

# **LAPORAN PRAKTIKUM**

## **GRAFIKA KOMPUTER**

Dosen pengampu: Rio Priantama, S.T., M.T.I

### **MODUL 3**



Nama : Rio Andika Andriansyah

NIM : 20230810155

Kelas : TINFC-2023-04

**TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS KUNINGAN**

# PRAKTIKUM

## Praktikum 1

The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar lists notebooks under 'PRAKTIKUM GRAFIKA KOMPUTER' and 'Modul 3'. The current notebook is 'Praktikum3.ipynb'. The code cell contains Python code for the DDA algorithm:

```
import matplotlib.pyplot as plt
import numpy as np

def garis_dda(x0, y0, x1, y1):

    dx = x1 - x0
    dy = y1 - y0

    step = max(abs(dx), abs(dy))

    XInc = dx / step
    YInc = dy / step

    x = x0
    y = y0

    x_points = []
    y_points = []

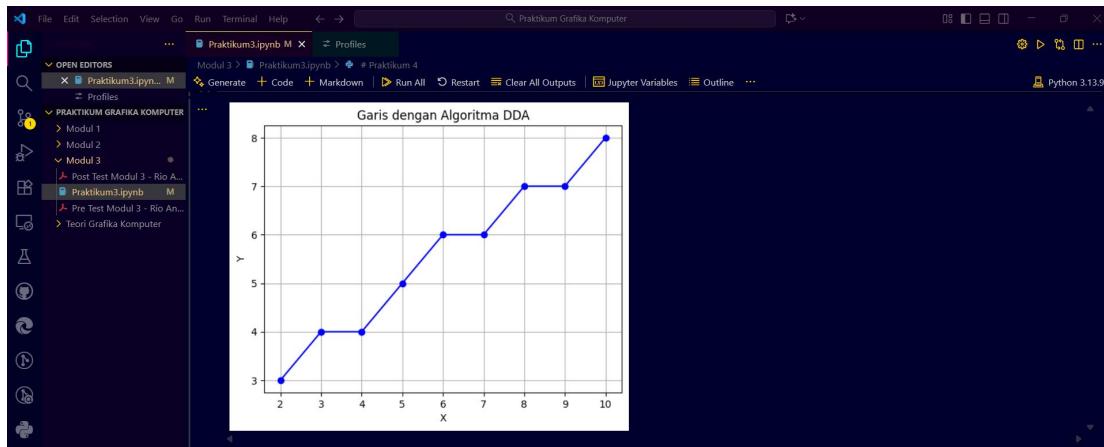
    for i in range(int(step) + 1):
        x_points.append(int(round(x)))
        y_points.append(int(round(y)))

        x += XInc
        y += YInc

    plt.plot(x_points, y_points, marker='o', color='blue')
    plt.title('Garis dengan Algoritma DDA')
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.grid(True)
    plt.show()

garis_dda(2, 3, 10, 8)
```

The output cell shows a plot titled "Garis dengan Algoritma DDA" with a blue line connecting points (2, 3), (3, 4), (4, 4), (5, 5), (6, 6), (7, 6), (8, 7), (9, 7), and (10, 8).



## Praktikum 2

The screenshot shows a Jupyter Notebook interface with a dark theme. The left sidebar lists notebooks under 'PRAKTIKUM GRAFIKA KOMPUTER' and 'Modul 3'. The current notebook is 'Praktikum3.ipynb'. The code cell contains Python code for the DDA algorithm:

```
# Praktikum 2

import matplotlib.pyplot as plt
import numpy as np

def dda_algorithm(x0, y0, x1, y1):

    dx = x1 - x0
    dy = y1 - y0

    step = max(abs(dx), abs(dy))

    XInc = dx / float(step)
    YInc = dy / float(step)

    x = x0
    y = y0

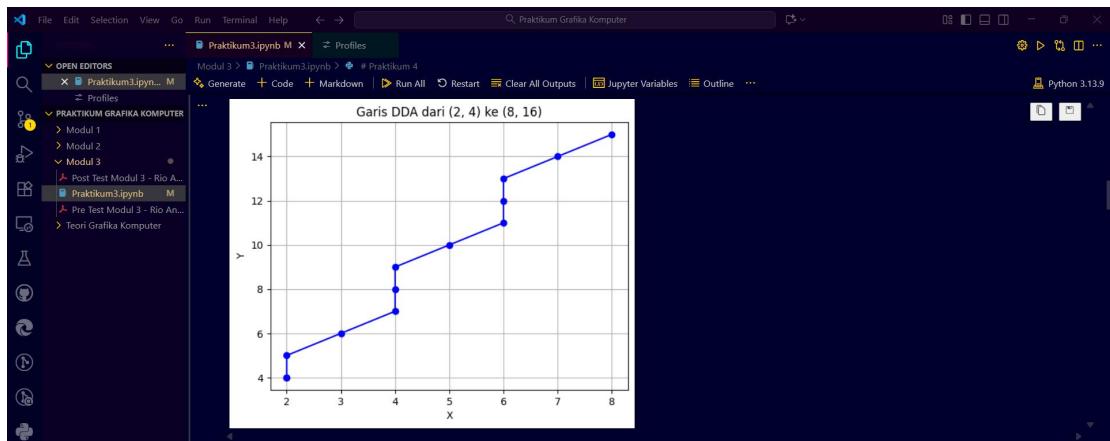
    x_points = [x]
    y_points = [y]

    for i in range(int(step)):
        x_points.append(int(round(x)))
        y_points.append(int(round(y)))

        x += XInc
        y += YInc

    plt.plot(x_points, y_points, marker='o', color='blue')
    plt.title("Garis DDA dari ((x0, y0)) ke ((x1, y1))")
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.grid(True)
    plt.show()

dda_algorithm(2, 3, 10, 8)
```



## Praktikum 3

```
# Praktikum 3

import matplotlib.pyplot as plt

# Fungsi untuk menggambar garis menggunakan algoritma Bresenham
def bresenham_algorithm(x0, y0, x1, y1):
    # Inisialisasi variabel
    x, y = x0, y0

    # Hitung perubahan absolut (delta)
    dx = abs(x1 - x0)
    dy = abs(y1 - y0)

    # Tentukan arah Langkah (sign)
    # Jika x1 > x0, sx = 1, jika tidak sx = -1
    sx = 1 if x0 < x1 else -1 # penambahan untuk x
    # Jika y1 > y0, sy = 1, jika tidak sy = -1
    sy = 1 if y0 < y1 else -1 # penambahan untuk y

    # Inisialisasi error (parameter keputusan)
    err = dx - dy

    # Daftar untuk menyimpan koordinat titik-titik
    x_points = [x]
    y_points = [y]

    # Looping selama titik saat ini bukan titik akhir
    while x != x1 or y != y1:
        # Hitung 2 * error
        e2 = 2 * err

        # Perbarui y jika e2 >= -dy
        if e2 > dy:
            err -= dy
            x += sx

        # Perbarui y jika e2 >= -dx
        if e2 < dx:
            err += dx
            y += sy

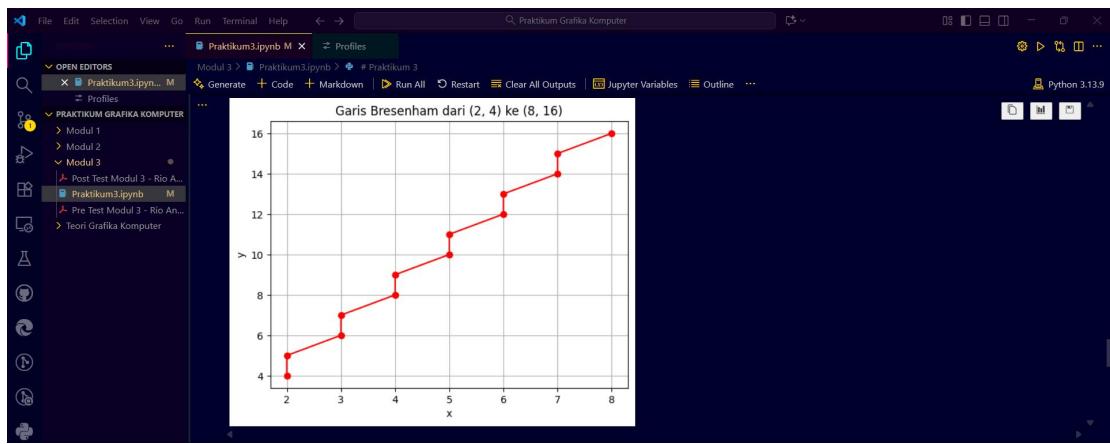
        # Tambahkan titik baru ke daftar
        x_points.append(x)
        y_points.append(y)

    # Menggambar garis menggunakan matplotlib
    plt.plot(x_points, y_points, marker="o", color="r")
    plt.title("Garis Bresenham dari ({}, {}) ke ({}, {})".format(x0, y0, x1, y1))
    plt.xlabel("x")
    plt.ylabel("y")
    plt.grid(True)
    plt.show()
```

```
# Meminta input dari pengguna untuk titik awal dan titik akhir
try:
    x0 = int(input("Masukkan koordinat x0 (titik awal): "))
    y0 = int(input("Masukkan koordinat y0 (titik awal): "))
    x1 = int(input("Masukkan koordinat x1 (titik akhir): "))
    y1 = int(input("Masukkan koordinat y1 (titik akhir): "))

    # Memanggil fungsi untuk menggambar garis dengan algoritma Bresenham
    bresenham_algorithm(x0, y0, x1, y1)

except ValueError:
    print("Input harus berupa bilangan bulat.")
```



## Praktikum 4

```
# Fungsi untuk menggambar garis menggunakan algoritma DDA
def dda_algorithm(x0, y0, x1, y1):
    # Hitung perubahan delta x dan delta y
    dx = x1 - x0
    dy = y1 - y0

    # Tentukan jumlah langkah (steps)
    # steps adalah nilai maksimum dari abs(dx) atau abs(dy)
    steps = abs(dx) if abs(dx) > abs(dy) else abs(dy)

    # Hitung inkrement (penambahan) untuk x dan y
    x_inc = dx / steps
    y_inc = dy / steps

    # Inisialisasi daftar titik
    x_points = [x0]
    y_points = [y0]

    # Inisialisasi koordinat saat ini
    x = x0
    y = y0

    # Looping untuk menentukan semua titik di sepanjang garis
    # Dimulai dari 1 karena titik awal (x0, y0) sudah ditambahkan
    for _ in range(int(steps)):
        x += x_inc
        y += y_inc

        # Tambahkan koordinat yang sudah dibulatkan
        x_points.append(round(x))
        y_points.append(round(y))

# Menggambar garis
plt.plot(x_points, y_points, marker="o", color="b")
```

```
# Fungsi untuk menggambar rumah
def draw_house():
    # Mengatur ukuran figure
    plt.figure(figsize=(6, 6))

    # Binding rumah (persegi)
    # Sisi Kiri
    dda_algorithm(2, 2, 6)
    # Sisi Atas
    dda_algorithm(2, 6, 6, 6)
    # Sisi Kanan
    dda_algorithm(6, 6, 6, 2)
    # Sisi Bawah
    dda_algorithm(6, 2, 2, 2)

    # Atap rumah (segitiga)
    # Kiri ke Puncak
    dda_algorithm(2, 6, 4, 8)
    # Puncak ke Kanan
    dda_algorithm(4, 8, 6, 6)

    # Pintu (sejajar kecil)
    # Sisi Kiri
    dda_algorithm(3, 2, 3, 4)
    # Sisi Atas
    dda_algorithm(3, 4, 4, 4)
    # Sisi Kanan
    dda_algorithm(4, 4, 4, 2)
```

```

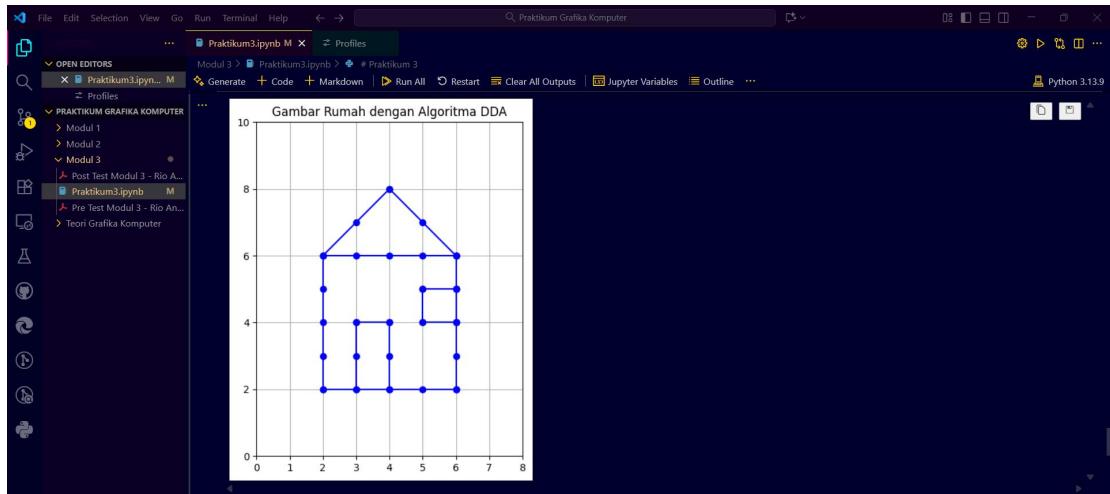
# Jendela (persegi kecil)
# Sisi Kiri
dda_algorithm(4, 5, 5)
# Sisi Atas
dda_algorithm(5, 5, 6, 5)
# Sisi Kanan
dda_algorithm(6, 5, 6, 4)
# Sisi Bawah
dda_algorithm(6, 4, 5, 4)

# Menampilkan hasil gambar rumah
plt.title("Gambar Rumah dengan Algoritma DDA")
plt.xlim(0, 8) # Batas sumbu X
plt.ylim(0, 10) # Batas sumbu Y

# Mengatur aspek rasio agar sumbu x dan y memiliki skala yang sama
plt.gca().set_aspect('equal', adjustable='box')
plt.grid(True)
plt.show()

# Memanggil fungsi untuk menggambar rumah
draw_house()

```



## TUGAS

- Buatlah kode program sederhana untuk menggambar rumah seperti gambar berikut:

