LAPORAN PRAKTIKUM

GRAFIKA KOMPUTER

(DOSEN PENGAMPU : Rio Priantama, S.T., M.T.I)

Modul 3

DISUSUN OLEH :

NAMA : MOHAMAD ABAN SY'BANA

NIM : 20230810012

KELAS : TINFC-2023-04

TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS KUNINGAN

2025

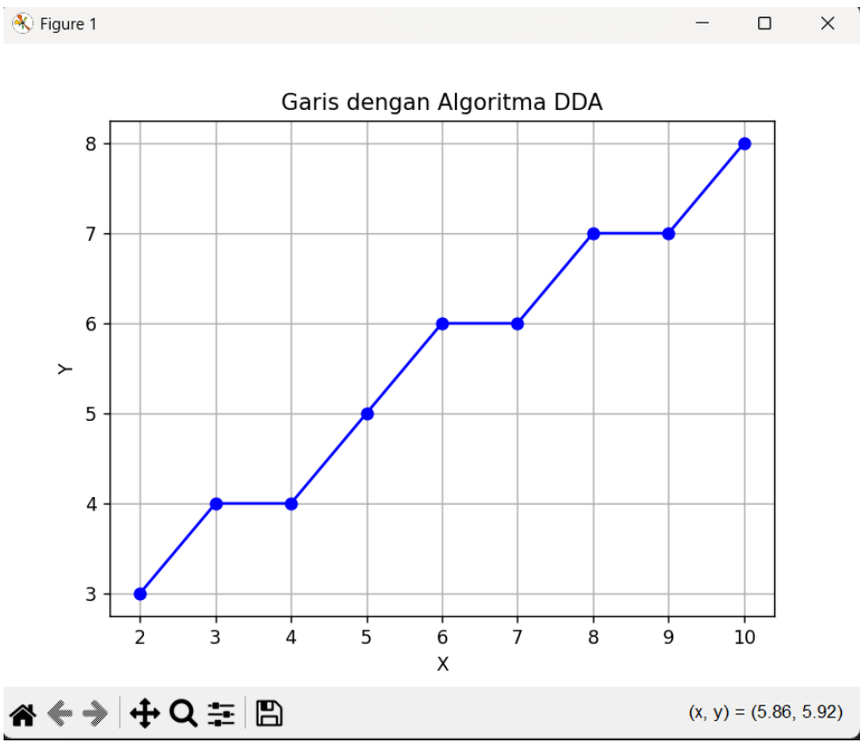
PRAKTIKUM

Praktikum 1: Implementasi algoritma DDA

Code Program

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

# Fungsi untuk menggambar garis menggunakan Algoritma DDA
def garis_dda(x0, y0, x1, y1):
    # Menghitung dx dan dy
    dx = x1 - x0
    dy = y1 - y0

    # Menentukan jumlah step
    step = max(abs(dx), abs(dy))

    # Menghitung penambahan koordinat Xinc dan Yinc
    Xinc = dx / step
    Yinc = dy / step

    # Inisialisasi titik awal
    x = x0
    y = y0

    # Simpan koordinat hasil DDA
    x_points = []
    y_points = []

    # Loop untuk menghitung setiap titik dari garis
    for i in range(int(step) + 1):
        # Menyimpan koordinat dengan pembulatan menjadi integer
        x_points.append(int(round(x)))
        y_points.append(int(round(y)))

        # Menambahkan Xinc dan Yinc untuk mendapatkan titik berikutnya
        x += Xinc
        y += Yinc

    # Gambar garis menggunakan matplotlib
    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()

# Memanggil fungsi untuk menggambar garis dengan titik awal (2, 3) dan titik akhir (10, 8)
garis_dda(2, 3, 10, 8)
```

Hasil RUN



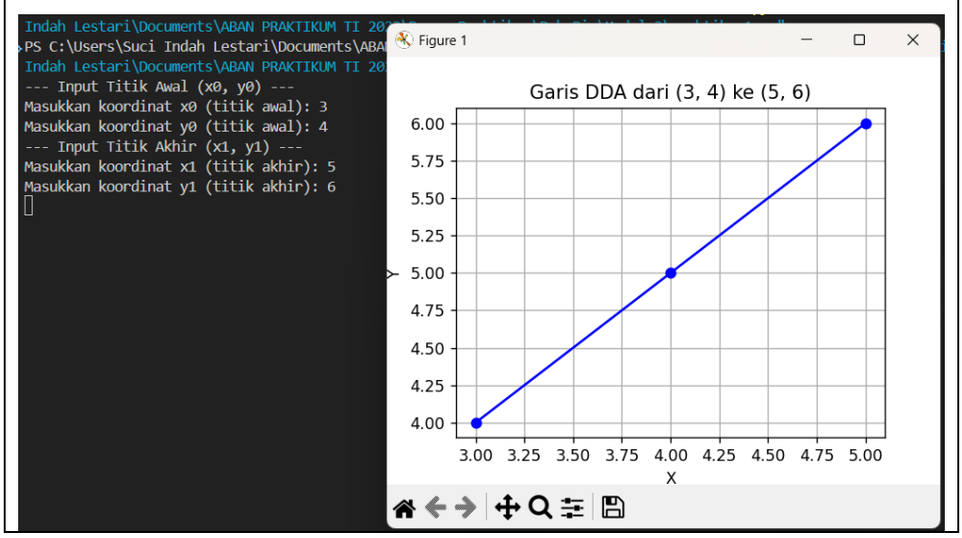
Praktikum 2: Implementasi algoritma DDA dengan Input Dinamis

Code Program

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

# Fungsi untuk menggambar garis menggunakan algoritma DDA
def dda_algorithm(x0, y0, x1, y1):
    # Menghitung perubahan koordinat
    dx = x1 - x0
    dy = y1 - y0

    # Menentukan jumlah langkah berdasarkan jarak terpanjang (step)
    steps = max(abs(dx), abs(dy))

    # Menghitung perubahan x dan y per langkah
    # Penggunaan float() memastikan hasil adalah pecahan
    Xinc = dx / float(steps)
    Yinc = dy / float(steps)

    # Inisialisasi titik awal
    x = x0
    y = y0

    # Menyimpan titik-titik hasil algoritma DDA
    x_points = [x]  # Tambahkan titik awal
    y_points = [y]  # Tambahkan titik awal

    # Loop untuk menghitung titik-titik sepanjang garis
    # Loop dijalankan sebanyak 'steps' kali (untuk menghitung piksel berikutnya)
    for i in range(int(steps)):
```

```
        x += Xinc
        y += Yinc

        # Menyimpan koordinat dengan pembulatan ke integer terdekat
        x_points.append(round(x))
        y_points.append(round(y))

    # Menggambar garis menggunakan matplotlib
    plt.plot(x_points, y_points, marker='o', color='b')
    plt.title(f"Garis DDA dari ({x0}, {y0}) ke ({x1}, {y1})")
    plt.xlabel("X")
    plt.ylabel("Y")
    plt.grid(True)
    plt.show()

# Meminta input dari pengguna untuk titik awal dan titik akhir
try:
    print("--- Input Titik Awal (x0, y0) ---")
    x0 = int(input("Masukkan koordinat x0 (titik awal): "))
    y0 = int(input("Masukkan koordinat y0 (titik awal): "))
    print("--- Input Titik Akhir (x1, y1) ---")
    x1 = int(input("Masukkan koordinat x1 (titik akhir): "))
    y1 = int(input("Masukkan koordinat y1 (titik akhir): "))

    # Memanggil fungsi DDA untuk menggambar garis
    dda_algorithm(x0, y0, x1, y1)

except ValueError:
    print("\nError: Masukkan harus berupa bilangan bulat (integer).")
```

Hasil RUN

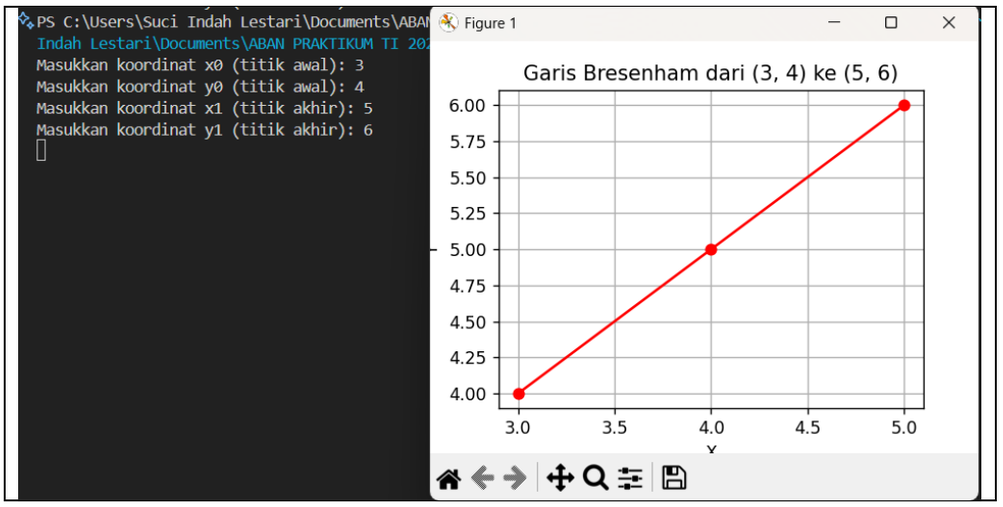


Praktikum 3: Implementasi algoritma Bresenham dengan Input Dinamis

Code Program

```
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
    dx = abs(x1 - x0)
    dy = abs(y1 - y0)
    sx = 1 if x0 < x1 else -1 # penambahan untuk x
    sy = 1 if y0 < y1 else -1 # penambahan untuk y
    err = dx - dy

    x_points = [x]
    y_points = [y]

    while x != x1 or y != y1:
        e2 = 2 * err
        if e2 > -dy:
            err -= dy
            x += sx
        if e2 < dx:
            err += dx
            y += sy
        x_points.append(x)
        y_points.append(y)

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

# Meminta input dari pengguna untuk titik awal dan titik akhir
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)
```

Hasil RUN



Praktikum 4: Menggambar Rumah dengan algoritma DDA

Code Program

```
import matplotlib.pyplot as plt

# Fungsi untuk menggambar garis menggunakan algoritma DDA
def dda_algorithm(x0, y0, x1, y1):
    # Inisialisasi titik awal
    x, y = x0, y0

    # Menghitung perbedaan koordinat
    dx = x1 - x0
    dy = y1 - y0

    # Menentukan jumlah langkah berdasarkan jarak terpanjang (steps)
    steps = abs(dx) if abs(dx) > abs(dy) else abs(dy)

    # Menghitung penambahan x dan y per langkah
    x_inc = dx / steps
    y_inc = dy / steps

    # Menyimpan titik-titik hasil algoritma DDA
    x_points = [x]
    y_points = [y]

    # Loop untuk menghitung titik-titik sepanjang garis
    # Loop berjalan sebanyak 'steps' kali
    for _ in range(int(steps)):
        x += x_inc
        y += y_inc

        # Menyimpan koordinat dengan pembulatan ke integer terdekat
        x_points.append(round(x))
        y_points.append(round(y))

    # Menggambar garis menggunakan matplotlib
    # Note: plt.plot() dalam dda_algorithm hanya menambahkan satu segmen garis ke plot aktif.
    plt.plot(x_points, y_points, marker="o", color="b")

# Fungsi untuk menggambar rumah
def draw_house():
    # Inisialisasi plot dengan ukuran tertentu
```

```
    plt.figure(figsize=(6, 6))

    # Dinding rumah (persegi)
    # Garis Kiri
    dda_algorithm(2, 2, 2, 6)
    # Garis Atas
    dda_algorithm(2, 6, 6, 6)
    # Garis Kanan
    dda_algorithm(6, 6, 6, 2)
    # Garis 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 (persegi kecil)
    # Kiri
    dda_algorithm(3, 2, 3, 4)
    # Atas
    dda_algorithm(3, 4, 4, 4)
    # Kanan
    dda_algorithm(4, 4, 4, 2)

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

    # Menampilkan hasil gambar rumah
    plt.title("Gambar Rumah dengan Algoritma DDA")
    # Batasan sumbu X
    plt.xlim(0, 8)
    # Batasan sumbu Y
    plt.ylim(0, 10)
    # Mengatur aspek rasio agar x dan y memiliki skala yang sama (persegi)
    plt.gca().set_aspect('equal', adjustable='box')
    plt.grid(True)
    plt.show()

# Memanggil fungsi untuk menggambar rumah
draw_house()
```

Hasil RUN

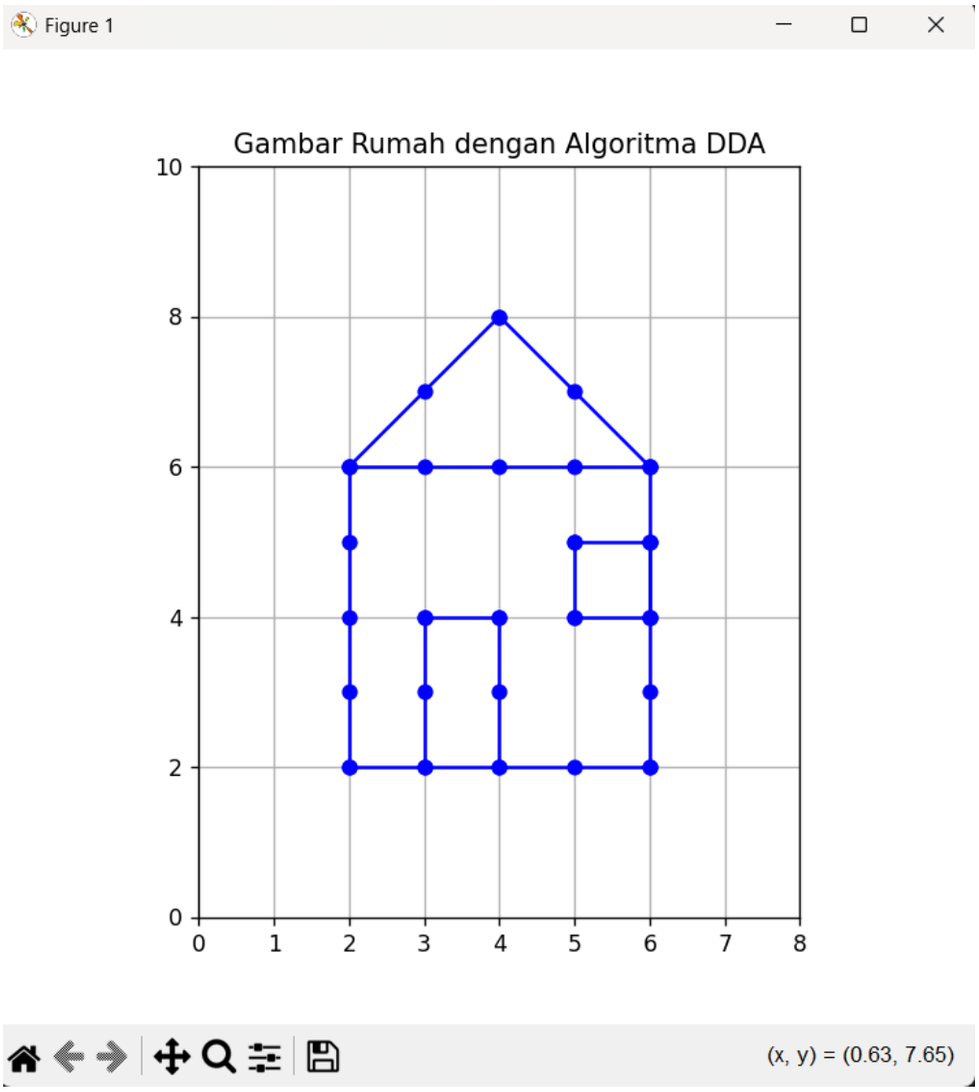


Figure 1
Gambar Rumah dengan Algoritma DDA
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8
(x, y) = (0.63, 7.65)

LATIHAN/TUGAS

Buatlah kode program sederhana untuk menggambar rumah ?

Code Program

```
import matplotlib.pyplot as plt

def dda_algorithm(x0, y0, x1, y1):
    x, y = x0, y0
    dx = x1 - x0
    dy = y1 - y0
    steps = abs(dx) if abs(dx) > abs(dy) else abs(dy)
    x_inc = dx / steps
    y_inc = dy / steps
    x_points = [x]
    y_points = [y]
    for _ in range(int(steps)):
        x += x_inc
        y += y_inc
        x_points.append(round(x))
        y_points.append(round(y))
    plt.plot(x_points, y_points, marker="o", color="b")

def draw_house():
    plt.figure(figsize=(6, 6))
    dda_algorithm(2, 2, 2, 6)
    dda_algorithm(2, 6, 6, 6)
    dda_algorithm(6, 6, 6, 2)
    dda_algorithm(6, 2, 2, 2)
    dda_algorithm(2, 6, 4, 8)
    dda_algorithm(4, 8, 6, 6)
    dda_algorithm(2, 4, 3, 4)
    dda_algorithm(3, 2, 3, 4)
    dda_algorithm(3, 4, 4, 4)
    dda_algorithm(4, 4, 4, 2)
    dda_algorithm(2, 5, 3, 5)
    dda_algorithm(3, 5, 3, 4)
    dda_algorithm(5, 4, 5, 5)
    dda_algorithm(5, 5, 6, 5)
    dda_algorithm(6, 5, 6, 4)
    dda_algorithm(6, 4, 5, 4)
    plt.title("Gambar Rumah dengan Algoritma DDA")
    plt.xlim(0, 8)
    plt.ylim(0, 10)
    plt.gca().set_aspect('equal', adjustable='box')
    plt.grid(True)
    plt.show()

draw_house()
```

Hasil RUN

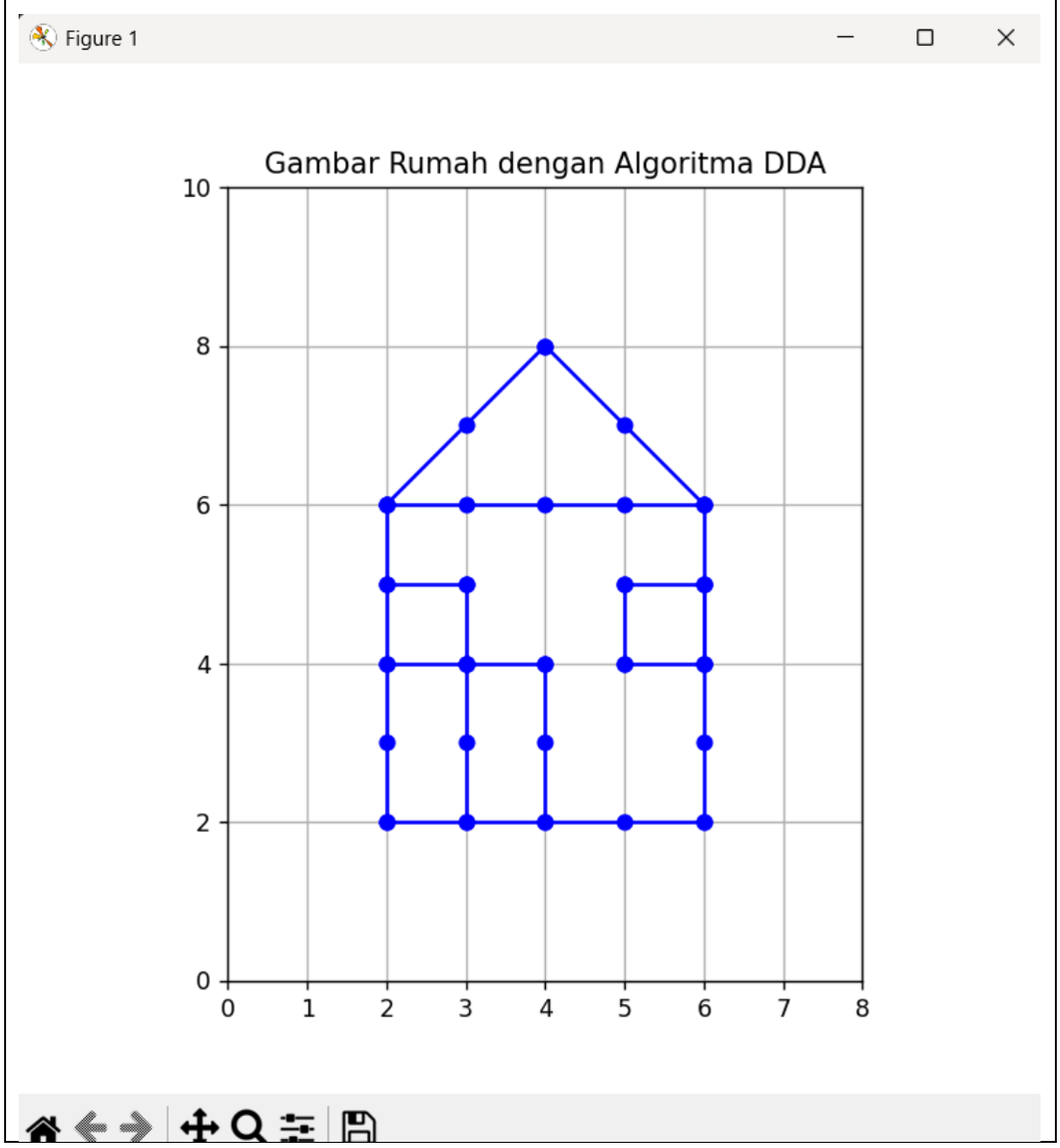


Figure 1
Gambar Rumah dengan Algoritma DDA
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8