

Tugas Breseham
Mata Kuliah Grafika Komputer



Oleh :
Isa Iman Muhammad
NIM. 20051397019
2020A D4 Manajemen Informatika

FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA
2022

Algoritma Bresenham

- **Pengertian**

Algoritma bresenham merupakan suatu algoritma (pendekatan) yang dikreasikan oleh bresenham yang tidak kalah akurat dan efisien dengan algoritma primitif lainnya (seperti DDA). Bagian pengkonversian (scan-knversi) garis akan melakukan kalkulasi untuk penambahan nilai-nilai integer (yang dibutuhkan untuk membentuk garis) yang disesuaikan dengan tipe grafik yang dipakai oleh layar komputer (keadaan monitor pc) kita. Untuk mengilustrasikan pendekatan bresenham, pertama kita harus memperhatikan proses scan- konvensi untuk garis dengan slope positif yang lebih kecil dari 1. Posisi pixel sepanjang line-path kemudian ditentukan dengan penyamplingan pada unit interval x.dimulai dari endpoint kiri (X_0, Y_0) dari garis yang diberikan, kita pindahkan beberapa kolom berturut-turut (berdasarkan posisi x) dan plot pixel-pixel yang mempunyai nilai scan-line y ke jarak yang paling dekat dengan line-path.

- **Cara Kerja**

Cara kerja dari algoritma ini adalah memeriksa garis yang telah diubah hanya dengan menggunakan perhitungan integer yang terus bertambah yang bisa diadaptasikan untuk menampilkan lingkaran dan bentuk kurva yang lain.

Langkah-langkah dalam algoritma Bresenham adalah:

1. Input dua titik, dan simpan titik yang paling kiri sebagai (x_0, y_0)
2. Plotkan titik pertama tersebut
3. Hitunglah Δx , Δy , $2\Delta y$ dan $2\Delta y - 2\Delta x$ serta perolehlah nilai awal parameter keputusan sbb: $p_0 = 2\Delta y - \Delta x$
4. Setiap x_k sepanjang garis, mulai dari $k = 0$, lakukan pengujian sbb:
Apabila $p_k < 0$, maka titik berikutnya yang akan diplot adalah (x_{k+1}, y_k) ,
kemudian : $p_{k+1} = p_k + 2\Delta y$
5. Apabila Sebaliknya, maka titik berikutnya bernilai (x_{k+1}, y_{k+1}) , lalu
perhitungannya: $p_{k+1} = p_k + 2\Delta y - 2\Delta x$
6. Ulangi langkah 4 sebanyak Δx kali.

- Source Code

```
# Created By :

# Nama : Isa Iman Muhammad
# NIM : 20051397019
# Kelas : 2020A D4 Manajemen Informatika

from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *

def init():
    glClearColor(0.0, 0.0, 0.0, 0.0)
    gluOrtho2D(-100.0, 100.0, -100.0, 100.0)
    glPointSize(10)

def plot(x, y):
    glBegin(GL_POINTS)
    glVertex2f(x, y)
    glEnd()

def bresenham_circle_drawing(r):

    # tempat lingkaran hasil output
    x_position = 15
    y_position = -15

    x = 0
    y = r

    # parameter keputusan
    d = 3 - 2 * r

    # membuat titik koordinat
    plot(x + x_position, y + y_position)

    while y > x:

        if d < 0:
            x += 1
            d += 4 * x + 6
        else:
```

```

    x += 1
    y -= 1
    d += (4 * (x - y)) + 6

    # mencari nilai (x, y)
    # membalikkan nilai menjadi (y, x)

    #nilai (x, y)

    # kuadran 1
    plot(x + x_position, y + y_position)

    # kuadran 2
    plot(x + x_position, -y + y_position)

    # kuadran 3
    plot(-x + x_position, -y + y_position)

    # kuadran 4
    plot(-x + x_position, y + y_position)

    #nilai (y, x)

    # kuadran 1
    plot(y + x_position, x + y_position)

    # kuadran 2
    plot(-y + x_position, x + y_position)

    # kuadran 3
    plot(-y + x_position, -x + y_position)

    # kuadran 4
    plot(y + x_position, -x + y_position)

```

```

def plotpoints():
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1, 1.0, 1.0)
    glBegin(GL_LINES)
    glVertex2f(-100, 0)
    glVertex2f(100, 0)
    glVertex2f(0, -100)
    glVertex2f(0, 100)
    glEnd()

```

```
    bresenham_circle_drawing(40)

    glFlush()

# menampilkan hasil output

def main():
    # menampilkan hasil output
    glutInit(sys.argv)
    # inisialisasi tipe display glut
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    # inisialisasi ukuran layar glut
    glutInitWindowSize(400, 400)
    # inisialisasi posisi layar glut
    glutInitWindowPosition(200, 200)
    # inisialisasi pembuatan window
    glutCreateWindow("Bresenham Lingkaran")
    glutDisplayFunc(plotpoints)

    init()
    glutMainLoop()

main()
```

- **Hasil Output**

