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

Modul 6

Nama : Muhammad Rizal Nurfirdaus

NIM : 20230810088

Kelas : TINFC-2023-04

TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS KUNINGAN

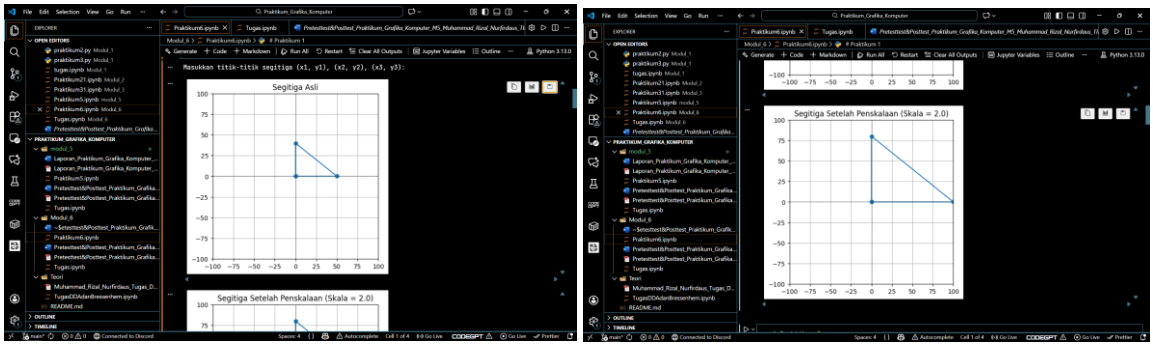
Praktikum

1. Praktikum 1 : Penskalaan Objek Segitiga
   Source Code:

```
# Praktikum 1
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:



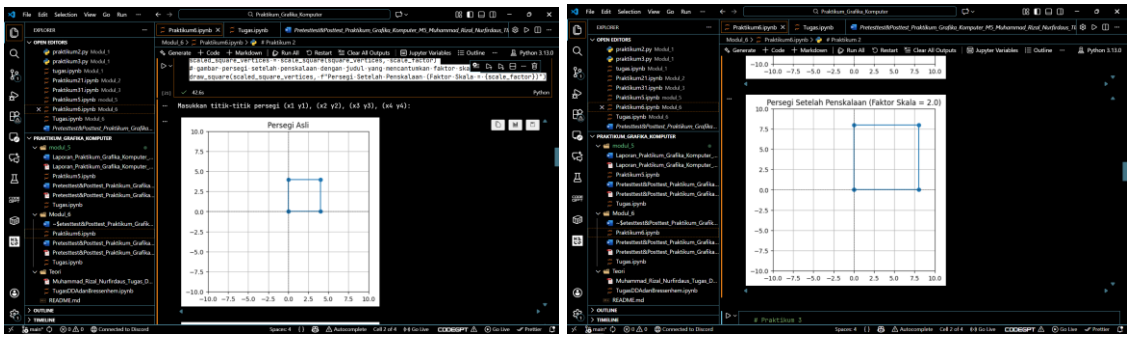
Analisis: Kode yang digunakan berfungsi untuk menggambar sebuah segitiga berdasarkan tiga titik input yang diberikan oleh pengguna. Proses penggambaran dilakukan dengan memecah koordinat menjadi nilai x dan y, lalu menghubungkannya hingga kembali ke titik awal sehingga terbentuk bentuk segitiga yang tertutup. Program juga menyediakan fungsi penskalaan, yaitu mengalikan setiap titik dengan faktor skala yang dimasukkan, sehingga ukuran segitiga dapat diperbesar atau diperkecil tanpa mengubah bentuk dasarnya. Secara keseluruhan, kode ini memperkenalkan konsep dasar transformasi geometri dalam grafika komputer, terutama bagaimana objek 2D dapat digambar dan dimodifikasi melalui perubahan koordinat matematis.

2. Praktikum 2 : Penskalaan Objek Persegi
   Source Code:

```
# Praktikum 2
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)  # pisahkan koordinat x dan y dari daftar
tupel 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)  # beri judul pada grafik
    plt.xlim(-10, 10)  # atur batas sumbu-x agar konsisten
    plt.ylim(-10, 10)  # atur batas sumbu-y agar konsisten
    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()  # tampilkan jendela plot

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):")
# minta input titik 1, ubah ke float dan pisahkan menjadi x1, y1
x1, y1 = map(float, input("Titik 1: ").split())
# minta input titik 2, ubah ke float dan pisahkan menjadi x2, y2
x2, y2 = map(float, input("Titik 2: ").split())
# minta input titik 3, ubah ke float dan pisahkan menjadi x3, y3
x3, y3 = map(float, input("Titik 3: ").split())
# minta input titik 4, ubah ke float dan pisahkan menjadi x4, y4
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:



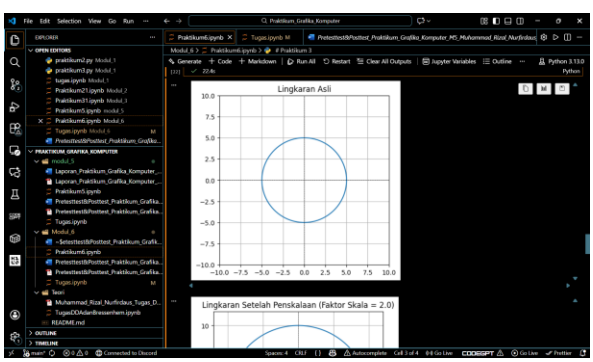
Analisis: Program digunakan untuk menggambar sebuah persegi berdasarkan empat titik yang dimasukkan oleh pengguna. Kode dimulai dengan memisahkan koordinat x dan y, kemudian menutup bentuk persegi dengan menambahkan titik pertama ke urutan terakhir. Fungsi penskalaan bekerja dengan mengalikan setiap koordinat titik dengan faktor skala sehingga ukuran persegi dapat diperbesar atau diperkecil secara proporsional. Program juga mengatur batas sumbu dan aspek rasio agar bentuk persegi tidak terdistorsi saat ditampilkan. Secara keseluruhan, praktikum ini memperlihatkan bagaimana objek 2D dapat direpresentasikan melalui koordinat dan dimanipulasi menggunakan transformasi geometri sederhana berupa scaling.

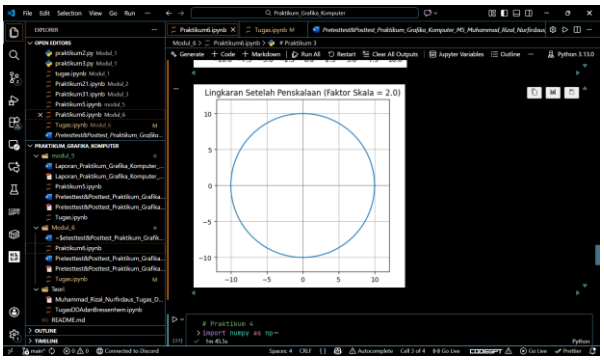
3. Praktikum 3 : Pensklaan Objek Lingkaran
   Source Code:

```
# Praktikum 3
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 :





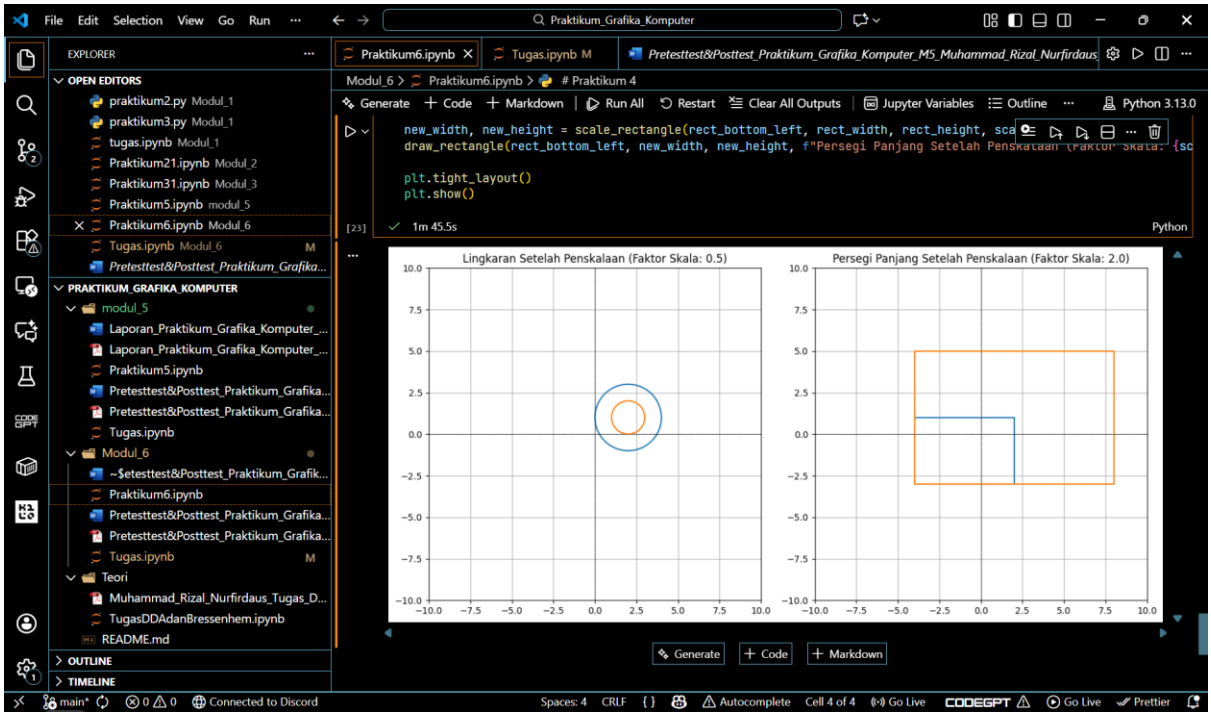
Analisis: Program digunakan untuk menggambar lingkaran berdasarkan koordinat pusat dan nilai radius yang dimasukkan pengguna. Fungsi menggambar lingkaran memanfaatkan rumus parametrik menggunakan nilai theta untuk menghasilkan koordinat x dan y. Program juga mengatur batas sumbu secara dinamis agar lingkaran selalu tampil penuh tanpa terpotong. Proses penskalaan dilakukan dengan sederhana, yaitu mengalikan radius asli dengan faktor skala sehingga ukuran lingkaran dapat membesar atau mengecil secara proporsional. Secara keseluruhan, praktikum ini menunjukkan bagaimana objek melingkar dapat divisualisasikan dan dimodifikasi ukurannya melalui transformasi scaling yang diterapkan langsung pada nilai radius.

4. Praktikum 4 : Penskalaan Objek Lingkaran dan Persegi Panjang

```
# Praktikum 4
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 :

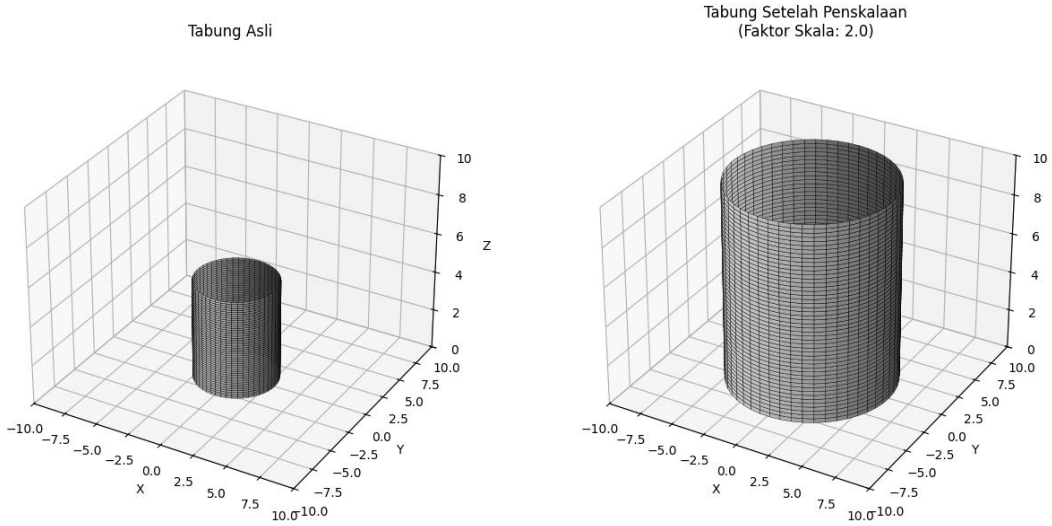


Analisis :

- **Fungsi draw_circle**
  - Menghitung titik-titik lingkaran menggunakan rumus parametris $x = x_c + r\cos\theta, y = y_c + r\sin\theta$.
  - Menampilkan lingkaran pada grafik dengan sumbu, grid, dan aspect ratio yang sama.
    Intinya: menggambar bentuk lingkaran berdasar pusat & radius.
- **Fungsi draw_rectangle**
  - Menghitung 4 titik sudut persegi panjang dari titik kiri bawah, lebar, dan tinggi.
  - Menghubungkan titik-titik tersebut menjadi bentuk persegi panjang.
    Intinya: menggambar persegi panjang dengan parameter dasar.
- **Fungsi scale_circle**
  - Melakukan penskalaan dengan mengalikan radius lama dengan faktor skala.
  - Menghasilkan radius baru sebagai output.
    Intinya: scaling lingkaran hanya mengubah radius.
- **Fungsi scale_rectangle**
  - Mengalikan lebar dan tinggi persegi panjang dengan faktor skala.
  - Menghasilkan ukuran baru.
    Intinya: scaling persegi panjang memperbesar/diperkecil dimensinya secara proporsional.
- **Bagian Input & Visualisasi**
  - Menerima 9 input.
  - Menggambar bangun asli.
  - Menggambar bangun setelah penskalaan di subplot yang sama.
    Intinya: membandingkan bentuk sebelum dan sesudah scaling dalam satu tampilan.

LATIHAN/TUGAS

1. Buatlah kode program sederhana untuk menggambar tabung seperti gambar berikut:

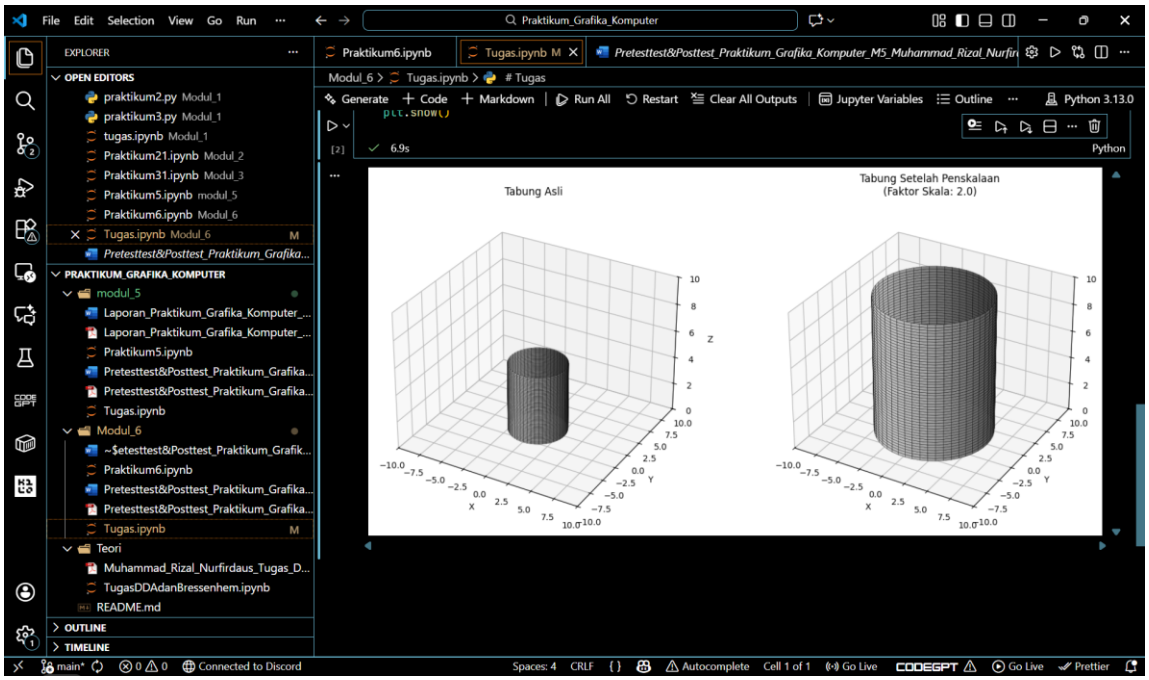


Source Code:

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

# ============================================================
# FUNGSI MENGGAMBAR TABUNG
# ============================================================
def draw_cylinder(r, h, title, ax):
    """Membangun model tabung 3D berdasarkan nilai jari-jari dan tinggi."""

    # Menyusun titik-titik sudut melingkar dan ketinggian tabung
    z_values = np.linspace(0, h, 50)
    angle_values = np.linspace(0, 2 * np.pi, 50)
    theta, z = np.meshgrid(angle_values, z_values)

    # Menghasilkan koordinat permukaan tabung
    x = r * np.cos(theta)
    y = r * np.sin(theta)

    # Menampilkan bentuk tabung pada axes 3D
    ax.plot_surface(
        x, y, z,
        color="lightgray",
        edgecolor="black",
        linewidth=0.3
    )

    # Pengaturan tampilan plot
    ax.set_title(title, pad=15)
    ax.set_xlim(-10, 10)
    ax.set_ylim(-10, 10)
    ax.set_zlim(0, 10)
```

```
    ax.set_xlabel("X")
    ax.set_ylabel("Y")
    ax.set_zlabel("Z")

    ax.grid(True)


# ============================================================
# INPUT PENGGUNA
# ============================================================
r_input = float(input("Masukkan radius tabung  : "))
h_input = float(input("Masukkan tinggi tabung  : "))
scale = float(input("Masukkan faktor skala   : "))

# Perhitungan hasil penskalaan
scaled_r = r_input * scale
scaled_h = h_input * scale


# ============================================================
# PEMBUATAN PLOT
# ============================================================
fig = plt.figure(figsize=(13, 6))

# Gambar tabung asli
ax1 = fig.add_subplot(1, 2, 1, projection="3d")
draw_cylinder(r_input, h_input, "Tabung Asli", ax1)

# Gambar tabung hasil pembesaran/pengurangan
ax2 = fig.add_subplot(1, 2, 2, projection="3d")
draw_cylinder(
    scaled_r,
    scaled_h,
    f"Tabung Setelah Penskalaan\n(Faktor Skala: {scale})",
    ax2
)

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

Hasil Run :



Analisis: Program tersebut bekerja dengan cara meminta tiga input utama yaitu radius, tinggi tabung, dan faktor skala yang kemudian digunakan untuk menghitung ukuran tabung baru setelah proses penskalaan. Fungsi draw_cylinder membangun tabung 3D menggunakan pendekatan parametrik melalui grid sudut (theta) dan ketinggian (z) sehingga menghasilkan koordinat permukaan x = r*cos(theta) dan y = r*sin(theta). Dua visualisasi ditampilkan berdampingan: tabung asli di sebelah kiri dan tabung hasil penskalaan di sebelah kanan. Ketika faktor skala diterapkan, baik radius maupun tinggi tabung dikalikan nilai skala tersebut, sehingga ukuran tabung tampak lebih besar atau lebih kecil sesuai nilai yang diberikan pengguna. Secara matematis, perubahan ukuran ini bersifat linear pada dimensi (r dan h), namun secara volumetrik perubahan terjadi sebesar faktor pangkat tiga terhadap skala. Visualisasi sudah menampilkan bentuk tabung dengan benar, meskipun masih terdapat beberapa keterbatasan seperti tidak adanya penutup atas/bawah, batas sumbu yang statis, dan aspek rasio 3D yang belum sepenuhnya akurat. Secara keseluruhan, program mampu menunjukkan perbandingan bentuk tabung sebelum dan sesudah penskalaan dengan jelas dan efektif.