585
206,6796924	67,6142459	207	67,8355307
214,9093666	73,9291506	215	74,0065792

Tabel 3. Nilai Real X

Bezier y	Real x Bezier	Bresenham y	Real x Bresenham
50	150	50	150
50,9560652	163,9749411	51	164,106736
53,2624472	175,3346774	53	174,3104916
56,7677968	186,1629131	57	186,7559519
61,3283500	196,2313583	61	195,5960525
66,8079413	205,4894708	67	205,7763391
73,0779702	213,8983672	73	213,8357267

Dengan menghitung selisih antara nilai koordinat X dan Y pada penggambaran lingkaran dengan nilai x dan y real dari hasil perhitungan persamaan umum lingkaran maka akan diperoleh harga rata-rata error koordinat X dan Y untuk setiap metode sebagai berikut:

- Rata-rata error koordinat X Bezier
  = 5,5635165 / 7 = 0,794788071
- Rata-rata error koordinat Y Bezier
  = 2,9417029 / 7 = 0,420243271
- Rata-rata error koordinat X Bresenham
  = 5,8321742 / 7 = 0,833167742
- Rata-rata error koordinat Y Bresenham
  = 3,1265817 / 7 = 0,446654528

Dari hasil perhitungan di atas dapat disimpulkan bahwa akurasi Bezier lebih baik daripada Bresenham yaitu untuk koordinat X terpaut 0,038379671 dan untuk koordinat Y terpaut 0,026411257.

5. KESIMPULAN

Dari hasil pengamatan dan uji coba yang dilakukan, dapat kesimpulan:

1. Pembuatan lingkaran dengan menggunakan algoritma Bresenham lebih cepat prosesnya sebesar 1.44 kali dibandingkan dengan metode Bezier .
2. Sedangkan ditinjau dari akurasi, lingkaran Bezier lebih baik dibandingkan lingkaran Bresenham dengan error koordinat X lebih kecil 0,038379671 dan error koordinat Y lebih kecil 0,026411257
3. Dengan kecepatan prosesor komputer seperti Pentium dengan aneka tipenya yang memiliki kecepatan eksekusi cukup tinggi maka dapat disimpulkan secara keseluruhan bahwa metode Bezier lebih baik dibandingkan algoritma Bresenham karena memiliki akurasi yang lebih baik.

DAFTAR PUSTAKA

1. Bresenham, Jack, *A Linear Algorithm for Incremental Digital Display of Circular Arcs*. Association for Computing Machinery, Inc.,1977.

2. Farin, Gerald, *Curves and Surfaces for CAGD*, 3rd ed. Arizona: Academic Press, 1988.

3. Hearn, Donald, and M. Pauline Baker, *Computer Graphics*, Prentice-Hall, 1986.

4. Piegl, L, *Interactive Data Interpolasi by Rational Bezier Curves*, IEEE. April 1987. p. 45-58.

5. Rogers, David F., *Mathematical Elements for Computer Graphic*, NewYork: McGraw Hill,1990.