

LAPORAN PRAKTIKUM ALGORITMA BRESENHEM
GRAFIKA KOMPUTER



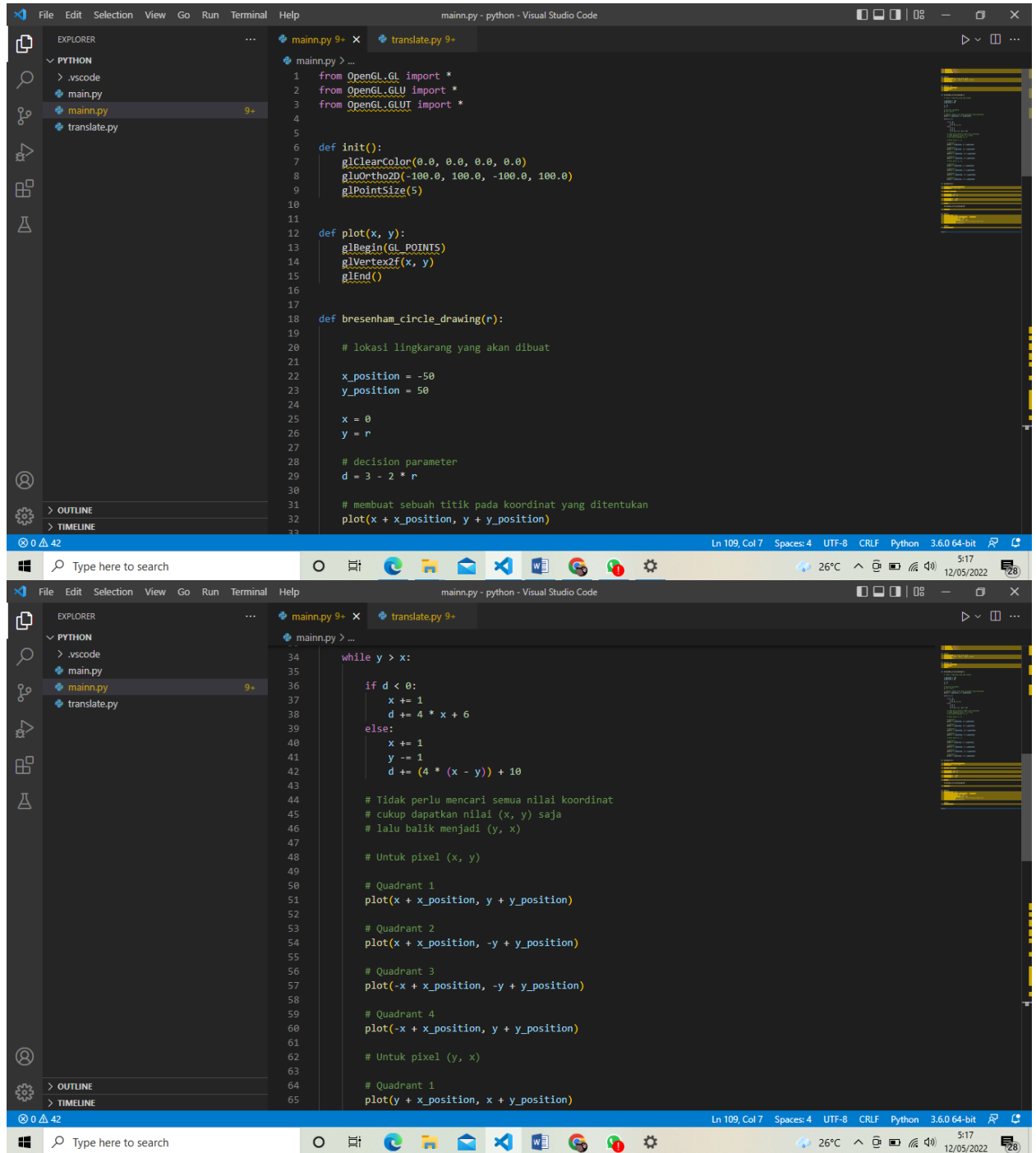
OLEH :

2020A

SAFITRI DIWANTI LESTARI (20051397039)

PROGRAM STUDI MANAJEMEN INFORMATIKA
FAKULTAS VOKASI
UNIVERSITAS NEGERI SURABAYA

1. Source Code



The image displays two screenshots of a Visual Studio Code editor window, showing the source code for a Python program that implements a circle-drawing algorithm using Bresenham's method. The editor is configured with a dark theme and the Python 3.6.0 64-bit interpreter.

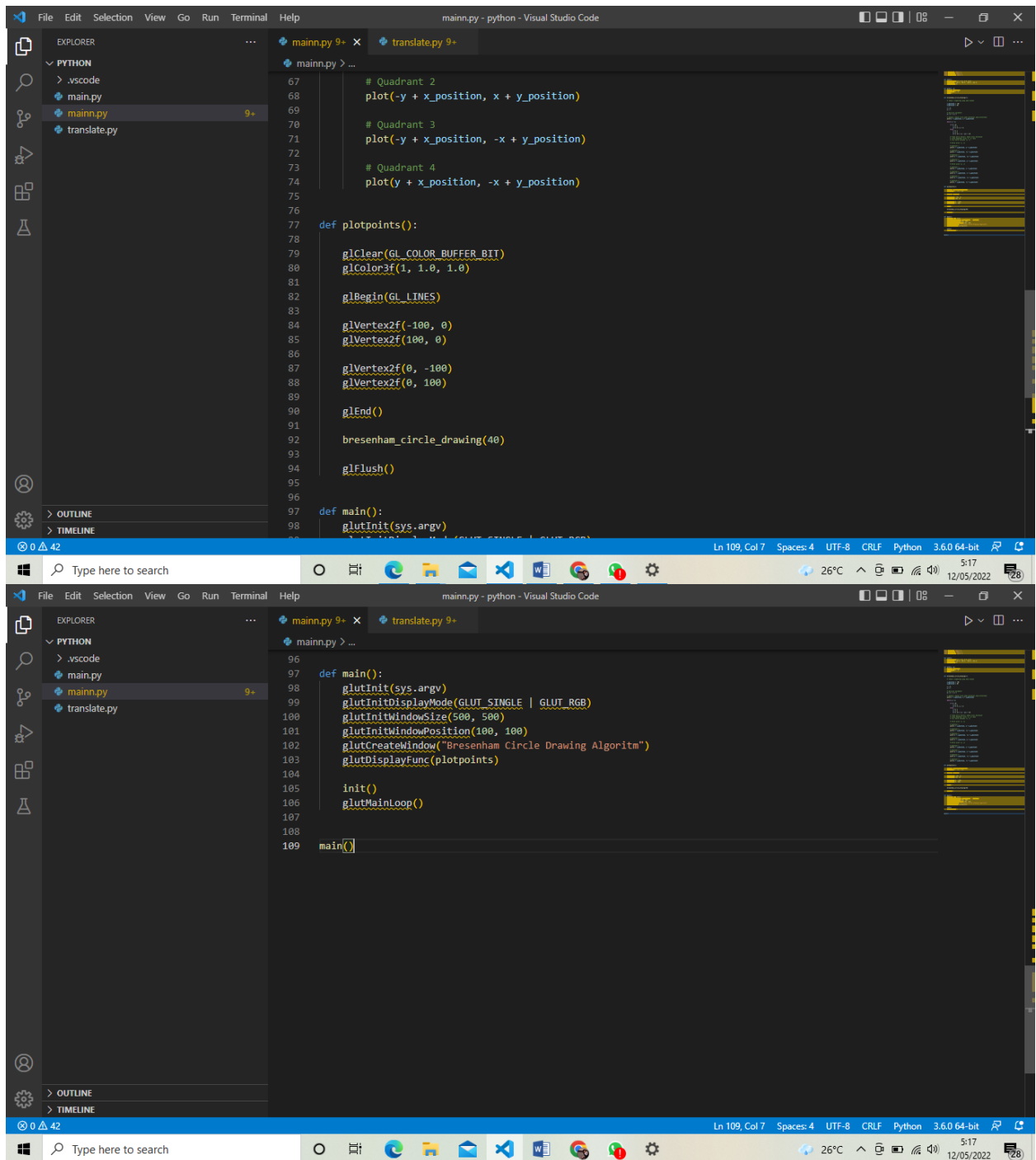
Top Screenshot: The code defines the initial setup and the first part of the circle-drawing function.

```
1 from OpenGL.GL import *
2 from OpenGL.GLU import *
3 from OpenGL.GLUT import *
4
5
6 def init():
7     glClearColor(0.0, 0.0, 0.0, 0.0)
8     gluOrtho2D(-100.0, 100.0, -100.0, 100.0)
9     glPointSize(5)
10
11
12 def plot(x, y):
13     glBegin(GL_POINTS)
14     glVertex2f(x, y)
15     glEnd()
16
17
18 def bresenham_circle_drawing(r):
19
20     # lokasi lingkaran yang akan dibuat
21
22     x_position = -50
23     y_position = 50
24
25     x = 0
26     y = r
27
28     # decision parameter
29     d = 3 - 2 * r
30
31     # membuat sebuah titik pada koordinat yang ditentukan
32     plot(x + x_position, y + y_position)
33
```

Bottom Screenshot: The code continues with the main loop of the circle-drawing function, calculating the next pixel coordinates and plotting them in all four quadrants.

```
34 while y > x:
35
36     if d < 0:
37         x += 1
38         d += 4 * x + 6
39     else:
40         x += 1
41         y -= 1
42         d += (4 * (x - y)) + 10
43
44     # Tidak perlu mencari semua nilai koordinat
45     # cukup dapatkan nilai (x, y) saja
46     # lalu balik menjadi (y, x)
47
48     # Untuk pixel (x, y)
49
50     # Quadrant 1
51     plot(x + x_position, y + y_position)
52
53     # Quadrant 2
54     plot(x + x_position, -y + y_position)
55
56     # Quadrant 3
57     plot(-x + x_position, -y + y_position)
58
59     # Quadrant 4
60     plot(-x + x_position, y + y_position)
61
62     # Untuk pixel (y, x)
63
64     # Quadrant 1
65     plot(y + x_position, x + y_position)

```



2. Analisa Program

Algoritma bresenham untuk membuat lingkaran, hal terpenting adalah menentukan nilai radiusnya. Dari situlah kita dapat menghitung posisi untuk koordinat x dan y nya.

Berdasarkan rumus di atas, jika kita memberikan nilai 5 pada radius, maka akan didapat 5 juga pada y dan -7 pada d-nya. 'D' di sini merupakan decision parameter, berguna untuk

menentukan apakah nilai dari koordinat x dan y itu bertambah atau berkurang. Setiap koordinat x dan y di sini mewakili satu buah pixel.

Bailah, sekarang mari kita buat lingkaran dengan menggunakan algoritma bresenham secara manual terlebih dahulu dengan nilai radius adalah 5. Maka akan didapat data seperti di bawah ini.

Perulangan	x	y	d (Decision Parameter)	d < 0 ?
0	0	5	$d_0 = 3 - (2 * 5) = 3 - 10 = -7$	T
1	1	5	$d_1 = -7 + (4 * 1) + 6 = 3$	F
2	2	4	$d_2 = 3 + (4 * (2 - 4)) + 10 = 5$	F
3	3	3	$d_3 = 5 + (4 * (3 - 3)) + 10 = 15$	F

Untuk mendapatkan nilai x, y dan d kita hanya perlu mengikuti rumus di atas. Jika seandainya nilai y tidak lebih besar dari x, maka kita hentikan perulangannya. Pada kasus ini, nilai 3 tidak lebih besar dari 3, sehingga perulangan cukup sampai di situ. Cara untuk mendapatkan koordinat yang lain jika kita berhenti di (3,3) yaitu dengan membalik posisi x dan y nya saja. Jika kita membalik posisi x dan y pada perulangan ke dua, maka akan didapat (4, 2). Dengan begitu, kita tidak harus menghabiskan waktu untuk menghitung semuanya satu-persatu.

Ini merupakan koordinat pada semua kuadran dengan nilai radius 5. Sekali lagi, untuk mendapatkan nilai koordinat setelah (3, 3) kita hanya perlu membalik nilai x dan y nya saja. Untuk nilai negatif, kita hanya perlu menyesuaikan dengan kuadran nya.

3. Output Program

Output program belum bisa muncul dikarenakan penginstalan modul OpenGL yang masih gagal.