

**LAPORAN TUGAS BRESENHAM**  
**MATA KULIAH GRAFIKA KOMPUTER**



Oleh:

Hana Farahdiana ( 20051397073 )

2020-A

## 1. Algoritma Bresenham

- Dikembangkan oleh Bresenham
- Berdasarkan selisih antara garis yang diinginkan terhadap setengah ukuran dari pixel yang sedang digunakan.

untuk menentukan jumlah pixel yang membuat suatu garis menggunakan algoritma bresenham adalah sebagai berikut :

1. Masukkan 2 endpoints, simpan endpoints kiri sebagai  $(x_0, y_0)$  dan kanan  $(x_1, y_1)$ .
2. Hitung konstanta  $\Delta x$ ,  $\Delta y$ ,  $2\Delta y$ ,  $2\Delta x$ ,  $2\Delta y - 2\Delta x$  dan nilai awal parameter keputusan  $p_0 = 2\Delta y - \Delta x$ .
3. Pada setiap  $x_k$  di garis, dimulai dari  $k=0$ , ujilah : Jika  $p_k \leq 0$  maka plot  $(x_{k+1}, y_{k+1})$  dan  $p_{k+1} = p_k + 2\Delta y - 2\Delta x$ .
4. Ulangi tahap 3 sampai mencapai  $x, y$  yang dituju.

- Source Code

🔗 bresenham.py > ...

```
1  #Hana Farahdiana 20051397073 2020MIA
2  # Algoritma Bresenham
3
4  from OpenGL.GL import *
5  from OpenGL.GLUT import *
6  from OpenGL.GLU import *
7
8  # bresenham line drawing algorithm
9  def bresenham(x0, y0, x1, y1):
10     dx = x1 - x0
11     dy = y1 - y0
12     xsign = 1 if dx > 0 else -1
13     ysign = 1 if dy > 0 else -1
14     dx = abs(dx)
15     dy = abs(dy)
16     if dx > dy:
17         xx, xy, yx, yy = xsign, 0, 0, ysign
18     else:
19         dx, dy = dy, dx
20         xx, xy, yx, yy = 0, ysign, xsign, 0
21     D = 2*dy - dx
22     y = 0
23     for x in range(dx+1):
24         glVertex2i(x0 + x*xx + y*yx, y0 + x*xy + y*yy)
25         if D > 0:
26             y += 1
27             D -= 2*dx
28         D += 2*dy
```

```
bresenham.py > ...
30 def display():
31     glClear(GL_COLOR_BUFFER_BIT)
32     glColor3f(1.0, 1.0, 1.0)
33     glBegin(GL_POINTS)
34     bresenham(10,20,30,40)
35     glEnd()
36     glFlush()
37
38 def main():
39     glutInit(sys.argv)
40     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
41     glutInitWindowSize(500, 500)
42     glutInitWindowPosition(100, 100)
43     glutCreateWindow(b"Bresenham's line algorithm")
44     glClearColor(0.0, 0.0, 0.0, 0.0)
45     gluOrtho2D(-50, 50, -50, 50)
46     glutDisplayFunc(display)
47     glutMainLoop()
48
49 main()
```

- **Output**

