

**LAPORAN PRAKTIKUM**  
**GRAFIKA KOMPUTER**  
**(DOSEN PENGAMPU : Rio Priantama, S.T., M.T.I)**  
**Modul 5**



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

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

## PRAKTIKUM

### Praktikum 1 : Translasi Objek Segitiga

#### Code Program

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

# Fungsi untuk menggambar segitiga
def gambar_segitiga(titik, warna='b', label=None):
    segitiga = np.array(titik)
    # Tambahkan titik pertama ke akhir untuk menutup segitiga
    segitiga = np.vstack([segitiga, segitiga[0]])
    plt.plot(segitiga[:, 0], segitiga[:, 1], color=warna, label=label)

# Fungsi untuk melakukan translasi
def translasi(titik, vektor):
    titik_baru = [(x + vektor[0], y + vektor[1]) for x, y in titik]
    return titik_baru

# Titik-titik awal segitiga
segitiga_awal = [(1, 2), (3, 5), (6, 2)]

# Vektor translasi
vektor_translasi = (2, 3)

# Translasi segitiga
segitiga_tertranslasi = translasi(segitiga_awal, vektor_translasi)

# Plot segitiga sebelum dan sesudah translasi
plt.figure()

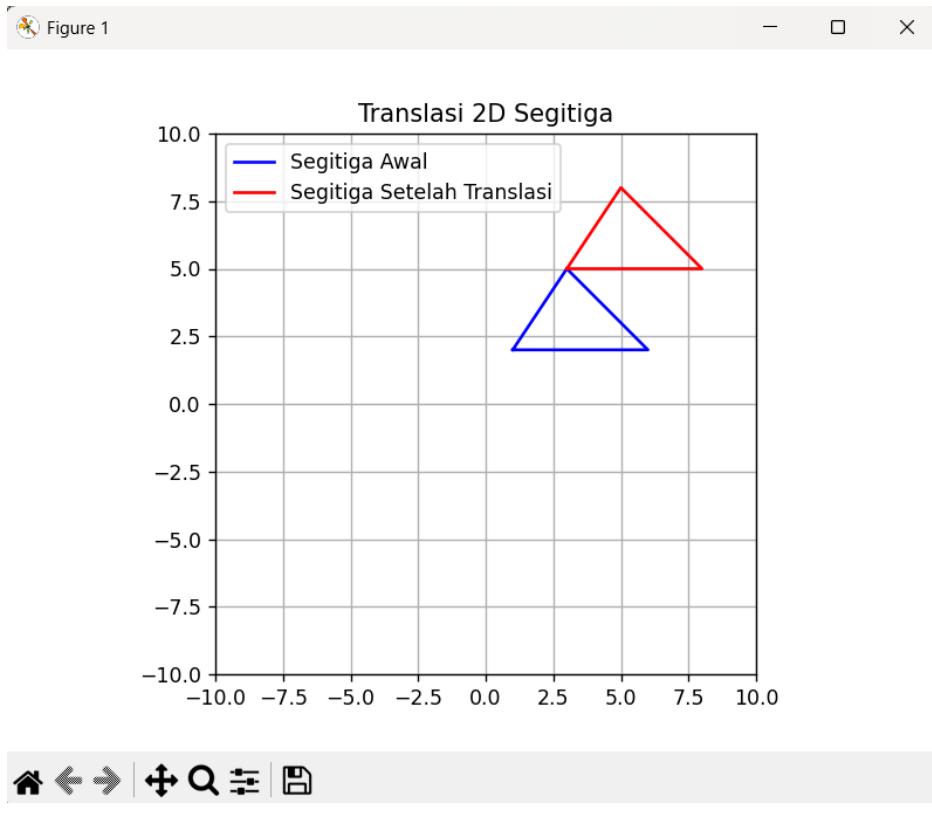
# Gambar segitiga awal
gambar_segitiga(segitiga_awal, warna='b', label='Segitiga Awal')

# Gambar segitiga setelah translasi
gambar_segitiga(segitiga_tertranslasi, warna='r', label='Segitiga Setelah
Translasi')

# Menampilkan grid dan setting sumbu
plt.grid(True)
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
plt.title('Translasi 2D Segitiga')

# Tampilkan plot
plt.show()
```

## Hasil RUN



## Praktikum 2 : Translasi Objek Segitiga dengan input dinamis

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

# Fungsi untuk menggambar segitiga
def gambar_segitiga(titik, warna='b', label=None):
    segitiga = np.array(titik)
    # Tambahkan titik pertama ke akhir untuk menutup segitiga
    segitiga = np.vstack([segitiga, segitiga[0]])
    plt.plot(segitiga[:, 0], segitiga[:, 1], color=warna, label=label)

# Fungsi untuk melakukan translasi
def translasi(titik, vektor):
    dx, dy = vektor
    titik_baru = [(x + dx, y + dy) for x, y in titik]
    return titik_baru

# Fungsi utama untuk input dinamis dan translasi
def main():
    # Input titik segitiga dari pengguna
    print("Masukkan koordinat segitiga (dalam format x,y):")
    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(','))

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

```

# Input vektor translasi
print("\nMasukkan vektor translasi (dalam format dx,dy):")
dx, dy = map(float, input("Vektor Translasi (dx, dy): ").split(','))

# Translasi segitiga
segitiga_tertranslasi = translasi(segitiga_awal, (dx, dy))

# Plot segitiga sebelum dan sesudah translasi
plt.figure()

# Gambar segitiga awal
gambar_segitiga(segitiga_awal, warna='b', label='Segitiga Awal')

# Gambar segitiga setelah translasi
gambar_segitiga(segitiga_tertranslasi, warna='r', label='Segitiga Setelah Translasi')

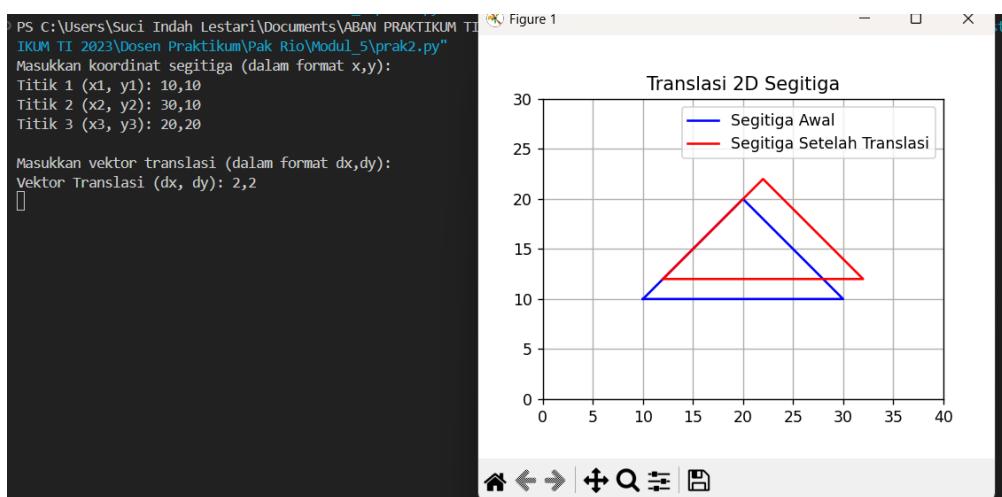
# Menampilkan grid dan setting sumbu
plt.grid(True)
plt.xlim(min(x1, x2, x3) - 10, max(x1, x2, x3) + 10)
plt.ylim(min(y1, y2, y3) - 10, max(y1, y2, y3) + 10)
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
plt.title('Translasi 2D Segitiga')

plt.show()

# Panggil fungsi utama
main()

```

### Hasil RUN



### Praktikum 3 : Translasi Objek Persegi Panjang dengan input dinamis

```

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

# Fungsi untuk menggambar persegi panjang
def gambar_persegi_panjang(titik, warna='b', label=None):
    persegi_panjang = np.array(titik)

```

```

# Tambahkan titik pertama ke akhir untuk menutup persegi panjang
persegi_panjang = np.vstack([persegi_panjang, persegi_panjang[0]])
plt.plot(persegi_panjang[:, 0], persegi_panjang[:, 1], color=warna, label=label)

# Fungsi untuk melakukan translasi
def translasi(titik, vektor):
    # Menambahkan vektor translasi ke semua titik persegi panjang
    titik_baru = [[x + vektor[0], y + vektor[1]] for x, y in titik]
    return titik_baru

# Fungsi utama untuk input dinamis dan translasi
def main():
    # Input titik persegi panjang dari pengguna
    print("Masukkan koordinat persegi panjang (dalam format x, y):")
    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(',')) 
    x4, y4 = map(float, input("Titik 4 (x4, y4): ").split(','))

    # Titik persegi panjang awal
    persegi_panjang_awal = [(x1, y1), (x2, y2), (x3, y3), (x4, y4)]

    # Input vektor translasi dari pengguna
    print("Masukkan vektor translasi (dalam format dx, dy):")
    dx, dy = map(float, input("Vektor Translasi (dx, dy): ").split(','))

    # Translasi persegi panjang
    persegi_panjang_tertranslasi = translasi(persegi_panjang_awal, (dx, dy))

    # Plot persegi panjang sebelum dan sesudah translasi
    plt.figure()

    # Gambar persegi panjang awal
    gambar_persegi_panjang(persegi_panjang_awal, warna='b', label='Persegi Panjang Awal')

    # Gambar persegi panjang setelah translasi
    gambar_persegi_panjang(persegi_panjang_tertranslasi, warna='r', label='Persegi Panjang Setelah Translasi')

    # Menampilkan grid dan setting sumbu
    plt.grid(True)

    # Mengumpulkan semua koordinat x dan y untuk mengatur batas plot
    semua_x = [x1, x2, x3, x4] + [titik[0] for titik in persegi_panjang_tertranslasi]
    semua_y = [y1, y2, y3, y4] + [titik[1] for titik in persegi_panjang_tertranslasi]

    plt.xlim(min(semua_x) - 10, max(semua_x) + 10)
    plt.ylim(min(semua_y) - 10, max(semua_y) + 10)

    # Note: Baris xlim/ylim di gambar asli hanya menggunakan titik awal,
    # yang mungkin tidak optimal jika hasil translasinya jauh.
    # Versi gambar asli:
    # plt.xlim(min(x1, x2, x3, x4) - 10, max(x1, x2, x3, x4) + 10)
    # plt.ylim(min(y1, y2, y3, y4) - 10, max(y1, y2, y3, y4) + 10)

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

```

```

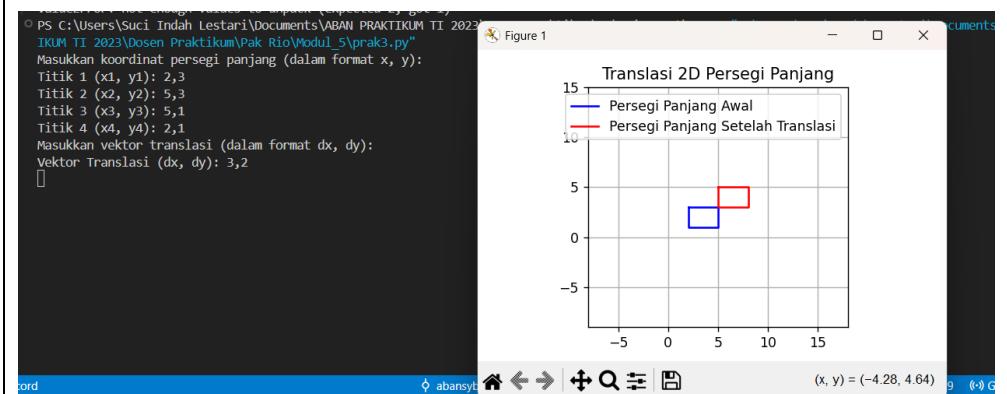
plt.title('Translasi 2D Persegi Panjang')

# Tampilkan plot
plt.show()

# Panggil fungsi utama
main()

```

### Hasil RUN



### Praktikum 4 : Translasi Objek Lingkaran

```

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

# Fungsi untuk menggambar lingkaran
def gambar_lingkaran(tengah, radius, warna='b', label=None):
    x, y = tengah
    theta = np.linspace(0, 2 * np.pi, 100)
    x_vals = x + radius * np.cos(theta)
    y_vals = y + radius * np.sin(theta)

    plt.plot(x_vals, y_vals, color=warna, label=label)

# Fungsi untuk melakukan translasi
def translasi(tengah, vektor):
    # Menambahkan nilai vektor translasi ke pusat lingkaran
    tengah_baru = (tengah[0] + vektor[0], tengah[1] + vektor[1])
    return tengah_baru

# Fungsi utama untuk input dinamis dan translasi
def main():
    # Input pusat lingkaran dan radius dari pengguna
    print("Masukkan pusat lingkaran (dalam format x,y):")
    x, y = map(float, input("Input Pusat Lingkaran (x, y): ").split(','))

    radius = float(input("Masukkan radius lingkaran:"))

    # Pusat lingkaran awal
    pusat_lingkaran_awal = (x, y)

    # Input vektor translasi dari pengguna
    print("\nMasukkan vektor translasi (dalam format dx,dy):")

```

```

dx, dy = map(float, input("Vektor Translasi (dx, dy): ").split(','))

# Translasi pusat lingkaran
pusat_lingkaran_tertranslasi = translasi(pusat_lingkaran_awal, (dx, dy))

# Plot lingkaran sebelum dan sesudah translasi
plt.figure()

# Gambar lingkaran awal
gambar_lingkaran(pusat_lingkaran_awal, radius, warna='b', label='Lingkaran Awal')

# Gambar lingkaran setelah translasi
gambar_lingkaran(pusat_lingkaran_tertranslasi, radius, warna='r',
label='Lingkaran Setelah Translasi')

# Menampilkan grid dan setting sumbu
plt.grid(True)

xmin = min(pusat_lingkaran_awal[0] - radius, pusat_lingkaran_tertranslasi[0] - radius)
xmax = max(pusat_lingkaran_awal[0] + radius, pusat_lingkaran_tertranslasi[0] + radius)
ymin = min(pusat_lingkaran_awal[1] - radius, pusat_lingkaran_tertranslasi[1] - radius)
ymax = max(pusat_lingkaran_awal[1] + radius, pusat_lingkaran_tertranslasi[1] + radius)

plt.xlim(xmin - 5, xmax + 5)
plt.ylim(ymin - 5, ymax + 5)

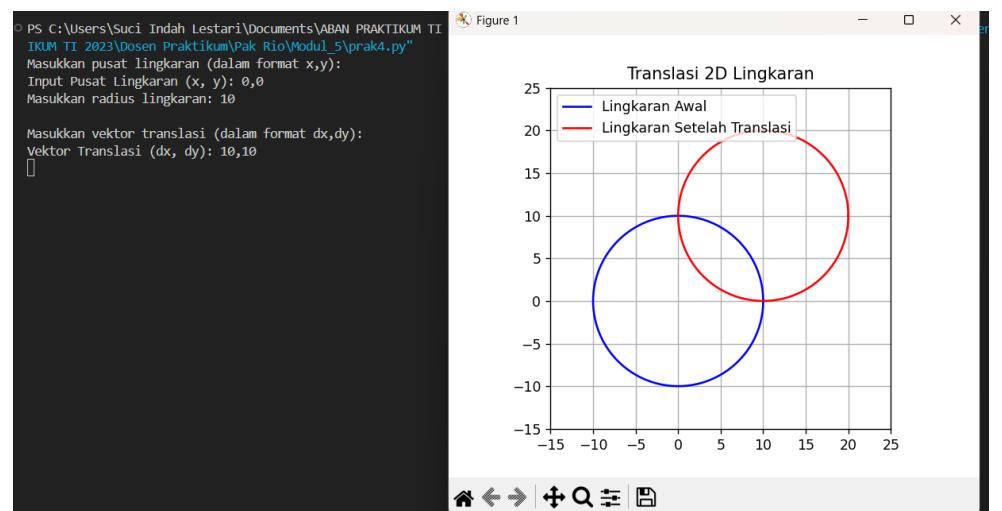
plt.gca().set_aspect('equal', adjustable='box')
plt.legend()
plt.title('Translasi 2D Lingkaran')

# Tampilkan plot
plt.show()

# Panggil fungsi utama
main()

```

### Hasil RUN



## TUGAS

1. Buatlah kode program sederhana untuk menggambar jajargenjang ?

### Code Program

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

def gambar_persegi_panjang(titik, warna='b', label=None):
    bentuk = np.array(titik)
    bentuk_tertutup = np.vstack([bentuk, bentuk[0]])
    plt.plot(bentuk_tertutup[:, 0], bentuk_tertutup[:, 1], color=warna,
             label=label, linewidth=2)

def translasi(titik, vektor):
    titik_baru = [[x + vektor[0], y + vektor[1]] for x, y in titik]
    return titik_baru

persegi_panjang_awal = [
    (10, 10),
    (40, 10),
    (50, 20),
    (20, 20)
]

vektor = (10, 10)
persegi_panjang_tertranslasi = translasi(persegi_panjang_awal, vektor)

plt.figure(figsize=(10, 5.5))
gambar_persegi_panjang(persegi_panjang_awal, warna='blue',
label='Persegi Panjang Awal')
gambar_persegi_panjang(persegi_panjang_tertranslasi, warna='red',
label='Persegi Panjang Setelah Translasi')
plt.xlim(0, 60)
plt.ylim(0, 30)
plt.xticks(np.arange(0, 61, 10))
plt.yticks(np.arange(0, 31, 5))
plt.grid(True)
plt.legend()
plt.title("Latihan Translasi Jajargenjang")
plt.gca().set_aspect('auto', adjustable='box')
plt.show()
```

### Hasil RUN

