

Laporan Bresenham
Mata Kuliah Grafika Komputer



Dibuat oleh :

Muhammad Kamaluddin Primajaya (20051397035)

2020A

Program Studi D4 Manajemen Informatika

Fakultas Vokasi

Universitas Negeri Surabaya

2022

Algoritma Bresenham

Algoritma Garis Bresenham adalah suatu garis yang menentukan titik-titik dalam dimensi yang membentuk dekat dengan garis lurus antara dua titik yang diberikan. Pendekatan yang biasa digunakan untuk menggambar garis pada layar komputer, karena hanya menggunakan penambahan bilangan bulat, pengurangan dan pergeseran. Algoritma ini merupakan salah satu algoritma paling awal yang dikembangkan di bidang komputer grafis.

Cara kerja Algoritma Bresenham adalah mengecek garis yang sudah diubah hanya dengan menggunakan metode perhitungan integer yang nantinya akan terus bertambah sehingga bisa menampilkan bentuk lingkaran dan bentuk kurva lainnya.

Langkah-langkah :

1. Masukkan 2 titik, kemudian simpan titik yang paling kiri sebagai nilai (X,Y)
2. Plotkan titik yang pertama.
3. Hitung Δx , Δy , $2\Delta y$ dan $2\Delta y - 2\Delta x$ dan dapatkan nilai awal parameter keputusan sbb: $p_0 = 2\Delta y - \Delta x$
4. Setiap X_k sepanjang garis, mulai dari $k=0$, lakukan langkah pengujian sbb : jika $p_k < 0$, maka titik selanjutnya yang akan diplot adalah (X_{k+1}, Y_k) , kemudian :
 $P_{k+1} = P_k + 2\Delta y$ jika sebaliknya, maka titik selanjutnya memiliki nilai (X_{k+1}, Y_{k+1}) , lalu perhitungannya : $p_{k+1} = p_k + 2\Delta y - p\Delta x$
5. Ulangi langkah ke-4 sebanyak Δx kali.

Source Code

```
1  # Muhammad Kamaluddin Primajaya
2  # 20051397035
3  # 2020A
4
5  from OpenGL.GL import *
6  from OpenGL.GLU import *
7  from OpenGL.GLUT import *
8
9  def BRESENHAM(x1,y1,x2,y2):
10     #menentukan delta X dan delta Y
11     x=x1
12     y=y1
13     deltaX = abs(x2-x1)
14     deltaY = abs(y2-y1)
15
16     #menghitung p, 2dx dan 2(dy-dx)
17     p = (2 * deltaY) - (deltaX)
18     duadx = 2 * deltaX
19     duaDyDx = 2 * (deltaY-deltaX)
20
21     #Menentukan titik awal dan titik akhir
22     if(x1>x2):
23         x = x2
24         y = y2
25         xend = x1
26
27     else:
28         x = x1
29         y = y1
30         xend = x2
```

```
31
32     #Memulai menggambar menggunakan BRESENHAM
33     #Membersihkan window
34     glClear(GL_COLOR_BUFFER_BIT)
35     #Menentukan warna
36     glColor3f(1.0,0.0,0.0)
37     #Spesifikasikan diameter dari pixel yang akan digambar
38     glPointSize(10.0)
39     #Memilih mode point
40     glBegin(GL_POINTS)
41
42     #Looping pada saat nilai x1 < x2
43     while x<xend:
44         #Menentukan titik yang akan diisi
45         x +=1
46
47         if(p<0):
48             p+=duadx
49         elif(y1>y2):
50             y-=1
51         else:
52             y+=1
53
54         p += duaDyDx
55
56         #Menggambar pixel
57         glVertex2i(x,y)
58
59     glEnd()
60     glFlush()
61
```

```

62 def main():
63     x1 = int(15)
64     y1 = int(10)
65     x2 = int(50)
66     y2 = int(35)
67
68     #inisialisasi glut
69     glutInit(sys.argv)
70     #inisialisasi tipe display glut
71     glutInitDisplayMode(GLUT_RGB)
72     #inisialisasi ukuran layar glut
73     glutInitWindowSize(500,500)
74     #inisialisasi posisi layar glut
75     glutInitWindowPosition(0,0)
76     #inisialisasi pembuatan window
77     glutCreateWindow("Menggambar garis BRESENHAM")
78     glutDisplayFunc(lambda: BRESENHAM(x1,y1,x2,y2))
79     glutIdleFunc(lambda: BRESENHAM(x1,y1,x2,y2))
80
81     #Memberikan layar dan memberikan warna
82     glClearColor(0.0,0.0,0.0,1.0)
83     #Set origin dari grid dan ukurannya 100 x 100
84     gluOrtho2D(0,100,0,100)
85     glutMainLoop()
86
87 main()

```

Output

