

LAPORAN PRAKTIKUM
GRAFIKA KOMPUTER
(DOSEN PENGAMPU : Rio Priantama, S.T., M.T.I)
Modul 6



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

TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS KUNINGAN
2025

PRAKTIKUM

Prak 1

Code Program

```
import matplotlib.pyplot as plt

def draw_triangle(vertices, title):
    """Fungsi untuk menggambar segitiga berdasarkan titik-titik yang diberikan."""
    # Memecah titik menjadi koordinat x dan y
    x, y = zip(*vertices)

    # Menutup segitiga dengan menambahkan titik pertama ke akhir
    x += (x[0],)
    y += (y[0],)

    plt.figure()
    plt.plot(x, y, marker='o')
    plt.title(title)
    plt.xlim(-100, 100)
    plt.ylim(-100, 100)

    plt.axhline(0, color='black', linewidth=0.5, ls='--') # Garis horizontal
    plt.axvline(0, color='black', linewidth=0.5, ls='--') # Garis vertikal
    plt.grid()
    plt.gca().set_aspect('equal', adjustable='box')
    plt.show()

def scale_triangle(vertices, scale_factor):
    """Melakukan penskalaan segitiga berdasarkan faktor skala."""
    scaled_vertices = [(x * scale_factor, y * scale_factor) for x, y in vertices]
    return scaled_vertices

# Input titik-titik segitiga
print("Masukkan titik-titik segitiga (x1, y1), (x2, y2), (x3, y3):")
x1, y1 = map(float, input("Titik 1 (x1 y1): ").split())
x2, y2 = map(float, input("Titik 2 (x2 y2): ").split())
x3, y3 = map(float, input("Titik 3 (x3 y3): ").split())
triangle_vertices = [(x1, y1), (x2, y2), (x3, y3)]

# Input faktor skala
scale_factor = float(input("Masukkan faktor skala: "))

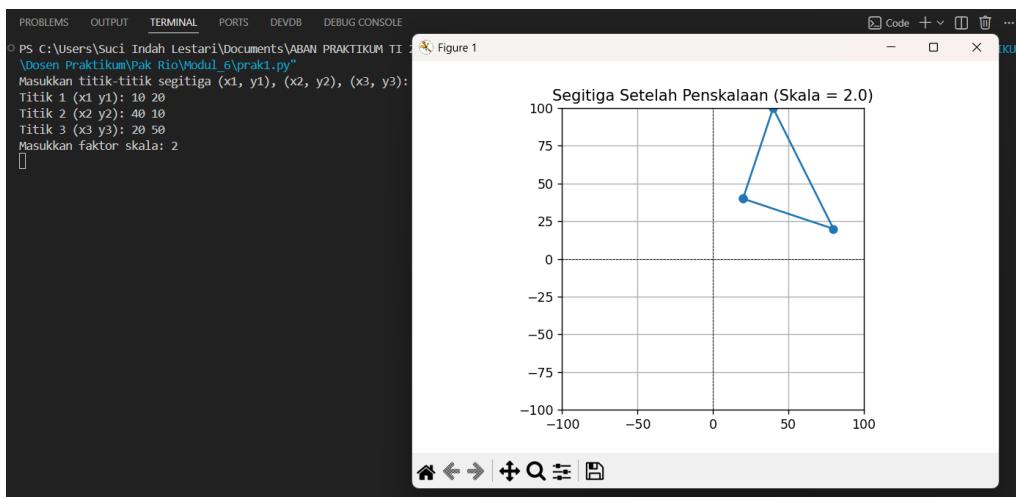
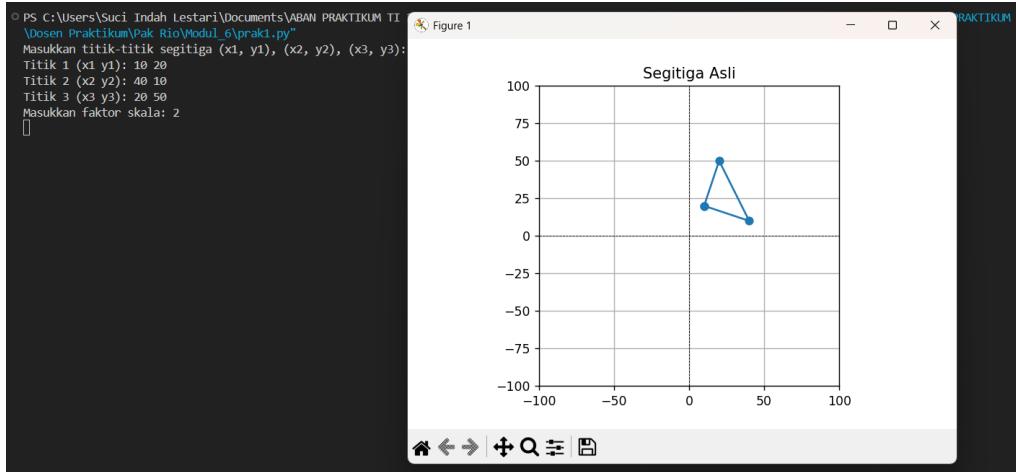
triangle_vertices = [(x1, y1), (x2, y2), (x3, y3)]

# Menggambar segitiga asli
draw_triangle(triangle_vertices, "Segitiga Asli")

# Melakukan penskalaan
scaled_triangle = scale_triangle(triangle_vertices, scale_factor)

# Menggambar segitiga hasil penskalaan
draw_triangle(scaled_triangle, f"Segitiga Setelah Penskalaan (Skala = {scale_factor})")
```

Hasil RUN



Prak 2

Code Program

```
import matplotlib.pyplot as plt # impor modul matplotlib untuk menggambar grafik

def draw_square(vertices, title):
    # fungsi untuk menggambar persegi berdasarkan daftar koordinat vertices
    x, y = zip(*vertices)
    x += (x[0],) # tambahkan titik pertama di akhir agar garis tertutup (kembali ke awal)
    y += (y[0],) # tambahkan titik pertama di akhir agar garis tertutup (kembali ke awal)

    plt.figure() # buat figure baru untuk gambar
    plt.plot(x, y, marker='o') # gambar garis yang menghubungkan titik-titik dan tandai setiap titik
    plt.title(title)
    plt.xlim(-10, 10)
    plt.ylim(-10, 10)
    plt.axhline(0, color='black', linewidth=0.5, ls='--') # gambar garis horizontal sumbu-x
    plt.axvline(0, color='black', linewidth=0.5, ls='--') # gambar garis vertikal sumbu-y
    plt.grid() # tampilkan grid pada plot
    plt.gca().set_aspect('equal', adjustable='box') # pastikan skala x dan y sama (persegi tidak terdistorsi)
    plt.show()

def scale_square(vertices, scale_factor):
    # fungsi untuk mengskalakan setiap titik berdasarkan faktor skala
```

```

scaled_vertices = [(x * scale_factor, y * scale_factor) for x, y in vertices]
# kembalikan daftar koordinat baru setelah dikalikan dengan faktor skala
return scaled_vertices

print("Masukkan titik-titik persegi (x1 y1), (x2 y2), (x3 y3), (x4 y4):")
x1, y1 = map(float, input("Titik 1: ").split())
x2, y2 = map(float, input("Titik 2: ").split())
x3, y3 = map(float, input("Titik 3: ").split())
x4, y4 = map(float, input("Titik 4: ").split())

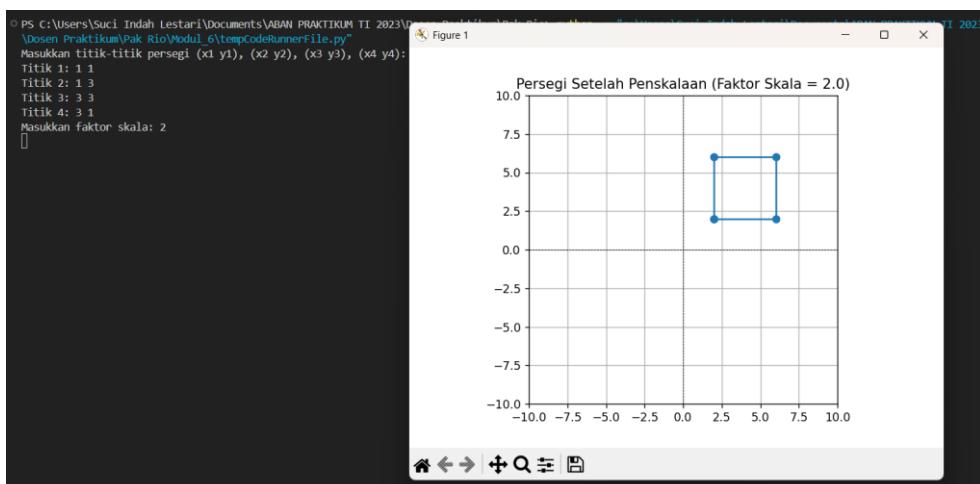
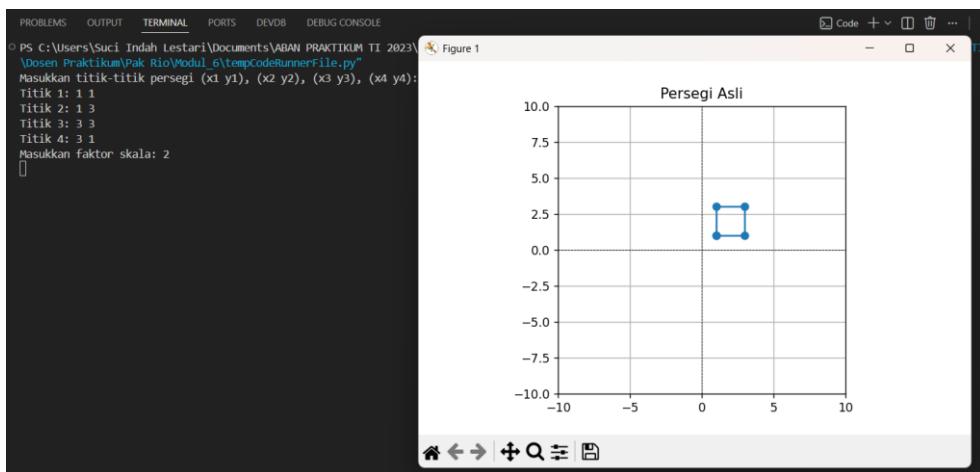
# gabungkan ke dalam daftar vertices sesuai urutan
square_vertices = [(x1, y1), (x2, y2), (x3, y3), (x4, y4)]
# minta input faktor skala dan ubah menjadi float
scale_factor = float(input("Masukkan faktor skala: "))

# gambar persegi asli sebelum penskalaan
draw_square(square_vertices, "Persegi Asli")

# hitung koordinat setelah penskalaan
scaled_square_vertices = scale_square(square_vertices, scale_factor)
# gambar persegi setelah penskalaan dengan judul yang mencantumkan faktor skala
draw_square(scaled_square_vertices, f"Persegi Setelah Penskalaan (Faktor Skala = {scale_factor})")

```

Hasil RUN



Prak 3

Code Program

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

def draw_circle(center, radius, title):
    """Fungsi untuk menggambar lingkaran berdasarkan pusat dan radius."""
    theta = np.linspace(0, 2 * np.pi, 200)
    x = center[0] + radius * np.cos(theta)
    y = center[1] + radius * np.sin(theta)

    plt.figure()
    plt.plot(x, y)
    plt.title(title)

    # Atur batas sumbu agar lingkaran selalu terlihat dengan margin
    margin = 1.2
    xmin = center[0] - radius * margin
    xmax = center[0] + radius * margin
    ymin = center[1] - radius * margin
    ymax = center[1] + radius * margin

    # Pastikan ukuran minimum jendela plotting (untuk kasus radius kecil)
    xmin = min(xmin, -10)
    xmax = max(xmax, 10)
    ymin = min(ymin, -10)
    ymax = max(ymax, 10)

    plt.xlim(xmin, xmax)
    plt.ylim(ymin, ymax)
    plt.axhline(0, color='black', linewidth=0.5, ls='--')
    plt.axvline(0, color='black', linewidth=0.5, ls='--')
    plt.grid()
    plt.gca().set_aspect('equal', adjustable='box')
    plt.show()

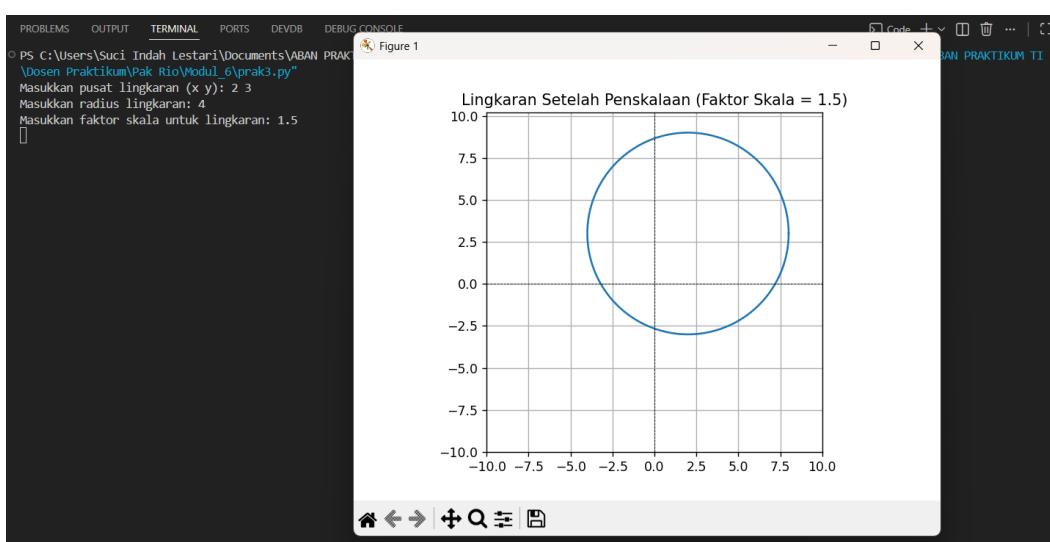
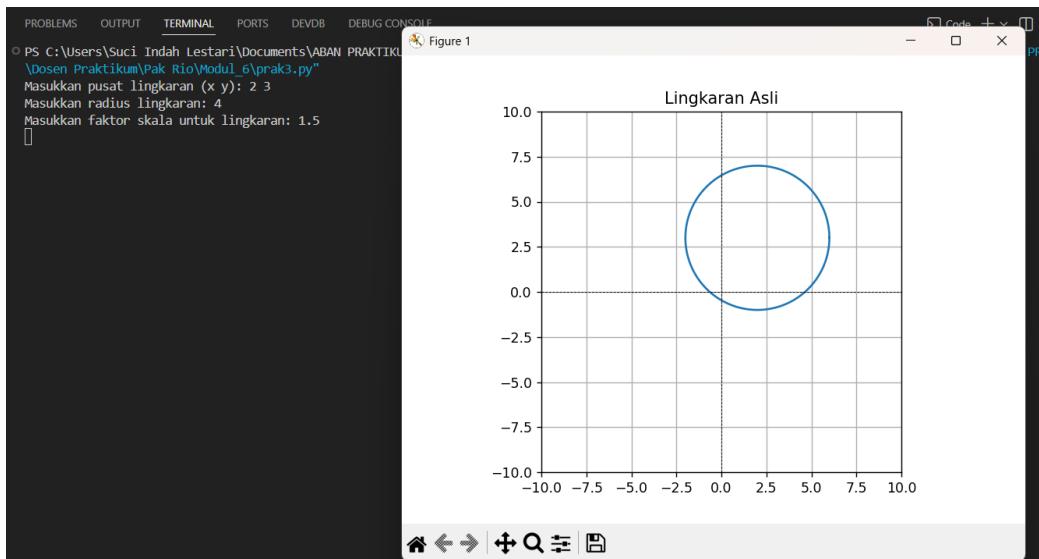
def scale_circle(radius, scale_factor):
    """Fungsi untuk melakukan penskalaan lingkaran berdasarkan faktor skala yang diberikan."""
    return radius * scale_factor

if __name__ == "__main__":
    try:
        center_x, center_y = map(float, input("Masukkan pusat lingkaran (x y): ").split())
        radius = float(input("Masukkan radius lingkaran: "))
        scale_factor = float(input("Masukkan faktor skala untuk lingkaran: "))
    except Exception as e:
        print("Input tidak valid:", e)
        raise SystemExit(1)

    circle_center = (center_x, center_y)
    # Gambar lingkaran asli
    draw_circle(circle_center, radius, "Lingkaran Asli")

    # Lakukan penskalaan dan gambar hasilnya
    new_radius = scale_circle(radius, scale_factor)
    draw_circle(circle_center, new_radius, f'Lingkaran Setelah Penskalaan (Faktor Skala = {scale_factor})')
```

Hasil RUN



Prak 4

Code Program

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

def draw_circle(center, radius, title):
    """Fungsi untuk menggambar lingkaran berdasarkan pusat dan radius."""
    theta = np.linspace(0, 2 * np.pi, 100) # Membuat sudut dari 0 hingga 2π
    x = center[0] + radius * np.cos(theta) # Menghitung koordinat x
    y = center[1] + radius * np.sin(theta) # Menghitung koordinat y

    plt.plot(x, y) # Menggambar lingkaran
    plt.title(title)
    plt.xlim(-10, 10) # Mengatur batas sumbu x
    plt.ylim(-10, 10) # Mengatur batas sumbu y
    plt.axhline(0, color='black', linewidth=0.5, ls='--') # Garis horizontal di sumbu x
    plt.axvline(0, color='black', linewidth=0.5, ls='--') # Garis vertikal di sumbu y
    plt.grid(True) # Menampilkan grid
    plt.gca().set_aspect('equal', adjustable='box') # Skala sama pada x dan y

def draw_rectangle(bottom_left, width, height, title):
```

```

"""Fungsi untuk menggambar persegi panjang berdasarkan titik kiri bawah, lebar, dan tinggi."""
x = [bottom_left[0], bottom_left[0] + width, bottom_left[0] + width, bottom_left[0], bottom_left[0]]
y = [bottom_left[1], bottom_left[1], bottom_left[1] + height, bottom_left[1] + height, bottom_left[1]]

plt.plot(x, y) # Menggambar persegi panjang
plt.title(title)
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(True)
plt.gca().set_aspect('equal', adjustable='box')

def scale_circle(center, radius, scale_factor):
    """Fungsi untuk melakukan penskalaan lingkaran berdasarkan faktor skala yang diberikan."""
    new_radius = radius * scale_factor
    return new_radius

def scale_rectangle(bottom_left, width, height, scale_factor):
    """Fungsi untuk melakukan penskalaan persegi panjang berdasarkan faktor skala."""
    new_width = width * scale_factor
    new_height = height * scale_factor
    return new_width, new_height

# INPUT DATA
circle_center_x, circle_center_y = map(float, input("Masukkan pusat lingkaran (x y): ").split())
circle_radius = float(input("Masukkan radius lingkaran: "))
scale_factor_circle = float(input("Masukkan faktor skala untuk lingkaran: "))

rect_bottom_left_x, rect_bottom_left_y = map(float, input("Masukkan titik kiri bawah persegi panjang (x y): ").split())
rect_width = float(input("Masukkan lebar persegi panjang: "))
rect_height = float(input("Masukkan tinggi persegi panjang: "))
scale_factor_rectangle = float(input("Masukkan faktor skala untuk persegi panjang: "))
circle_center = (circle_center_x, circle_center_y)
rect_bottom_left = (rect_bottom_left_x, rect_bottom_left_y)

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
draw_circle(circle_center, circle_radius, "Lingkaran Asli")

# HASIL PENSKALAAN
new_circle_radius = scale_circle(circle_center, circle_radius, scale_factor_circle)
draw_circle(circle_center, new_circle_radius, f'Lingkaran Setelah Penskalaan (Faktor Skala: {scale_factor_circle})')

plt.subplot(1, 2, 2)
draw_rectangle(rect_bottom_left, rect_width, rect_height, "Persegi Panjang Asli")

new_width, new_height = scale_rectangle(rect_bottom_left, rect_width, rect_height, scale_factor_rectangle)
draw_rectangle(rect_bottom_left, new_width, new_height, f'Persegi Panjang Setelah Penskalaan (Faktor Skala: {scale_factor_rectangle})')

```

```
plt.tight_layout()  
plt.show()
```

Hasil RUN

The figure consists of two side-by-side plots on a coordinate system with x and y axes ranging from -10.0 to 10.0. The left plot shows a blue circle centered at (0, 0) with a radius of 2.0, and an orange concentric circle with a radius of 2.0. The right plot shows a blue rectangle with vertices at (0, 0), (2, 0), (2, 2), and (0, 2), and an orange larger rectangle with vertices at (-1, 0), (3, 0), (3, 2), and (-1, 2), representing a scaling factor of 1.5.

Tugas

1. Buatlah Code program sederhana untuk menggambar tabung

Code Program

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# =====
# FUNGSI UNTUK MENGGAMBAR TABUNG
# =====
def draw_cylinder(radius, height, title, ax):
    """
    Menggambar tabung 3D menggunakan jaring-jaring mesh.
    radius : jari-jari tabung
    height : tinggi tabung
    title : judul tampilan
    ax   : axes untuk plot 3D
    """

    # Membuat grid theta (sudut) dan z (tinggi)
    z = np.linspace(0, height, 50)
    theta = np.linspace(0, 2 * np.pi, 50)
    theta, z = np.meshgrid(theta, z)

    # Rumus koordinat silinder
    x = radius * np.cos(theta)
    y = radius * np.sin(theta)

    # Plot permukaan tabung
    ax.plot_surface(x, y, z, color='gray', edgecolor='black', linewidth=0.3)

    # Judul di tengah
    ax.set_title(title, pad=20, fontsize=12)

    # Mengatur batas sumbu agar proporsional
    ax.set_xlim([-10, 10])
    ax.set_ylim([-10, 10])
    ax.set_zlim([0, 10])

    # Label sumbu
    ax.set_xlabel("X", labelpad=10)
    ax.set_ylabel("Y", labelpad=10)
    ax.set_zlabel("Z", labelpad=10)

    # Grid menyala
    ax.grid(True)

    # =====
    # INPUT DATA DARI USER
    # =====
    radius = float(input("Masukkan radius tabung : "))
    height = float(input("Masukkan tinggi tabung : "))
    scale_factor = float(input("Masukkan faktor penskalaan :"))

    # Hasil skala
    new_radius = radius * scale_factor
```

```

new_height = height * scale_factor

# =====
# MENAMPILKAN GAMBAR
# =====
fig = plt.figure(figsize=(13, 6))

# Plot tabung asli
ax1 = fig.add_subplot(1, 2, 1, projection='3d')
draw_cylinder(radius, height, "Tabung Asli", ax1)

# Plot tabung setelah penskalaan
ax2 = fig.add_subplot(1, 2, 2, projection='3d')
draw_cylinder(new_radius, new_height,
              f"Tabung Setelah Penskalaan\n(Faktor Skala: {scale_factor})", ax2)

# Mengatur tata letak
plt.tight_layout()
plt.show()

```

Hasil RUN

