

# **LAPORAN PRAKTIKUM**

## **GRAFIKA KOMPUTER**

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

### **Modul 8**



**DISUSUN OLEH :**

**NAMA: MOHAMAD ABAN SY'BANA**

**NIM : 20230810012**

**KELAS : TINFC-2023-04**

**TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS KUNINGAN**

**2025**

## PRAKTIKUM

### 1. Prak 1

#### Code Program

```
import matplotlib.pyplot as plt

# Fungsi untuk menggambar segitiga
def gambar_segitiga(titik):
    # Menyusun titik segitiga
    x = [titik[0][0], titik[1][0], titik[2][0], titik[0][0]]
    y = [titik[0][1], titik[1][1], titik[2][1], titik[0][1]]

    # Menggambar segitiga
    plt.plot(x, y, marker='o')
    plt.fill(x, y, alpha=0.3) # Mengisi segitiga dengan warna transparan

# Fungsi untuk melakukan pencerminan
def pencerminan(titik, sumbu):
    # Melakukan pencerminan berdasarkan sumbu
    if sumbu == 'x':
        # Pencerminan terhadap sumbu x
        return [(x, -y) for (x, y) in titik]
    elif sumbu == 'y':
        # Pencerminan terhadap sumbu y
        return [(-x, y) for (x, y) in titik]
    elif sumbu == 'origin':
        # Pencerminan terhadap origin
        return [(-x, -y) for (x, y) in titik]
    else:
        raise ValueError("Sumbu tidak valid. Pilih 'x', 'y', atau 'origin'.")

# Input titik segitiga
titik_segitiga = []
print("Masukkan koordinat titik segitiga (format: x,y):")
for i in range(3):
    koordinat = input(f"Titik {i+1}: ")
    x, y = map(float, koordinat.split(','))
    titik_segitiga.append((x, y))

# Input jenis sumbu pencerminan
sumbu = input("Masukkan sumbu pencerminan ('x', 'y', atau 'origin'): ")

# Menggambar segitiga asli
plt.figure(figsize=(8, 8))
plt.title("Pencerminan Segitiga 2D")
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)

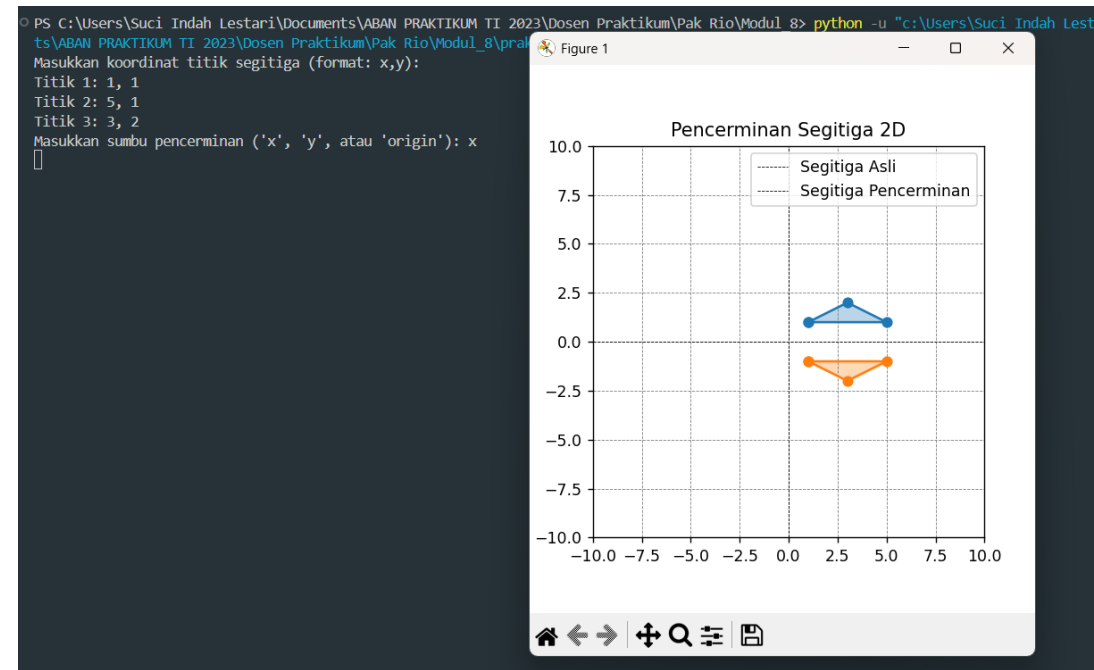
# Gambar segitiga asli
gambar_segitiga(titik_segitiga)

# Pencerminan segitiga
titik_segitiga_pencerminan = pencerminan(titik_segitiga, sumbu)

# Menggambar segitiga setelah pencerminan
gambar_segitiga(titik_segitiga_pencerminan)
```

```
# Menambahkan legend dan menampilkan plot
plt.legend(['Segitiga Asli', 'Segitiga Pencerminkan'])
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.gca().set_aspect('equal', adjustable='box')
plt.show()
```

Hasil RUN



## 2. Prak 2

### Code Program

```
import matplotlib.pyplot as plt

# Fungsi untuk menggambar persegi panjang
def gambar_persegi_panjang(ax, x, y, lebar, tinggi, label="Persegi Panjang"):
    persegi_panjang = plt.Rectangle(
        (x, y),
        lebar,
        tinggi,
        fill=False,
        edgecolor='blue',
        linewidth=2,
        label=label
    )
    ax.add_patch(persegi_panjang)

# Fungsi untuk mencerminkan persegi panjang
def pencerminan_persegi_panjang(x, y, lebar, tinggi, sumbu):
    if sumbu == 'X':
        # Pencerminan terhadap sumbu X
        return x, -y - tinggi, lebar, tinggi
    elif sumbu == 'Y':
        # Pencerminan terhadap sumbu Y
        return -x - lebar, y, lebar, tinggi
    elif sumbu == 'ORIGIN':
        # Pencerminan terhadap titik origin
        return -x - lebar, -y - tinggi, lebar, tinggi
    else:
        raise ValueError("Sumbu tidak valid. Gunakan X, Y, atau ORIGIN.")

# ===== INPUT DATA =====
x = float(input("Masukkan koordinat x sudut kiri bawah: "))
y = float(input("Masukkan koordinat y sudut kiri bawah: "))
lebar = float(input("Masukkan lebar persegi panjang: "))
tinggi = float(input("Masukkan tinggi persegi panjang: "))
sumbu = input("Masukkan sumbu pencerminan (X, Y, atau ORIGIN): ").strip().upper()

# ===== PLOT =====
fig, ax = plt.subplots(figsize=(8, 8))

# Persegi panjang asli
gambar_persegi_panjang(ax, x, y, lebar, tinggi, label="Persegi Panjang Asli")

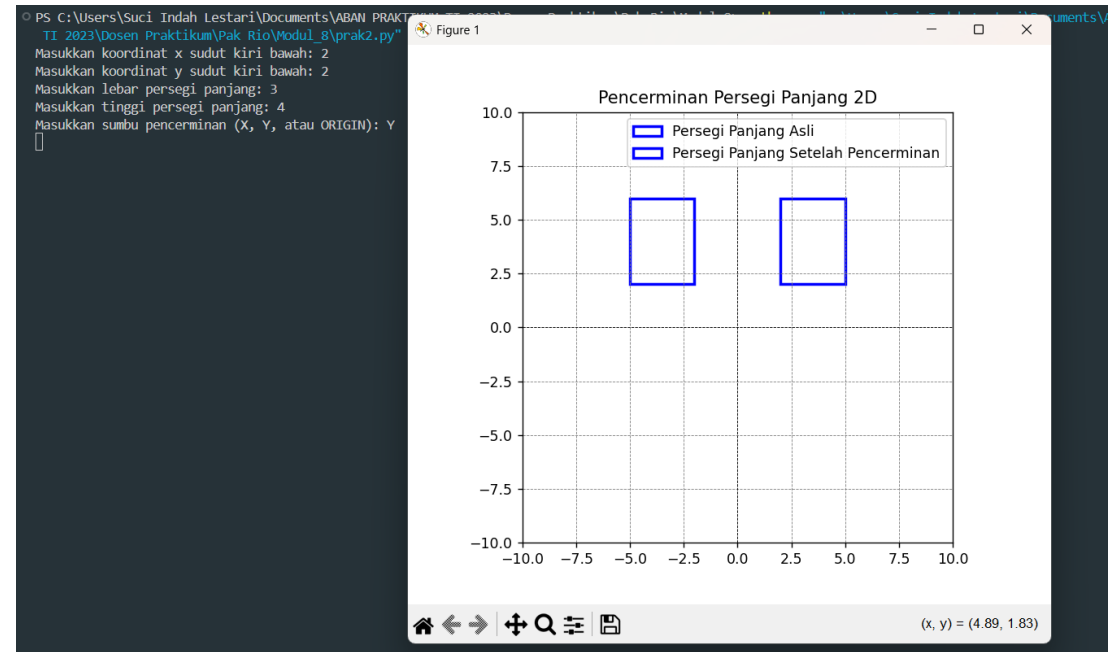
# Persegi panjang hasil pencerminan
x_baru, y_baru, lebar_baru, tinggi_baru = pencerminan_persegi_panjang(
    x, y, lebar, tinggi, sumbu
)
gambar_persegi_panjang(
    ax,
    x_baru,
    y_baru,
    lebar_baru,
    tinggi_baru,
    label="Persegi Panjang Setelah Pencerminan"
)
```

```
# ===== DETAIL PLOT =====
ax.set_xlim(-10, 10)
ax.set_ylim(-10, 10)
ax.set_aspect('equal', adjustable='box')

# Sumbu X dan Y
ax.axhline(0, color='black', linewidth=0.5, ls='--')
ax.axvline(0, color='black', linewidth=0.5, ls='--')

# Grid dan keterangan
plt.grid(color='gray', linestyle='--', linewidth=0.5)
plt.legend()
plt.title("Pencerminan Persegi Panjang 2D")
plt.show()
```

Hasil RUN



### 3. Prak 3

#### Code Program

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

# Fungsi untuk menggambar lingkaran
def gambar_lingkaran(ax, x, y, radius, label='Lingkaran'):
    """
    Fungsi untuk menggambar lingkaran pada grafik
    """
    lingkaran = plt.Circle(
        (x, y),
        radius,
        fill=False,
        edgecolor='blue',
        linewidth=2,
        label=label
    )
    ax.add_patch(lingkaran)

# Fungsi untuk mencerminkan lingkaran
def pencerminan_lingkaran(x, y, radius, sumbu):
    """
    Fungsi untuk melakukan pencerminan lingkaran
    """
    if sumbu == 'X':
        # Pencerminan terhadap sumbu X
        return x, -y, radius
    elif sumbu == 'Y':
        # Pencerminan terhadap sumbu Y
        return -x, y, radius
    elif sumbu == 'ORIGIN':
        # Pencerminan terhadap titik origin
        return -x, -y, radius
    else:
        raise ValueError("Sumbu tidak valid. Pilih X, Y, atau ORIGIN.")

# ===== INPUT DARI PENGGUNA =====
x = float(input("Masukkan koordinat X pusat lingkaran: "))
y = float(input("Masukkan koordinat Y pusat lingkaran: "))
radius = float(input("Masukkan radius lingkaran: "))
sumbu = input("Pilih sumbu pencerminan (X, Y, atau Origin): ").strip().upper()

# ===== MEMBUAT PLOT =====
fig, ax = plt.subplots(figsize=(8, 8))

# Menggambar lingkaran asli
gambar_lingkaran(ax, x, y, radius, label='Lingkaran Asli')

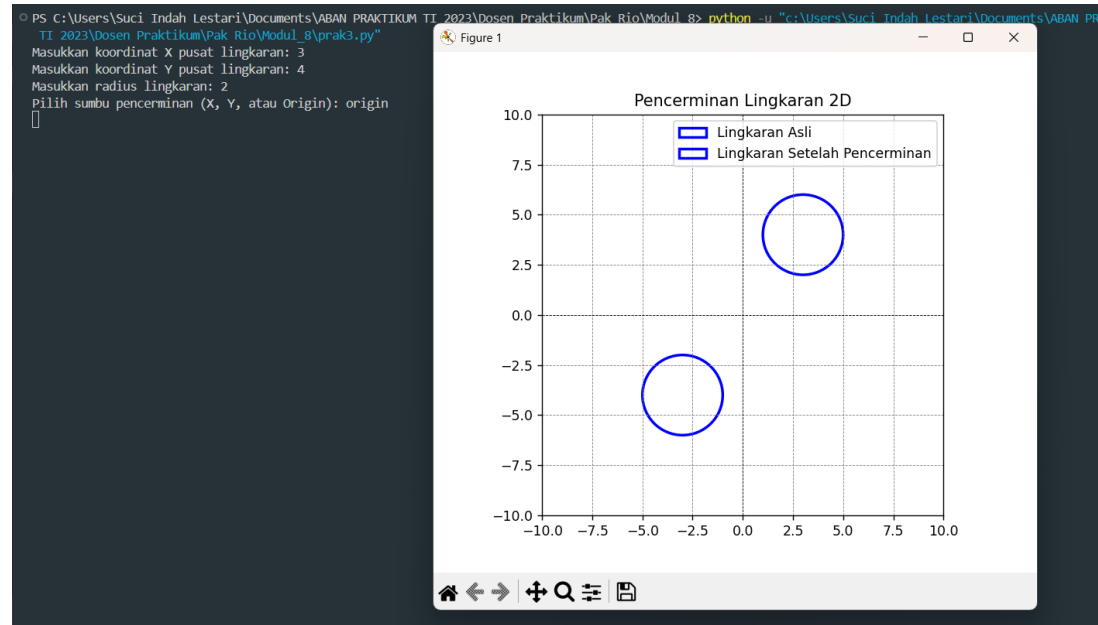
# Menggambar lingkaran setelah pencerminan
x_baru, y_baru, radius_baru = pencerminan_lingkaran(x, y, radius, sumbu)
gambar_lingkaran(
    ax,
    x_baru,
    y_baru,
    radius_baru,
    label='Lingkaran Setelah Pencerminan'
)
```

```
# ===== DETAIL PLOT =====
ax.set_xlim(-10, 10)
ax.set_ylim(-10, 10)
ax.set_aspect('equal', adjustable='box')

# Sumbu X dan Y
ax.axhline(0, color='black', linewidth=0.5, ls='--')
ax.axvline(0, color='black', linewidth=0.5, ls='--')

# Grid, legend, dan judul
plt.grid(color='gray', linestyle='--', linewidth=0.5)
plt.legend()
plt.title('Pencerminan Lingkaran 2D')
plt.show()
```

Hasil RUN



## Tugas

1. **Buatlah kode program sederhana untuk menggambar objek belahketupat yang di cerminkan terhadap sumbu X.**

### Code Program

```
import matplotlib.pyplot as plt

# Input titik belah ketupat dari pengguna
titik_belah_ketupat_asli = []
print("Masukkan koordinat belah ketupat (format: x,y). Harus 4 titik:")
for i in range(4):
    x, y = map(float, input(f"Titik {i+1}: ").split(","))
    titik_belah_ketupat_asli.append([x, y])

# Pilih sumbu pencerminan
sumbu = input("Masukkan sumbu pencerminan (x, y, origin): ").strip().lower()

# Fungsi pencerminan umum
def pencerminan(titik_asli, sumbu):
    if sumbu == 'x':
        return [[x, -y] for x, y in titik_asli]
    elif sumbu == 'y':
        return [[-x, y] for x, y in titik_asli]
    elif sumbu == 'origin':
        return [[-x, -y] for x, y in titik_asli]
    else:
        raise ValueError("Sumbu tidak valid. Pilih 'x', 'y', atau 'origin'.")

# Terapkan pencerminan
titik_belah_ketupat_pencerminan = pencerminan(titik_belah_ketupat_asli,
sumbu)

# Input offset opsional
offset_y = float(input("Masukkan offset Y (misalnya 0.9): "))
offset_x = float(input("Masukkan offset X (misalnya 0): "))
titik_belah_ketupat_pencerminan = [[x + offset_x, y + offset_y] for x, y in
titik_belah_ketupat_pencerminan]

# Fungsi menggambar objek
def gambar_objek(titik, label_teks, warna, z_order):
    x_coords = [p[0] for p in titik] + [titik[0][0]]
    y_coords = [p[1] for p in titik] + [titik[0][1]]
    plt.fill(x_coords, y_coords, color=warna, alpha=0.9, label=label_teks,
zorder=z_order)

# Plot
plt.figure(figsize=(8, 8))
plt.title("Pencerminan Belah Ketupat 2D")

plt.axhline(0, color='black', linewidth=0.8)
plt.axvline(0, color='black', linewidth=0.8)
plt.grid(color='gray', linestyle='-', linewidth=0.5, alpha=0.5)

gambar_objek(titik_belah_ketupat_asli, 'Belah Ketupat Asli', '#555555', 1)
gambar_objek(titik_belah_ketupat_pencerminan, 'Belah Ketupat Setelah
Pencerminan', '#778899', 2)

plt.xlim(-10, 10)
plt.ylim(-10, 10)
```



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

plt.legend(loc='upper right')
plt.show()
```

Hasil RUN

