

Nama : Rizka Nurul Septiani Hakim

NIM : 20051397026

Kelas : 2020B

Prodi : D4 Manajemen Informatika

Tugas Membuat Algoritma Bresenham - Praktikum Ke-11 Grafika Komputer

Cara Kerja Singkat Algoritma Bresenham :

Salah satu dari algoritma menggambar garis dalam grafika komputer yaitu Algoritma Bresenham, dikembangkan oleh Jack Bresenham pada tahun 1962 di IBM. Bresenham pada tahun 1965, melakukan perbaikan dari algoritma perhitungan koordinat piksel yang menggunakan persamaan $y = mx + c$, dengan cara menggantikan operasi bilangan riil perkalian dengan operasi penjumlahan, yang kemudian dikenal dengan Algoritma Bresenham. Pada algoritma bresenham, nilai y kedua dan seterusnya, dihitung dari nilai y sebelumnya, sehingga hanya titik y pertama yang perlu dilakukan operasi secara lengkap. Perbaikan algoritma ini ternyata tidak menghasilkan perbaikan yang cukup signifikan. Perbaikan berikutnya dilakukan dengan cara menghilangkan operasi bilangan riil dengan operasi bilangan integer. Operasi bilangan integer jauh lebih cepat dibandingkan dengan operasi bilangan riil, terutama pada penambahan dan pengurangan.

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 : $p_{k+1} = p_k + 2\Delta y$

Apabila sebaliknya, maka titik berikutnya bernilai (x_{k+1}, y_{k+1}) , lalu perhitungannya: $p_{k+1} = p_k + 2\Delta y - 2\Delta x$

5. Ulangi langkah 4 sebanyak Δx kali.

```
#memanggil OpenGL untuk menampilkan gambar hasil perhitungan algoritma bresenham
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

def initFun():
    #membersihkan background layar dan memberikan warna
    glClearColor(1.0,1.0,1.0,0.0)
    #menentukan warna garis
    glColor3f(0.0,1.0, 0.0)
    #spesifikasikan diameter dari pixel yang akan digambar
    glPointSize(5.0)
    #merubah status OpenGL ke mode proyeksi (transformasi)
    glMatrixMode(GL_PROJECTION)
```

```

#memanggil matriks identitas dan mereset project matrix sebelumnya
glLoadIdentity()
#set origin dari grid dan ukurannya
gluOrtho2D(0.0,640.0,0.0,480.0)

def AlgBresenham():
    #tentukan titik awal dan akhir
    x1 = 20
    y1 = 20
    x2 = 500
    y2 = 400
    x = x1
    y = y1

    #hitung dx dan dy
    dx = abs(x2 - x1)
    dy = abs(y2 - y1)

    #hitung p (parameter)
    p = 2 * dy - dx
    duady = 2 * dy
    duadydx = 2 * (dy - dx)

    #menentukan titik awal dan akhir
    if (x1 > x2):
        x = x2
        y = y2
        xend = x1
    else:
        x = x1
        y = y1
        xend = x2

    #memilih mode point
    glBegin(GL_POINTS)
    #menentukan titik-titik
    glVertex2i(x, y)

    #perulangan untuk menggambar titik-titik (looping)
    while (x < xend):
        x = x+1
        if (p < 0):
            p += duady
        else:
            if (y1 > y2):
                y = y-1
            else:
                y = y+1
            p += duadydx
        glVertex2i(x, y)

    glEnd()
    glFlush()

```

```
if __name__ == '__main__':  
    glutInit()  
    #inisialisasi pembuatan window  
    glutInitWindowSize(700,500)  
    #menampilkan tulisan ke window  
    glutCreateWindow("Tampilan Hasil Algoritma Bresenham")  
    #inisialisasi modus display untuk memanggil fungsi  
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA)  
    glutDisplayFunc(AlgBresenham)  
    initFun()  
    #sintaks untuk me-looping atau mengulang fungsi/method main  
    glutMainLoop()
```

Output :

