

**LAPORAN PRAKTIKUM
GRAFIKA KOMPUTER**
(DOSEN PENGAMPU : Rio Priantama, S.T., M.T.I)

Modul 7



DISUSUN OLEH :
NAMA: MOHAMAD ABAN SY'BANA
NIM : 20230810012
KELAS : TINFC-2023-04

**TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS KUNINGAN
2025**

PRAKTIKUM

PRAKTIKUM 1

Code Program

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

# Fungsi untuk menggambar segitiga
def gambar_segitiga(titik):
    segitiga = plt.Polygon(titik, closed=True, fill=False, edgecolor='b')
    plt.gca().add_patch(segitiga)

# Fungsi untuk rotasi titik menggunakan matriks rotasi
def rotasi(titik, sudut):
    radian = np.radians(sudut)
    # Matriks rotasi 2D
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)], [np.sin(radian), np.cos(radian)]])
    # Perkalian matriks untuk transformasi rotasi
    titik_rotated = np.dot(titik, rotasi_matrix.T)
    return titik_rotated

# Input koordinat titik-titik segitiga
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: '))
sudut_rotasi = float(input('Masukkan sudut rotasi (dalam derajat): '))

# Buat array numpy dari titik-titik segitiga
titik_asli = np.array([[x1, y1], [x2, y2], [x3, y3]])

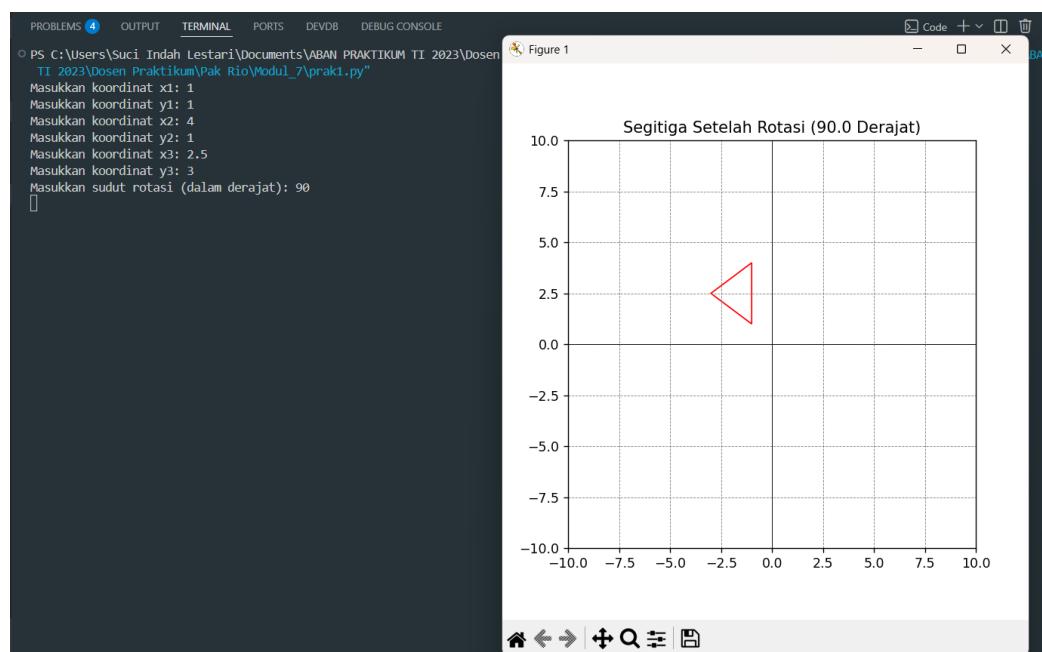
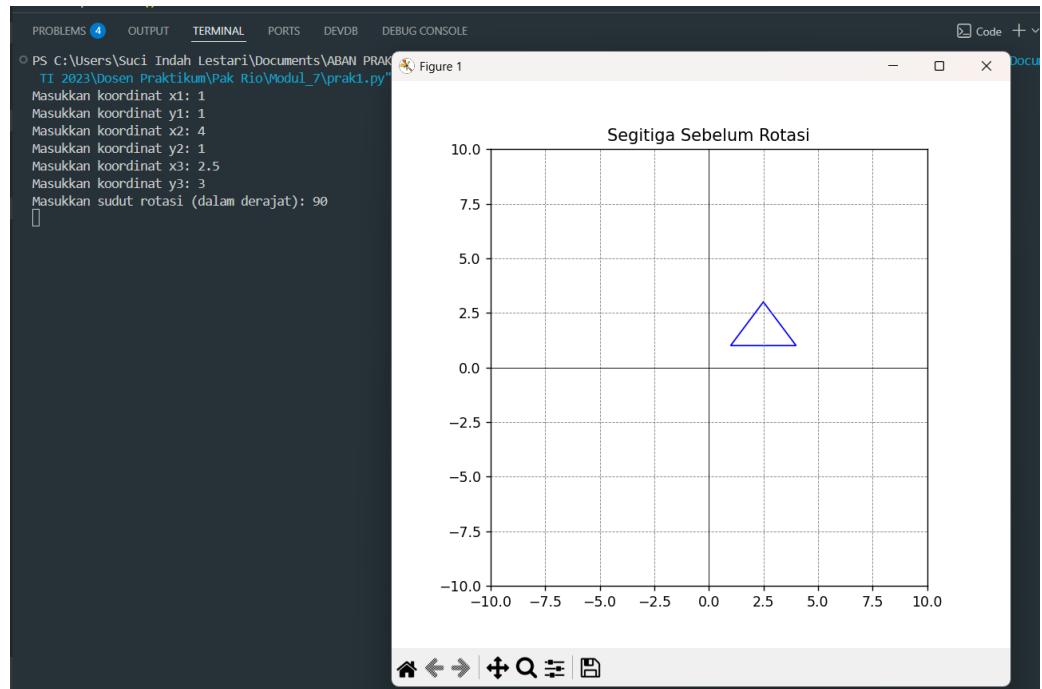
# Gambar segitiga sebelum rotasi
plt.figure(figsize=(8, 8))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
gambar_segitiga(titik_asli)
plt.title('Segitiga Sebelum Rotasi')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

# Proses rotasi segitiga
titik_rotated = rotasi(titik_asli, sudut_rotasi)

plt.figure(figsize=(8, 8))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
# Gambar segitiga hasil rotasi dengan warna merah
seg = plt.Polygon(titik_rotated, closed=True, fill=False, edgecolor='r')
plt.gca().add_patch(seg)
plt.title(f'Segitiga Setelah Rotasi ({sudut_rotasi} Derajat)')
```

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

Hasil RUN



PRAKTIKUM 2

Code Program

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

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

def rotasi(titik, sudut):
    # Konversi derajat ke radian
    radian = np.radians(sudut)

    # Matriks rotasi 2D
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)],
                             [np.sin(radian), np.cos(radian)]])

    # Mengalikan titik dengan matriks rotasi
    titik_rotated = np.dot(titik, rotasi_matrix)
    return titik_rotated

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: "))

# Menyimpan koordinat 4 titik persegi panjang dalam array
titik_asli = np.array([[x1, y1], [x2, y2], [x3, y3], [x4, y4]])

sudut_rotasi = float(input("Masukkan sudut rotasi (dalam derajat): "))

# Visualisasi persegi panjang sebelum rotasi
plt.figure(figsize=(8, 8))
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)

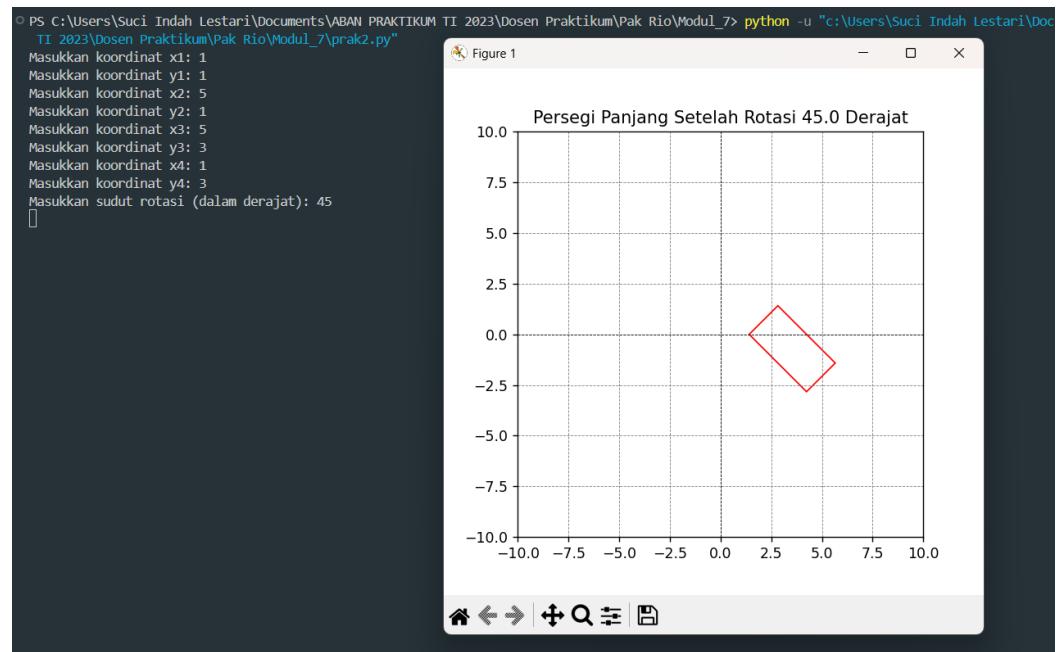
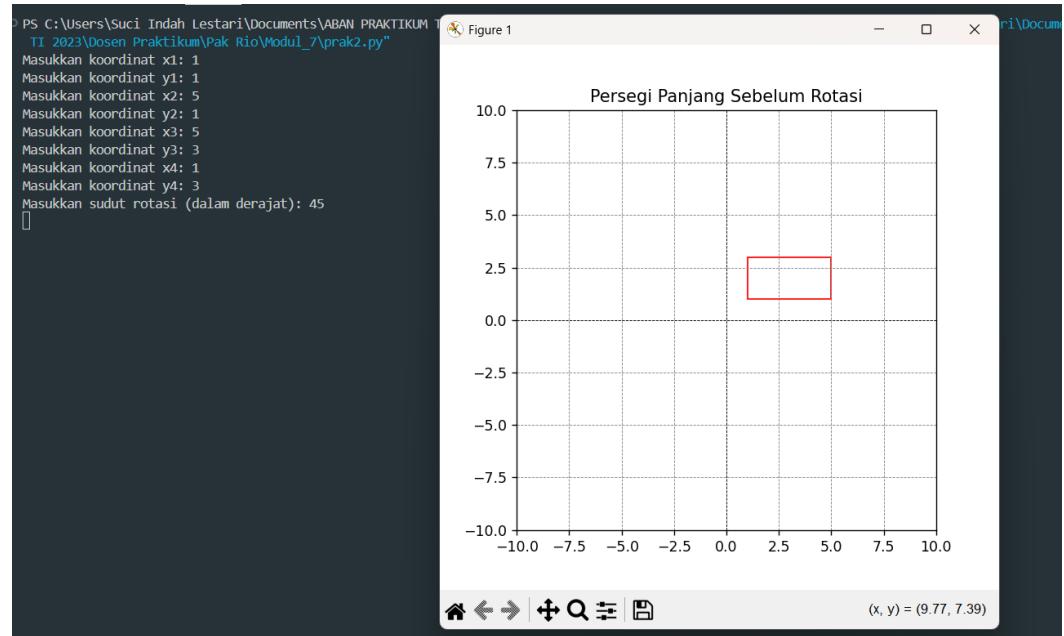
gambar_persegi_panjang(titik_asli)
plt.title('Persegi Panjang Sebelum Rotasi')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

# Melakukan rotasi pada titik-titik persegi panjang
titik_rotated = rotasi(titik_asli, sudut_rotasi)

# Visualisasi persegi panjang setelah rotasi
plt.figure(figsize=(8, 8))
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(titik_rotated)
plt.title('Persegi Panjang Setelah Rotasi {sudut_rotasi} Derajat')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

Hasil RUN



PRAKTIKUM 3

Code Program

```
# Import library untuk operasi matematika dan plotting
import numpy as np
import matplotlib.pyplot as plt

# Fungsi untuk menggambar lingkaran
def gambar_lingkaran(radius, center):
    # Membuat 200 titik untuk menggambar lingkaran
    theta = np.linspace(0, 2 * np.pi, 200)
    # Konversi koordinat polar ke kartesian
    x = radius * np.cos(theta) + center[0]
    y = radius * np.sin(theta) + center[1]
    plt.plot(x, y, label='Lingkaran', color='b')

# Fungsi untuk merotasi lingkaran
def rotasi(center, radius, sudut):
    theta = np.linspace(0, 2 * np.pi, 200)
    # Konversi derajat ke radian
    rad = np.radians(sudut)
    # Rotasi lingkaran dengan menambahkan sudut rotasi pada theta
    x_rot = radius * np.cos(theta + rad) + center[0]
    y_rot = radius * np.sin(theta + rad) + center[1]
    return x_rot, y_rot

# Input data dari user
center_x = float(input('Masukkan koordinat pusat x: '))
center_y = float(input('Masukkan koordinat pusat y: '))
radius = float(input('Masukkan radius lingkaran: '))
sudut_rotasi = float(input('Masukkan sudut rotasi (dalam derajat): '))

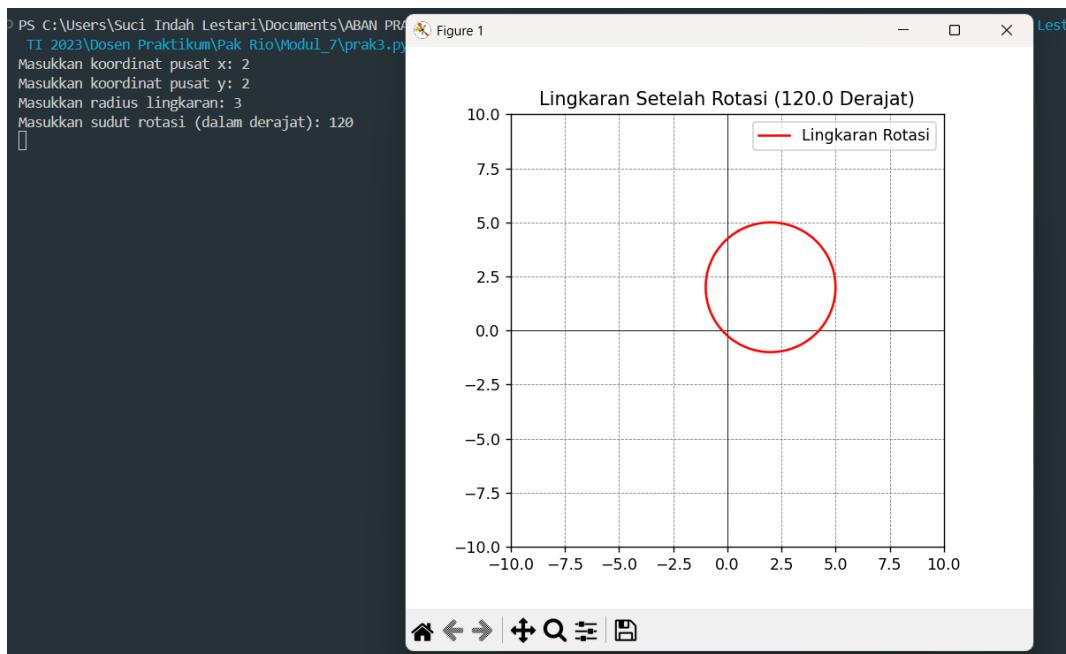
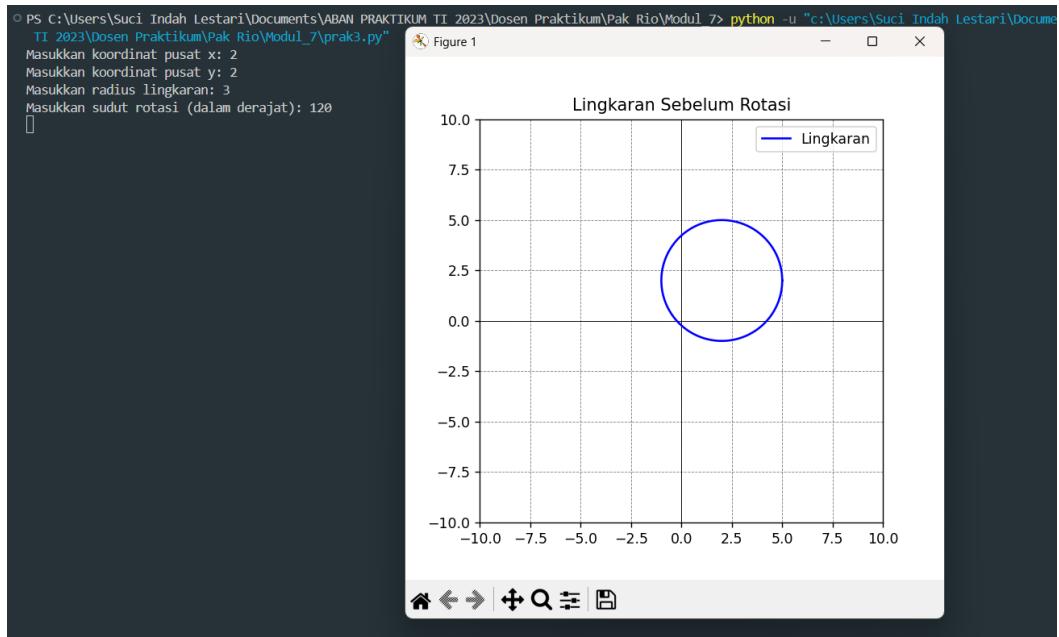
# Visualisasi lingkaran sebelum rotasi
plt.figure(figsize=(8, 8))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
gambar_lingkaran(radius, (center_x, center_y))
plt.title('Lingkaran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

# Melakukan rotasi pada lingkaran
x_rot, y_rot = rotasi((center_x, center_y), radius, sudut_rotasi)

# Visualisasi lingkaran setelah rotasi
plt.figure(figsize=(8, 8))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
plt.plot(x_rot, y_rot, label='Lingkaran Rotasi', color='r')
plt.title(f'Lingkaran Setelah Rotasi ({sudut_rotasi} Derajat)')
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
```

```
plt.show()
```

Hasil RUN



PRAKTIKUM 4

Code Program

```
# Import library untuk operasi array dan plotting
import numpy as np
import matplotlib.pyplot as plt

# Fungsi untuk menggambar segi enam
def gambar_segi_enam(titik):
    # Menambahkan titik pertama di akhir untuk menutup bentuk
    titik_plot = np.vstack([titik, titik[0]])
    plt.plot(titik_plot[:,0], titik_plot[:,1], label='Segi Enam', color='b')

# Fungsi untuk melakukan rotasi titik-titik
def rotasi(titik, sudut):
    radian = np.radians(sudut)
    # Matriks rotasi 2D
    rotasi_matrix = np.array([[np.cos(radian), -np.sin(radian)], [np.sin(radian), np.cos(radian)]])
    # Mengalikan titik dengan matriks rotasi
    titik_rotated = np.dot(titik, rotasi_matrix.T)
    return titik_rotated

# Input koordinat 6 titik segi enam dari user
print('Masukkan koordinat titik-titik segi enam (x, y):')
titik = []
for i in range(6):
    x = float(input(f'Koordinat x titik {i+1}: '))
    y = float(input(f'Koordinat y titik {i+1}: '))
    titik.append([x, y])
# Konversi list ke numpy array
titik = np.array(titik)
# Input sudut rotasi dari user
sudut_rotasi = float(input('Masukkan sudut rotasi (dalam derajat): '))

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

# Melakukan rotasi pada titik-titik segi enam
titik_rotated = rotasi(titik, sudut_rotasi)

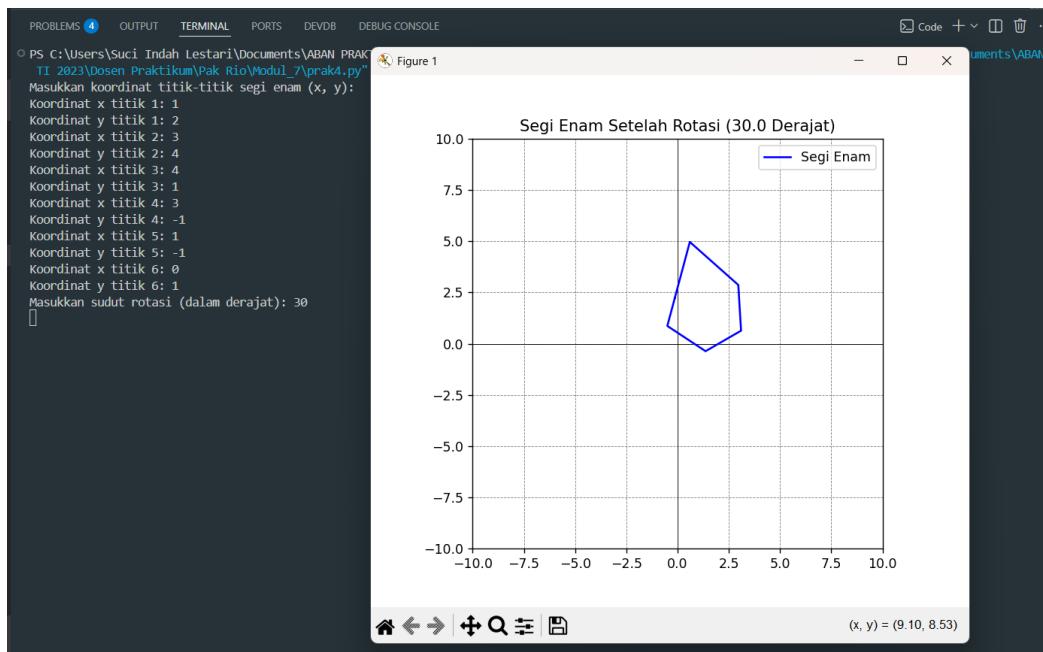
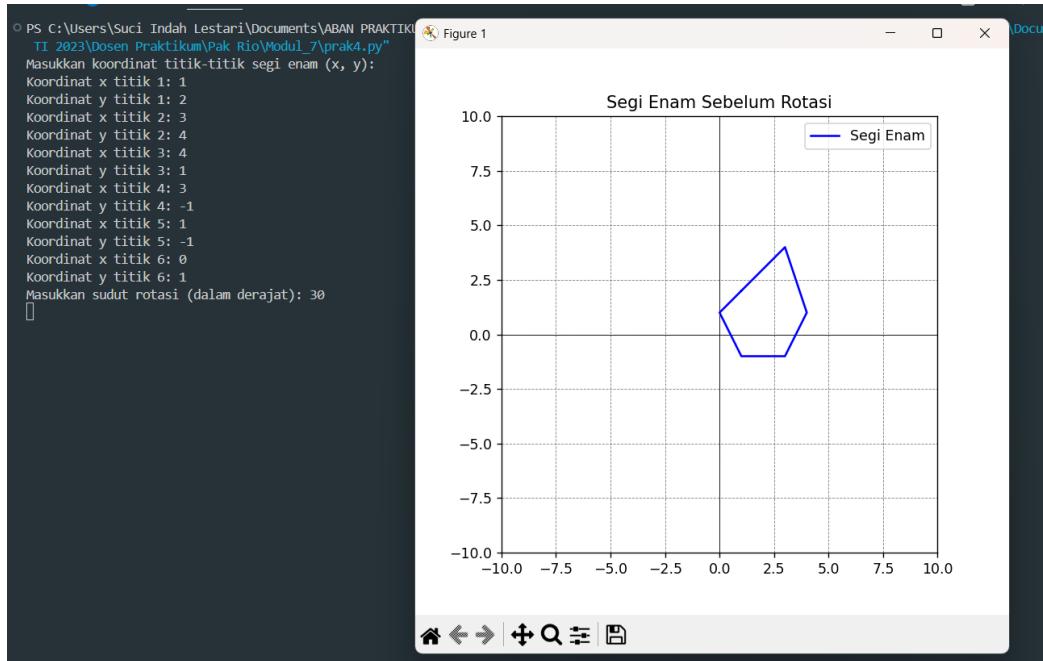
# Visualisasi segi enam setelah rotasi
plt.figure(figsize=(8, 8))
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
gambar_segi_enam(titik_rotated)
```

```

plt.title(f'Segi Enam Setelah Rotasi ({sudut_rotasi} Derajat)')
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
plt.show()

```

Hasil RUN



TUGAS

1. Buatlah kode program sederhana untuk menggambar objek jajargenjang yang dirotasi sebanyak 180 derajat.

Code Program

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

def draw_parallelogram(pts):
    poly = plt.Polygon(pts, closed=True, fill=False, edgecolor='b')
    plt.gca().add_patch(poly)

def rotate_about_centroid(pts, angle):
    rad = np.radians(angle)
    R = np.array([[np.cos(rad), -np.sin(rad)], [np.sin(rad), np.cos(rad)]])
    c = pts.mean(axis=0)
    return (pts - c).dot(R.T) + c

# Input koordinat titik-titik jajargenjang
print("Masukkan koordinat 4 titik jajargenjang:")
x1 = float(input("x1: "))
y1 = float(input("y1: "))
x2 = float(input("x2: "))
y2 = float(input("y2: "))
x3 = float(input("x3: "))
y3 = float(input("y3: "))
x4 = float(input("x4: "))
y4 = float(input("y4: "))

pts = np.array([[x1, y1], [x2, y2], [x3, y3], [x4, y4]])
angle = int(input("Masukkan sudut rotasi (derajat): "))

plt.figure(figsize=(6, 6))
plt.xlim(-6, 6)
plt.ylim(-6, 6)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
draw_parallelogram(pts)
plt.title('Jajargenjang Sebelum Rotasi')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()

rot_pts = rotate_about_centroid(pts, angle)

plt.figure(figsize=(6, 6))
plt.xlim(-6, 6)
plt.ylim(-6, 6)
plt.axhline(0, color='black', linewidth=0.5)
plt.axvline(0, color='black', linewidth=0.5)
plt.grid(color='gray', linestyle='--', linewidth=0.5)
plt.gca().add_patch(plt.Polygon(rot_pts, closed=True, fill=False, edgecolor='r'))
plt.title(f'Jajargenjang Setelah Rotasi ({angle} Derajat)')
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

Hasil RUN

```
PROBLEMS 5 OUTPUT TERMINAL PORTS DEVDB DEBUG Figure 1
PS C:\Users\Suci Indah Lestari\Documents\ABAN PRAKTIKUM
TI 2023\Dosen Praktikum\Pak Rio\Modul_7\tugas.py"
Masukkan koordinat 4 titik jajargenjang:
x1: 1
y1: 1
x2: 4
y2: 1
x3: 5
y3: 3
x4: 2
y4: 3
Masukkan sudut rotasi (derajat): 180
[]
```

Jajargenjang Sebelum Rotasi

A coordinate grid showing a blue parallelogram. The x-axis and y-axis both range from -6 to 6. The parallelogram has vertices at approximately (1, 1), (4, 1), (5, 3), and (2, 3).

```
PROBLEMS 5 OUTPUT TERMINAL PORTS DEVDB DEBUG CONSOLE Figure 1
PS C:\Users\Suci Indah Lestari\Documents\ABAN PRAKTIKUM
TI 2023\Dosen Praktikum\Pak Rio\Modul_7\tugas.py"
Masukkan koordinat 4 titik jajargenjang:
x1: 1
y1: 1
x2: 4
y2: 1
x3: 5
y3: 3
x4: 2
y4: 3
Masukkan sudut rotasi (derajat): 180
[]
```

Jajargenjang Setelah Rotasi (180 Derajat)

A coordinate grid showing a red parallelogram. The x-axis and y-axis both range from -6 to 6. The parallelogram has vertices at approximately (1, 1), (4, 1), (3, 3), and (2, 3), which are the result of a 180-degree counter-clockwise rotation of the original blue parallelogram.