

## Membuat Garis Menggunakan Algoritma DDA Bresenham

### Cara Kerja Algoritma

1. Tentukan dua titik yang akan dihubungkan dalam pembentukan garis.
2. Tentukan salah satu sebagai titik awal ( $x_0, y_0$ ) dan titik akhir ( $x_1, y_1$ ).
3. Hitung  $dx$ ,  $dy$ ,  $2dy$  dan  $2dy - 2dx$
4. Hitung parameter :  $p_0 = 2dy - dx$
5. Untuk setiap  $x_k$  sepanjang jalur garis, dimulai dengan  $k=0$  bila  $p_k < 0$  maka titik selanjutnya adalah: ( $x_{k+1}, y_k$ ) dan  $p_{k+1} = p_k + 2dy$  bila tidak, titik selanjutnya adalah: ( $x_{k+1}, y_{k+1}$ ) dan  $p_{k+1} = p_k + 2dy - 2dx$
6. Ulangi nomor 5 untuk menentukan posisi pixel berikutnya, sampai  $x = x_1$  atau  $y = y_1$ .

### Langkah Pengerjaan

1. Import Library

```
import sys
import time
from OpenGL import *
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
```

2. Definisikan ukuran lebar dan Panjang jendela serta jumlah jendela

```
window = 0

width, height = 500, 500
```

3. Buat Fungsi Algoritma DDA

```
def beginDda(x1, y1, x2, y2):

    dx = abs(x2 - x1)
    dy = abs(y2 - y1)
    slope = dy/float(dx)

    x, y = x1, y1

    if slope > 1:
        dx, dy = dy, dx
```

```

        x, y = y, x
        x1, y1 = y1, x1
        x2, y2 = y2, x2

    p = 2 * dy - dx

    glVertex2f(x, y)

    for k in range(2, dx):
        if p > 0:
            y = y + 1 if y < y2 else y - 1
            p = p + 2*(dy - dx)
        else:
            p = p + 2*dy

        x = x + 1 if x < x2 else x - 1

    time.sleep(0.01)
    glVertex2f(x, y)

```

4. Buat Fungsi untuk menggambar garis dari nilai-nilai pengembalian fungsi Algoritma

```

def lineBres():
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT)
    glBegin(GL_LINES)
    glColor(1.0, 1.0, 0.0)

    beginDda(50, 50, 350, 350)

    glEnd()
    glutSwapBuffers()

```

5. Buat fungsi utama untuk dijalankan

```

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_RGBA|GLUT_DOUBLE|GLUT_ALPHA|GLUT_DEPTH)
    glutInitWindowSize(width, height)
    glutInitWindowPosition(0,0)
    glutCreateWindow(b'Bresenham DDA Algorithm')
    glClearColor(0.0, 0.0, 0.0, 0.0)
    gluOrtho2D(0.0, 500.0, 0.0, 400.0)
    glutDisplayFunc(lineBres)

```

```
glutIdleFunc(lineBres)  
glutMainLoop()
```

## 6. Jalankan fungsi utama

```
main()
```

## Hasil

