

LAPORAN PRAKTIKUM

GRAFIKA KOMPUTER

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

MODUL 6



Nama : Rio Andika Andriansyah

NIM : 20230810155

Kelas : TINFC-2023-04

TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

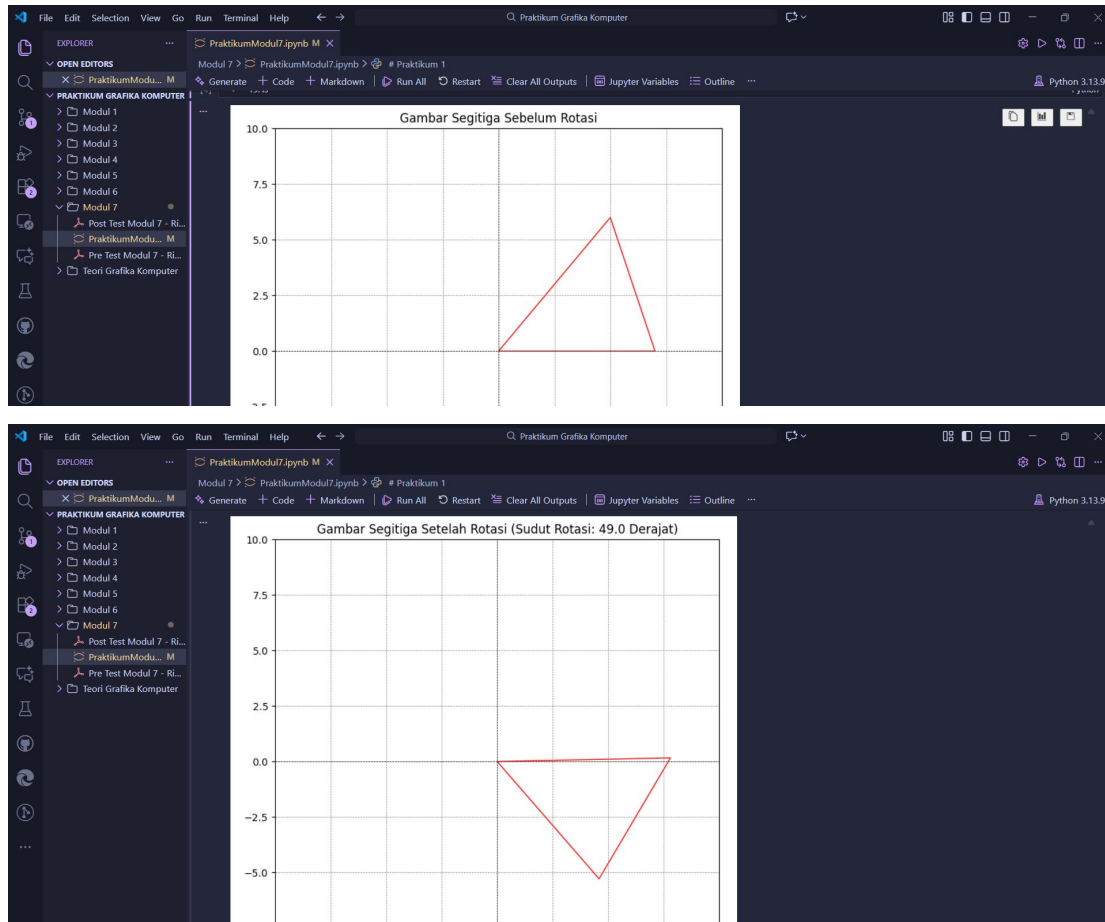
UNIVERSITAS KUNINGAN

DAFTAR ISI

DAFTAR ISI	i
PRAKTIKUM	1
Praktikum 1	1
Praktikum 2	2
Praktikum 3	4
Praktikum 4	6
TUGAS	8

PRAKTIKUM

Praktikum 1



Source Code:

```
# Praktikum 1

# Import library yang diperlukan
import numpy as np
import matplotlib.pyplot as plt

# Fungsi untuk menggambar segitiga
def gambar_segitiga(titik):
    # Menggambar segitiga berdasarkan titik yang diberikan
    segitiga = plt.Polygon(titik, closed=True, fill=None, edgecolor='r')
    plt.gca().add_patch(segitiga)

# Fungsi untuk melakukan rotasi
def rotasi(titik, sudut):
    # Menghitung sudut dalam radian
    radian = np.radians(sudut)

    # Matriks rotasi
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)],
                              [np.sin(radian), np.cos(radian)]])

    # Melakukan rotasi pada setiap titik segitiga
    titik_rotated = np.dot(titik, rotasi_matrix)
    return titik_rotated

# Input titik segitiga dari pengguna
x1 = float(input("Masukkan koordinat x1: "))
```

```

y1 = float(input("Masukkan koordinat y1: "))
x2 = float(input("Masukkan koordinat x2: "))
y2 = float(input("Masukkan koordinat y2: "))
x3 = float(input("Masukkan koordinat x3: "))
y3 = float(input("Masukkan koordinat y3: "))

# Titik-titik segitiga
titik_segitiga = np.array([[x1, y1], [x2, y2], [x3, y3]])

# Input sudut rotasi dari pengguna
sudut_rotasi = float(input("Masukkan sudut rotasi (dalam derajat): "))

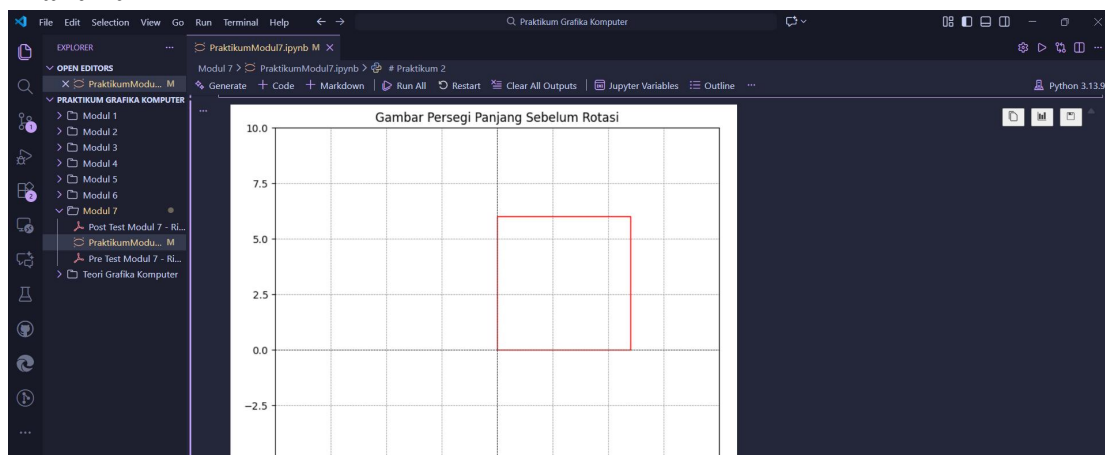
# Menggambar segitiga sebelum rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Segitiga Sebelum Rotasi')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar segitiga asli
gambar_segitiga(titik_segitiga)
plt.show()

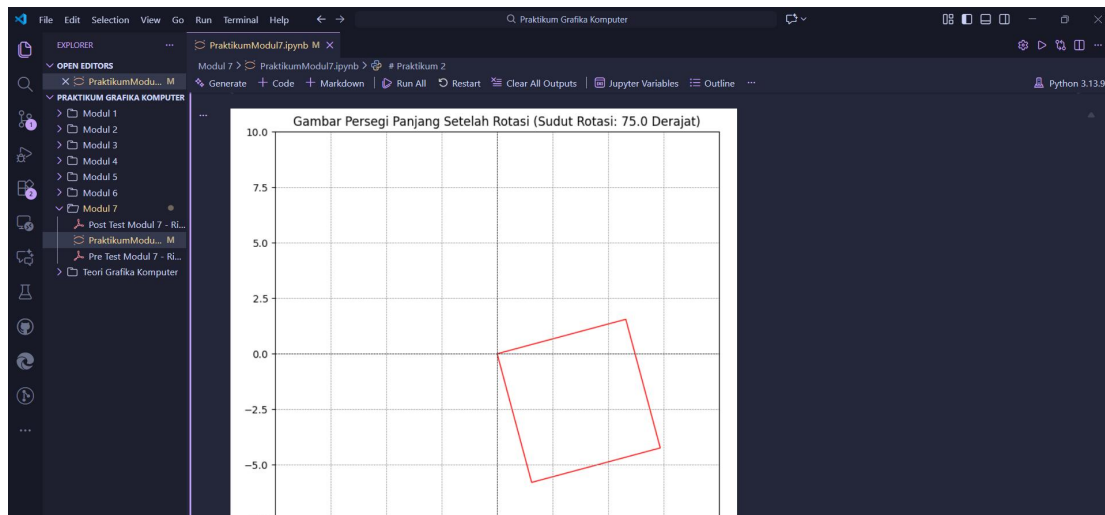
# Melakukan rotasi
titik_rotated = rotasi(titik_segitiga, sudut_rotasi)

# Menggambar segitiga setelah rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Segitiga Setelah Rotasi (Sudut Rotasi: {} Derajat)'.format(sudut_rotasi))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar segitiga yang sudah dirotasi
gambar_segitiga(titik_rotated)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
# Amati hasilnya

```

Praktikum 2





Source Code:

```
# Praktikum 2

# Fungsi untuk menggambar persegi panjang
def gambar_persegi_panjang(titik):
    # Menggambar persegi panjang berdasarkan titik yang diberikan
    persegi_panjang = plt.Polygon(titik, closed=True, fill=None,
    edgecolor='r')
    plt.gca().add_patch(persegi_panjang)

# Titik-titik persegi panjang
# Input titik-titik persegi panjang dari pengguna
x1 = float(input("Masukkan koordinat x1: "))
y1 = float(input("Masukkan koordinat y1: "))
x2 = float(input("Masukkan koordinat x2: "))
y2 = float(input("Masukkan koordinat y2: "))
x3 = float(input("Masukkan koordinat x3: "))
y3 = float(input("Masukkan koordinat y3: "))
x4 = float(input("Masukkan koordinat x4: "))
y4 = float(input("Masukkan koordinat y4: "))
titik_persegi_panjang = np.array([[x1, y1], [x2, y2], [x3, y3], [x4, y4]])

# Input sudut rotasi dari pengguna
sudut_rotasi = float(input("Masukkan sudut rotasi (dalam derajat): "))

# Fungsi untuk melakukan rotasi
def rotasi(titik, sudut):
    # Menghitung sudut dalam radian
    radian = np.radians(sudut)

    # Matriks rotasi
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)],
    [np.sin(radian), np.cos(radian)]])

    # Melakukan rotasi pada setiap titik segitiga
    titik_rotated = np.dot(titik, rotasi_matrix)
    return titik_rotated

# Menggambar persegi panjang sebelum rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Persegi Panjang Sebelum Rotasi')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar persegi panjang asli
```

```

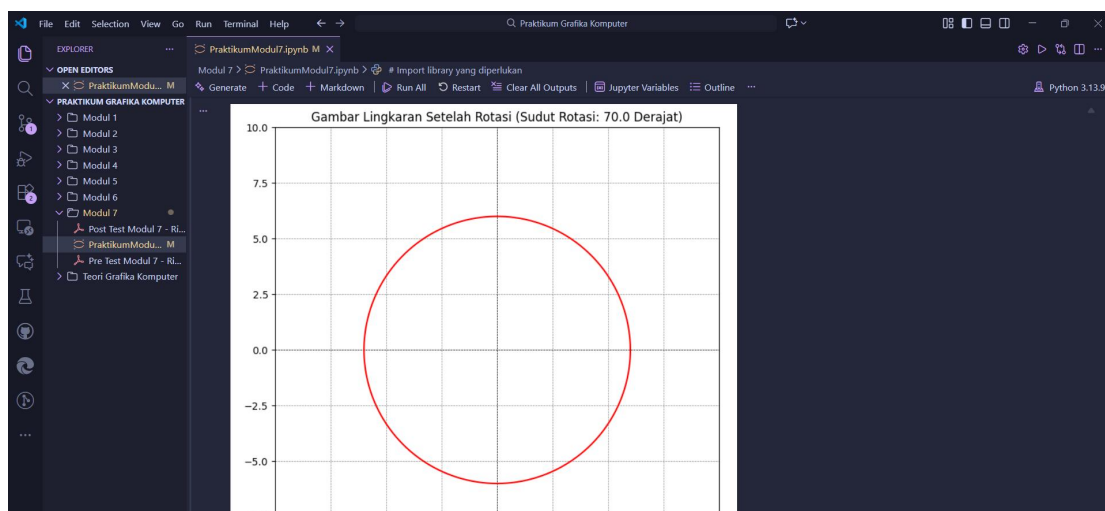
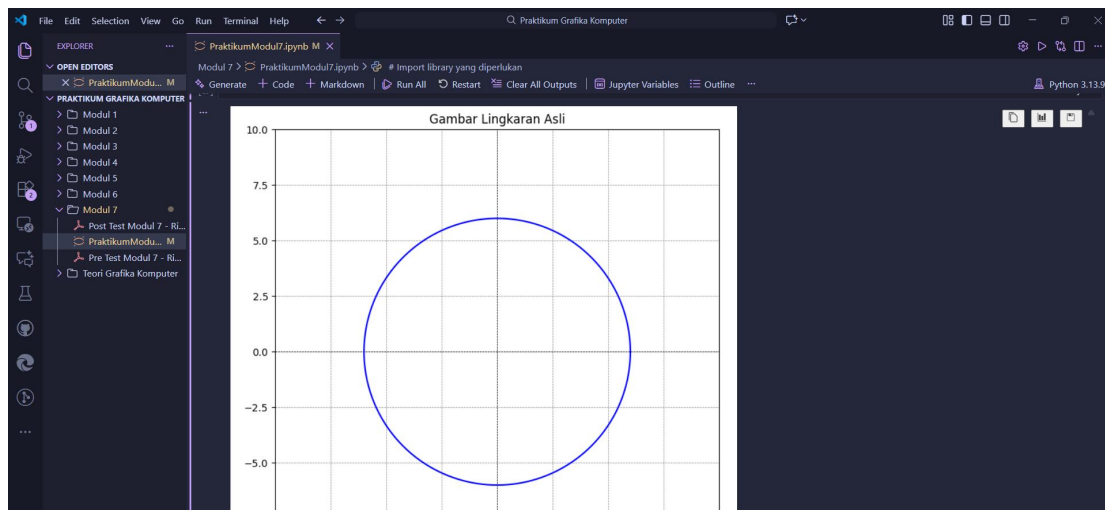
gambar_persegi_panjang(titik_persegi_panjang)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

# Melakukan rotasi
titik_rotated = rotasi(titik_persegi_panjang, sudut_rotasi)

# Menggambar persegi panjang setelah rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Persegi Panjang Setelah Rotasi (Sudut Rotasi: {}
Derajat)'.format(sudut_rotasi))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar persegi panjang yang sudah dirotasi
gambar_persegi_panjang(titik_rotated)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

```

Praktikum 3



Source Code:

```

# Import library yang diperlukan
import numpy as np

```

```

import matplotlib.pyplot as plt

# Fungsi untuk menggambar lingkaran
def gambar_lingkaran(center_x, center_y, radius, color='b'):
    # Membuat data lingkaran (1000 titik)
    theta = np.linspace(0, 2*np.pi, 1001)
    x = radius * np.cos(theta) + center_x
    y = radius * np.sin(theta) + center_y

    # Menggambar lingkaran
    plt.plot(x, y, color=color)

# Fungsi untuk melakukan rotasi titik
def rotasi_titik(titik, center, radius, sudut):
    # Menghitung sudut dalam radian
    radian = np.radians(sudut)

    # Memindahkan titik ke pusat (0, 0)
    x_translated = titik[0] - center[0]
    y_translated = titik[1] - center[1]

    # Melakukan rotasi
    x_rotated = x_translated * np.cos(radian) - y_translated * np.sin(radian)
    y_rotated = x_translated * np.sin(radian) + y_translated * np.cos(radian)

    # Memindahkan kembali ke posisi semula
    x_rotated += center[0]
    y_rotated += center[1]

    return np.array([x_rotated, y_rotated])

# Input pusat lingkaran dan radius dari pengguna
center_x = float(input("Masukkan koordinat pusat x: "))
center_y = float(input("Masukkan koordinat pusat y: "))
radius = float(input("Masukkan radius lingkaran: "))
center = np.array([center_x, center_y])

# Input sudut rotasi dari pengguna
sudut_rotasi = float(input("Masukkan sudut rotasi (dalam derajat): "))

# Menggambar lingkaran sebelum rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Lingkaran Asli')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar lingkaran
gambar_lingkaran(center_x, center_y, radius, 'b')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

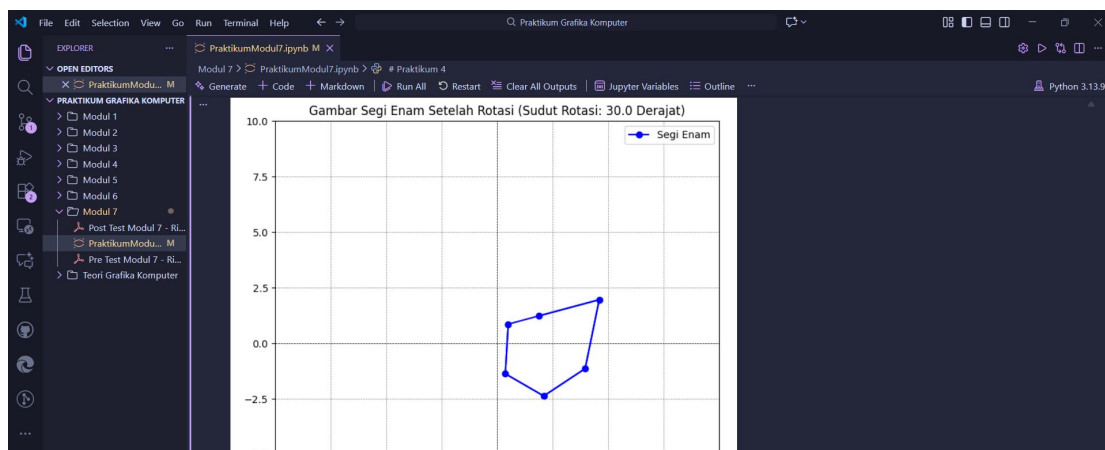
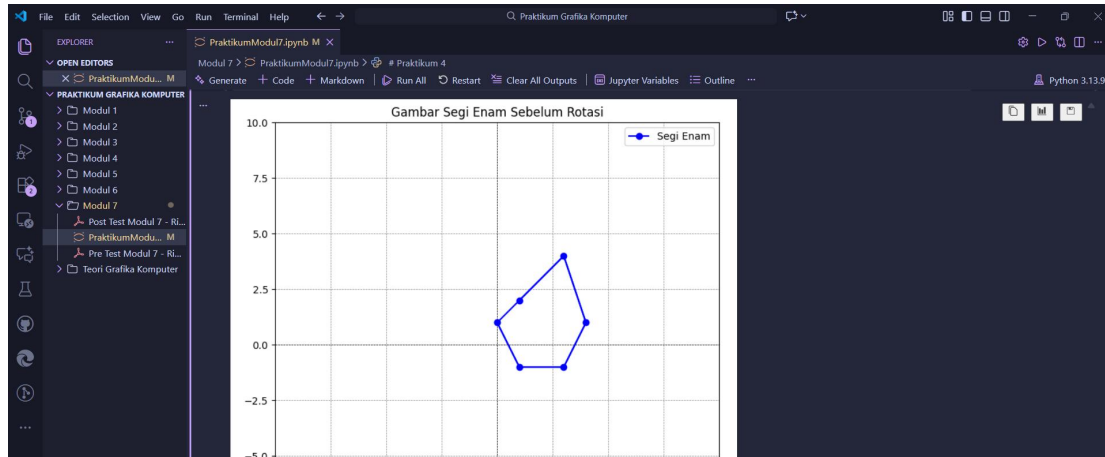
# Melakukan rotasi
# Rotasi lingkaran dilakukan pada pusatnya (rotasi pusat)
x_rotated, y_rotated = rotasi_titik(center, center, radius, sudut_rotasi)

# Menggambar lingkaran setelah rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Lingkaran Setelah Rotasi (Sudut Rotasi: {} Derajat)'.format(sudut_rotasi))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)

```

```
# Gambar lingkaran yang sudah dirotasi
gambar_lingkaran(x_rotated, y_rotated, radius, 'r')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
# Amati hasilnya
```

Praktikum 4



Source Code:

```
# Praktikum 4

# Import library yang diperlukan
import numpy as np
import matplotlib.pyplot as plt

# Fungsi untuk menggambar segi enam
def gambar_segi_enam(titik):
    # Menggambar segi enam dengan menambahkan titik pertama ke akhir (agar tertutup)
    plt.plot(titik[:, 0], titik[:, 1], 'o-', label='Segi Enam', color='b')
    titik_tutup = np.append(titik, [titik[0]], axis=0)
    plt.plot(titik_tutup[:, 0], titik_tutup[:, 1], 'k-', color='b')

# Fungsi untuk melakukan rotasi
def rotasi(titik, sudut):
    # Menghitung sudut dalam radian
    radian = np.radians(sudut)

    # Matriks rotasi
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)],
                               [np.sin(radian), np.cos(radian)]])
```



```

    # Melakukan rotasi pada setiap titik
    titik_rotated = np.dot(titik, rotasi_matrix)
    return titik_rotated

# Input titik-titik segi enam dari pengguna
x = []
y = []
print("Input koordinat titik segi enam (x, y):")
for i in range(1, 7):
    x_i = float(input(f"Masukkan koordinat x titik {i}: "))
    y_i = float(input(f"Masukkan koordinat y titik {i}: "))
    x.append(x_i)
    y.append(y_i)

# Konversi list menjadi array numpy
titik = np.array([x, y]).T
titik_array = np.array(titik)

# Input sudut rotasi dari pengguna
sudut_rotasi = float(input("Masukkan sudut rotasi (dalam derajat): "))

# Menggambar segi enam sebelum rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Segi Enam Sebelum Rotasi')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar segi enam asli
gambar_segi_enam(titik)
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
plt.show()

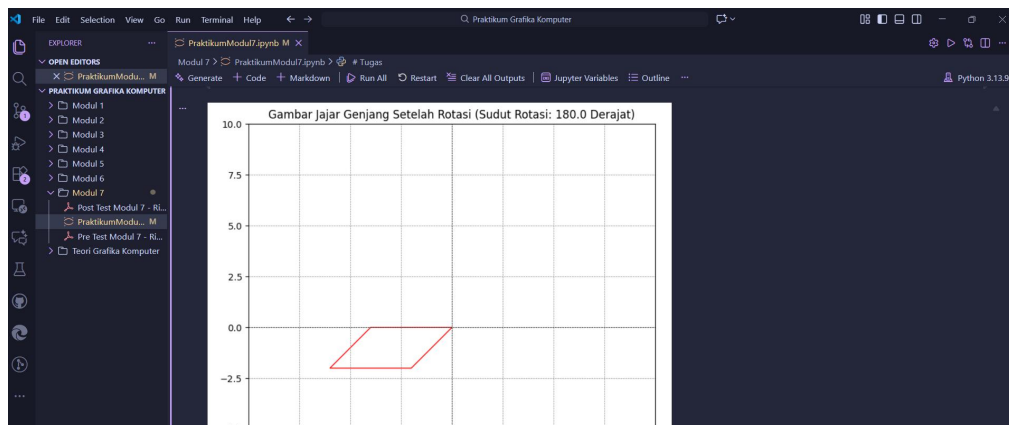
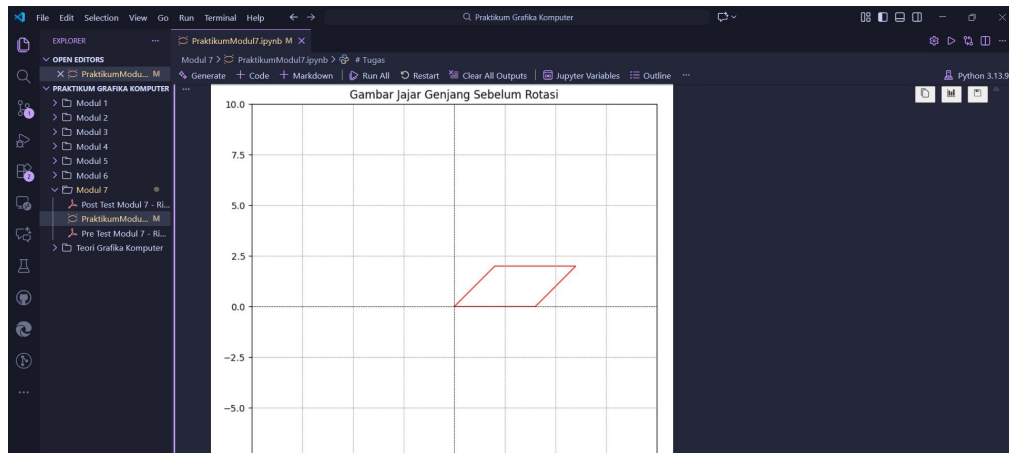
# Melakukan rotasi
titik_rotated = rotasi(titik, sudut_rotasi)

# Menggambar segi enam setelah rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Segi Enam Setelah Rotasi (Sudut Rotasi: {} Derajat)'.format(sudut_rotasi))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar segi enam yang sudah dirotasi
gambar_segi_enam(titik_rotated)
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
plt.show()
# Amati hasilnya

```

TUGAS

1. Buatlah kode program sederhana untuk merotasikan jajar genjang sebanyak 180 derajat!



Source Code:

```
# Tugas

def gambar_jajar_genjang(titik):
    jajar_genjang = plt.Polygon(titik, closed=True, fill=None,
                                edgecolor='r')
    plt.gca().add_patch(jajar_genjang)

x1 = float(input("Masukkan koordinat x1: "))
y1 = float(input("Masukkan koordinat y1: "))
x2 = float(input("Masukkan koordinat x2: "))
y2 = float(input("Masukkan koordinat y2: "))
x3 = float(input("Masukkan koordinat x3: "))
y3 = float(input("Masukkan koordinat y3: "))
x4 = float(input("Masukkan koordinat x4: "))
y4 = float(input("Masukkan koordinat y4: "))
titik_jajar_genjang = np.array([[x1, y1], [x2, y2], [x3, y3], [x4, y4]])

# Input sudut rotasi dari pengguna
sudut_rotasi = 180.0 # Perintah rotasi 180 derajat
```

```

# Fungsi untuk melakukan rotasi
def rotasi(titik, sudut):
    # Menghitung sudut dalam radian
    radian = np.radians(sudut)

    # Matriks rotasi
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)],
                               [np.sin(radian), np.cos(radian)]])

    # Melakukan rotasi pada setiap titik segitiga
    titik_rotated = np.dot(titik, rotasi_matrix)
    return titik_rotated

# Menggambar persegi panjang sebelum rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Jajar Genjang Sebelum Rotasi')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar persegi panjang asli
gambar_persegi_panjang(titik_jajar_genjang)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

# Melakukan rotasi
titik_rotated = rotasi(titik_jajar_genjang, sudut_rotasi)

# Menggambar jajar genjang setelah rotasi
plt.figure(figsize=(8, 8))
plt.title('Gambar Jajar Genjang Setelah Rotasi (Sudut Rotasi: {}
Derajat)'.format(sudut_rotasi))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar jajar genjang yang sudah dirotasi
gambar_jajar_genjang(titik_rotated)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

```

Github: <https://github.com/Andrian206/Praktikum-Grafika-Komputer/tree/main/Modul%207>