Nama : Himmatuz Zahiroh

Nim : 20051397070

Kelas : Manajemen Informatika 2020B

PENJELASAN ALGORITMA BRESENHAM

ATURAN BRESENHAM :

-   Jika Pk bernilai positif (+), maka tambahkan hasilnya dengan B dan nilai x dan y ditambah 1.

- Jika Pk bernilai negatif (-), maka tambahkan hasilnya dengan A dan nilai x ditambah 1, sedangkan y ditambah 0 (tetap).

-  Putaran dihentikan jika koordinat x dan y sudah mencapai batas akhir.

PRINSIP DARI ALGORITMA BRESENHAM

~ Sumbu vertikal memperlihatkan posisi scan line.

~  Sumbu horizontal memperlihatkan kolom pixel.

~ Pada tiap langkah, penentuan pixel selanjutnya didasari oleh parameter integer yang nilainya proporsional dengan pengurangan antara vertical separations dari dua posisi piksel dari nilai actual.

Software yang diperlukan :  
IDLE Pyhton 3.10.2 (IDLE)

Adapun penjelasan Source Code nya sebagai berikut :

|  |
| --- |
| From OpenGL.GL import \*  from OpenGL.GLU import \*  from OpenGL.GLUT import \*  def init():      glClearColor(0.0, 0.0, 0.0, 0.0)      gluOrtho2D(-50.0, 50.0, -50.0, 50.0)      glPointSize(5)  def plot(x, y):      glBegin(GL\_POINTS)      glVertex2f(x, y)      glEnd()  def bresenham\_drawing\_circle(r):    **# contoh soal jika nilai center circle (3,3) dan r = 9**      x\_center = 3      y\_center = 3      r = 9      x = 0      y = r  **# parameter keputusan bresenham**      d = 3 - 2 \* r      # membuat titik pada koordinat yang ditentukan      plot(x + x\_center, y + y\_center)      while y > x:          if d < 0:              x += 1              d += 4 \* x + 6          else:              x += 1              y -= 1              d += (4 \* (x - y)) + 10  **# Jika x <= y sudah terpenuhi, maka perulangan akan di hentikan.**  **# jika nilai (x,y)sudah didapatkan, lalu membalikan nilai (y, x)**  **# Untuk pixel (x, y)**  **# Quadrant 1**          plot(x + x\_center, y + y\_center)  **# Quadrant 2**          plot(x + x\_center, -y + y\_center)  **# Quadrant 3**          plot(-x + x\_center, -y + y\_center)  **# Quadrant 4**          plot(-x + x\_center, y + y\_center)  **# Untuk pixel (y, x)**  **Quadrant 1)**          plot(y + x\_center, x + y\_center)  **# Quadrant 2**          plot(-y + x\_center, x + y\_center)  **# Quadrant 3**          plot(-y + x\_center, -x + y\_center)  **# Quadrant 4**          plot(y + x\_center, -x + y\_center)  def plotpoints():      glClear(GL\_COLOR\_BUFFER\_BIT)      glColor3f(1.0, 0.0, 1.0)      glBegin(GL\_LINES)      glVertex2f(-50, 0)      glVertex2f(50, 0)      glVertex2f(0, -50)      glVertex2f(0, 50)      glEnd()      bresenham\_drawing\_circle(40)      glFlush()  def main():      glutInit(sys.argv)      glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB)      glutInitWindowSize(500, 500)      glutInitWindowPosition(100, 100)      glutCreateWindow("Bresenham Cricle TUGAS 4 Kel 2 MM3")      glutDisplayFunc(plotpoints)      init()      glutMainLoop()  main() |