

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

**Modul 8**



**Nama : Muhammad Rizal Nurfirdaus**

**NIM : 20230810088**

**Kelas : TINFC-2023-04**

**TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS KUNINGAN**

## Praktikum

### 1. Praktikum 1 : Pencerminkan Objek Segitiga

Source Code:

```
# Praktikum 1
import matplotlib.pyplot as plt
import numpy as np

# Fungsi untuk menggambar segitiga
def gambar_segitiga(titik):
    x = [titik[i][0] for i in range(len(titik))]
    y = [titik[i][1] for i in range(len(titik))]

    # Menutup segitiga (kembali ke titik awal)
    x.append(titik[0][0])
    y.append(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, titik):
    if sumbu == 'x':
        # Pencerminkan terhadap sumbu X
        return (titik[0], -titik[1])
    elif sumbu == 'y':
        # Pencerminkan terhadap sumbu Y
        return (-titik[0], titik[1])
    elif sumbu == 'origin':
        # Pencerminkan terhadap origin
        return (-titik[0], -titik[1])
    else:
        raise ValueError("Sumbu tidak valid. Pilih 'x', 'y', atau 'origin'.")

# --- Input Titik Segitiga ---
titik_segitiga = []
print("Input koordinat titik segitiga (format: x,y)")
for i in range(3):
    koordinat_input = input(f"Titik {i+1}: ")
    try:
        x, y = map(float, koordinat_input.split(','))
        titik_segitiga.append((x, y))
    except ValueError:
        print("Input tidak valid. Harap masukkan dua angka dipisahkan koma (misal: 1,2)")
```

```

        exit() # Keluar jika input salah

# Input jenis sumbu pencerminan
sumbu = input("Masukkan sumbu pencerminan ('x', 'y', atau 'origin'): ")
sumbu = sumbu.lower() # Pastikan huruf kecil untuk perbandingan

# --- Proses Pencerminan ---
titik_segitiga_pencerminan = []
try:
    for titik in titik_segitiga:
        titik_pencerminan = pencerminan_titik(sumbu, titik)
        titik_segitiga_pencerminan.append(titik_pencerminan)
except ValueError as e:
    print(f"Error: {e}")
    exit()

# --- Menggambar Segitiga Asli dan Hasil Pencerminan ---

# Membuat figure dan axes untuk plot
plt.figure(figsize=(8, 8))
plt.title(f"Pencerminan Segitiga Terhadap Sumbu {sumbu.upper()}")

# Garis bantu (grid)
plt.grid(color='gray', linestyle='--', linewidth=0.5)

# Gambar segitiga asli (warna biru)
gambar_segitiga(titik_segitiga)

# Tambahkan label untuk segitiga asli
plt.plot([], [], 'b-', label='Segitiga Asli') # Membuat entri dummy
untuk legend

# Gambar segitiga setelah pencerminan (warna oranye)
gambar_segitiga(titik_segitiga_pencerminan)

# Tambahkan label untuk segitiga hasil pencerminan
plt.plot([], [], 'C1-', label=f'Segitiga Pencerminan') # C1 adalah
warna default plot kedua

# Menambahkan legend dan mengatur tampilan plot
plt.legend(loc='best') # Menampilkan legend

# Mengatur batas sumbu x dan y agar terlihat seimbang
all_x = [p[0] for p in titik_segitiga + titik_segitiga_pencerminan]
all_y = [p[1] for p in titik_segitiga + titik_segitiga_pencerminan]

# Menghitung batas sumbu
x_min, x_max = min(all_x), max(all_x)

```

```

y_min, y_max = min(all_y), max(all_y)

# Menambahkan padding
padding = max(abs(x_max - x_min), abs(y_max - y_min)) * 0.15 + 1

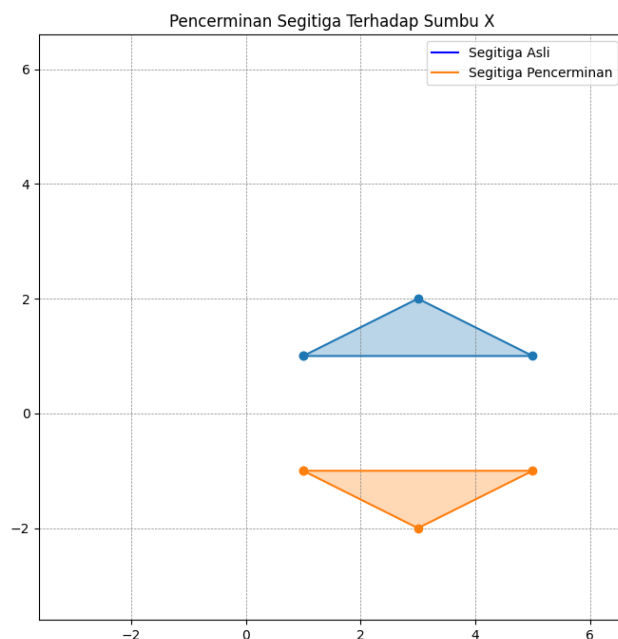
plt.xlim(min(x_min, y_min) - padding, max(x_max, y_max) + padding)
plt.ylim(min(x_min, y_min) - padding, max(x_max, y_max) + padding)

# Mengatur aspek rasio agar sumbu x dan y memiliki skala yang sama
(penting untuk geometri)
plt.gca().set_aspect('equal', adjustable='box')

# Menampilkan plot
plt.show()

```

Hasil Run:



Analisis: Program Praktikum 1 ini bertujuan untuk memvisualisasikan proses pencerminan segitiga pada bidang koordinat dua dimensi menggunakan bahasa Python dan pustaka matplotlib. Pengguna diminta memasukkan tiga titik koordinat yang membentuk sebuah segitiga, kemudian memilih jenis sumbu pencerminan, yaitu sumbu X, sumbu Y, atau titik origin. Pendekatan ini memungkinkan pengguna memahami konsep transformasi geometri secara interaktif.

Fungsi gambar\_segitiga digunakan untuk menggambar segitiga berdasarkan daftar titik yang diberikan. Fungsi ini bekerja dengan memisahkan koordinat x dan y dari setiap titik, lalu menutup segitiga dengan menghubungkan kembali titik terakhir ke titik awal. Segitiga digambar menggunakan garis dan diisi dengan warna transparan agar bentuknya mudah dikenali pada grafik.

Proses pencerminan dilakukan oleh fungsi pencerminan\_titik, yang menghitung koordinat baru setiap titik berdasarkan sumbu yang dipilih. Jika pencerminan dilakukan

terhadap sumbu X, nilai koordinat y dibalik tandanya. Untuk sumbu Y, koordinat x yang dibalik, sedangkan pencerminan terhadap origin membalik kedua koordinat. Validasi sumbu dilakukan untuk mencegah kesalahan input dari pengguna.

Setelah proses pencerminan selesai, program menampilkan segitiga asli dan segitiga hasil pencerminan pada satu grafik. Pengaturan grid, legenda, batas sumbu, dan aspek rasio yang sama digunakan agar perbandingan kedua segitiga terlihat jelas dan tidak mengalami distorsi. Secara keseluruhan, hasil visualisasi menunjukkan bahwa pencerminan tidak mengubah bentuk dan ukuran segitiga, melainkan hanya mengubah posisinya sesuai dengan sumbu pencerminan yang dipilih.

## 2. Praktikum 2 : Pencerminan Objek Persegi Panjang

Source Code:

```
# Praktikum 2
import matplotlib.pyplot as plt
import matplotlib.patches as patches

# Fungsi untuk menggambar persegi panjang
def gambar_persegi_panjang(ax, x, y, lebar, tinggi, label='Persegi
Panjang',
                           fill=True, edgecolor='blue', linewidth=2,
                           linestyle='-'):
    """
    Fungsi untuk menggambar persegi panjang pada grafik ax.
    x: koordinat x sudut kiri bawah
    y: koordinat y sudut kiri bawah
    lebar: lebar persegi panjang
    tinggi: tinggi persegi panjang
    """
    # Membuat objek persegi panjang
    persegi_panjang = patches.Rectangle((x, y), lebar, tinggi,
                                         edgecolor=edgecolor,
                                         facecolor='none',
                                         fill=fill, linewidth=linewidth,
                                         linestyle=linestyle)

    # Menambahkan persegi panjang ke axes
    ax.add_patch(persegi_panjang)

    # Menambahkan entri dummy untuk legend
    ax.plot([], [], color=edgecolor, linestyle=linestyle,
            linewidth=linewidth, label=label)

# Fungsi untuk melakukan pencerminan persegi panjang
def pencerminan_persegi_panjang(x, y, lebar, tinggi, sumbu):
    """
    Fungsi untuk melakukan pencerminan persegi panjang terhadap sumbu
```

```

yang ditentukan ('X', 'Y', atau 'Origin').
"""

# Pencerminkan terhadap sumbu X
if sumbu == 'X':
    # Koordinat baru: x tetap, y baru = y lama - tinggi. Tinggi
    tetap, Lebar tetap.
    # Catatan: Pencerminkan sumbu X membalik posisi y dan y+tinggi
    return x, -(y + tinggi), lebar, tinggi

# Pencerminkan terhadap sumbu Y
elif sumbu == 'Y':
    # Koordinat baru: x baru = x lama - lebar. Lebar tetap, y
    tetap, tinggi tetap.
    # Catatan: Pencerminkan sumbu Y membalik posisi x dan x+lebar
    return -(x + lebar), y, lebar, tinggi

# Pencerminkan terhadap origin
elif sumbu == 'ORIGIN':
    # Menggabungkan efek pencerminan X dan Y
    return -(x + lebar), -(y + tinggi), lebar, tinggi

# Input sumbu tidak valid
return -x, -y, lebar, tinggi # Kasus default atau error, mungkin
bukan yang dimaksud di kode asli

# --- Input dari pengguna ---
print("Masukkan detail untuk Persegi Panjang Asli:")
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: "))

# Input sumbu pencerminan
sumbu = input("Pilih sumbu pencerminan (X, Y, atau Origin):
").strip().upper()

# --- Membuat plot ---
fig, ax = plt.subplots(figsize=(8, 8))

# --- Menggambar persegi panjang asli ---
gambar_persegi_panjang(ax, x, y, lebar, tinggi, label='Persegi Panjang
Asli',
                        edgecolor='blue', linestyle='-')

# --- Menghitung dan Menggambar persegi panjang setelah 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=f'Persegi Panjang Setelah Pencermidan
({sumbu})',
                        edgecolor='red', linestyle='--')

# --- Menambahkan detail ke plot ---

# Mengatur batas sumbu X dan Y
ax.set_xlim(-10, 10)
ax.set_ylim(-10, 10)

# Mengatur aspek rasio agar skala sumbu X dan Y sama (penting untuk
geometri)
ax.set_aspect('equal', adjustable='box')

# Menambahkan garis bantu sumbu X dan Y yang lebih jelas
ax.axhline(0, color='black', linewidth=0.5, ls='--') # Sumbu X
ax.axvline(0, color='black', linewidth=0.5, ls='--') # Sumbu Y

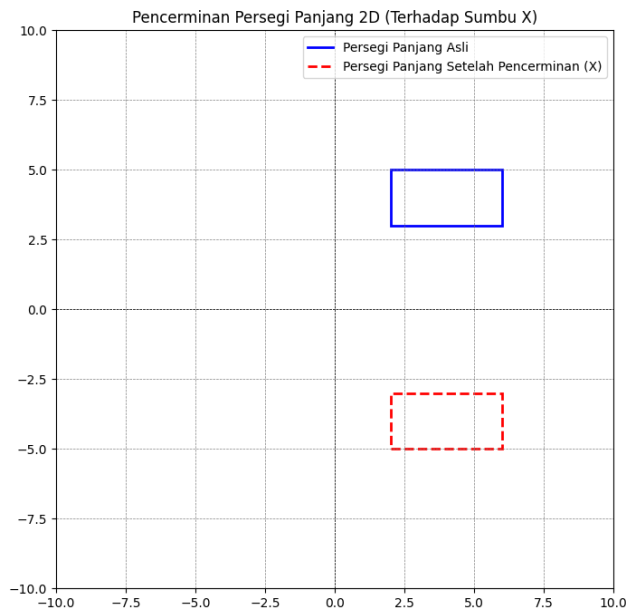
# Menambahkan grid
plt.grid(color='gray', linestyle='--', linewidth=0.5)

# Menampilkan legend
plt.legend(loc='best')

# Menambahkan judul plot
plt.title(f'Pencermidan Persegi Panjang 2D (Terhadap Sumbu {sumbu})')

# Menampilkan plot
plt.show()
```

### Hasil Run:



Analisis: Program ini bertujuan untuk memvisualisasikan pencerminan persegi panjang dalam bidang dua dimensi menggunakan pustaka matplotlib. Proses dimulai dengan mendefinisikan fungsi `gambar_persegi_panjang`, yang berfungsi menggambar persegi panjang berdasarkan koordinat sudut kiri bawah, lebar, dan tinggi. Fungsi ini juga menambahkan label ke legenda agar objek mudah dibedakan.

Selanjutnya, fungsi `pencerminan_persegi_panjang` digunakan untuk menghitung koordinat baru hasil pencerminan terhadap sumbu X, sumbu Y, atau titik origin. Pencerminan terhadap sumbu X dilakukan dengan membalik posisi vertikal persegi panjang, sedangkan pencerminan terhadap sumbu Y membalik posisi horizontal. Untuk pencerminan terhadap origin, kedua transformasi tersebut digabungkan.

Program kemudian menerima input dari pengguna berupa posisi dan ukuran persegi panjang serta pilihan sumbu pencerminan. Persegi panjang asli digambar terlebih dahulu, lalu hasil pencerminan digambar dengan warna dan garis yang berbeda. Pengaturan sumbu, grid, aspek rasio, dan legenda dilakukan agar visualisasi geometrinya akurat dan mudah dipahami.

### 3. Praktikum 3 : Pencerminan Objek Lingkaran

Source Code:

```
# Praktikum 3
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

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



```

radius: jari-jari lingkaran
"""
# Membuat objek lingkaran
lingkaran = patches.Circle((x, y), radius,
                             fill=fill, edgecolor=edgecolor,
                             linewidth=linewidth)

# Menambahkan lingkaran ke axes
ax.add_patch(lingkaran)

# Menambahkan entri dummy untuk legend
ax.plot([], [], color=edgecolor, linewidth=linewidth, label=label)

# Fungsi untuk mencerminkan lingkaran
def pencerminan_lingkaran(x, y, radius, sumbu):
    """
    Fungsi untuk melakukan pencerminan lingkaran.
    Pencerminan lingkaran hanya memengaruhi posisi pusat (x, y); jari-
    jari (radius) tetap.
    """

    # Pencerminan terhadap sumbu X
    if sumbu == 'X':
        # x tetap, y menjadi -y
        return x, -y, radius

    # Pencerminan terhadap sumbu Y
    elif sumbu == 'Y':
        # x menjadi -x, y tetap
        return -x, y, radius

    # Pencerminan terhadap Origin
    elif sumbu == 'ORIGIN':
        # x menjadi -x, y menjadi -y
        return -x, -y, radius

    # Sumbu tidak valid
    else:
        raise ValueError("Sumbu tidak valid. Pilih 'X', 'Y', atau
'Origin'.")

# --- Input dari pengguna ---
try:
    print("Masukkan detail untuk Lingkaran Asli:")
    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()

    if radius <= 0:
        raise ValueError("Radius harus lebih besar dari 0.")

except ValueError as e:
    print(f"Input tidak valid: {e}")
    exit()

# --- Membuat plot (Figure dan Axes) ---
fig, ax = plt.subplots(figsize=(8, 8))

# --- Menggambar lingkaran asli (warna biru) ---
gambar_lingkaran(ax, x, y, radius, label='Lingkaran Asli',
                  edgecolor='blue', linewidth=2)

# --- Menghitung dan Menggambar lingkaran setelah pencerminan (warna
merah) ---
try:
    x_baru, y_baru, radius_baru = pencerminan_lingkaran(x, y, radius,
sumbu)

    gambar_lingkaran(ax, x_baru, y_baru, radius_baru,
                    label=f'Lingkaran Setelah Pencerminan ({sumbu})',
                    edgecolor='red', linewidth=2)

except ValueError as e:
    print(f"Error dalam pencerminan: {e}")
    plt.close(fig) # Tutup figure yang sudah dibuat
    exit()

# --- Menambahkan detail ke plot ---

# Mengatur batas sumbu X dan Y (sesuai gambar: -10 hingga 10)
ax.set_xlim(-10, 10)
ax.set_ylim(-10, 10)

# Mengatur aspek rasio agar skala sumbu X dan Y sama (penting untuk
geometri)
ax.set_aspect('equal', adjustable='box')

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

# Menambahkan grid
plt.grid(color='gray', linestyle='--', linewidth=0.5)

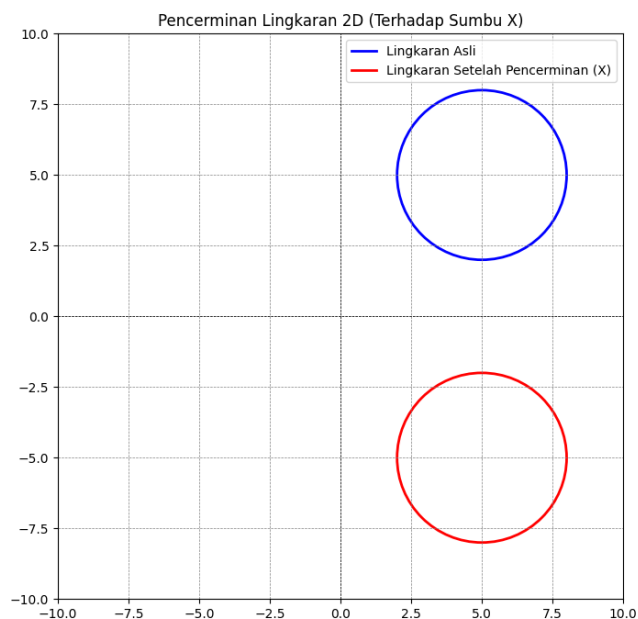
```

```
# Menampilkan legend
plt.legend(loc='best')

# Menambahkan judul plot
plt.title(f"Pencerminan Lingkaran 2D (Terhadap Sumbu {sumbu})")

# Menampilkan plot
plt.show()
```

Hasil Run :



Analisis: Program Praktikum 3 ini bertujuan untuk memvisualisasikan pencerminan lingkaran pada bidang koordinat dua dimensi menggunakan Python dan pustaka matplotlib. Pengguna diminta memasukkan koordinat pusat lingkaran, nilai jari-jari, serta memilih sumbu pencerminan, yaitu sumbu X, sumbu Y, atau titik origin. Dengan pendekatan ini, konsep pencerminan dapat diamati secara langsung melalui perubahan posisi objek pada grafik.

Fungsi gambar\_lingkaran digunakan untuk menggambar lingkaran berdasarkan koordinat pusat dan jari-jari yang diberikan. Lingkaran digambar sebagai objek Circle dari matplotlib.patches dan ditambahkan ke area grafik (axes). Untuk memudahkan identifikasi objek, program menambahkan entri legenda secara manual sehingga lingkaran asli dan hasil pencerminan dapat dibedakan dengan jelas.

Proses pencerminan dilakukan oleh fungsi pencerminan\_lingkaran, yang hanya mengubah posisi pusat lingkaran sesuai dengan sumbu pencerminan yang dipilih. Pada pencerminan terhadap sumbu X, koordinat y pusat dibalik tandanya, sedangkan pada sumbu Y koordinat x yang dibalik. Untuk pencerminan terhadap origin, kedua koordinat pusat dibalik. Jari-jari lingkaran tidak berubah karena pencerminan tidak memengaruhi ukuran objek.

Hasil visualisasi menunjukkan dua lingkaran dengan ukuran yang sama, tetapi berada pada posisi yang saling simetris terhadap sumbu yang dipilih. Lingkaran asli ditampilkan dengan warna biru, sedangkan lingkaran hasil pencerminan ditampilkan dengan warna merah. Pengaturan batas sumbu, grid, garis bantu sumbu, dan aspek rasio yang sama memastikan bahwa bentuk lingkaran tetap proporsional dan hubungan simetrinya mudah diamati. Secara keseluruhan, program ini berhasil menunjukkan bahwa pencerminan lingkaran hanya memengaruhi posisi pusat tanpa mengubah bentuk maupun ukuran lingkaran.

## LATIHAN/TUGAS

1. Buatlah kode program sederhana untuk menggambar objek belah ketupat yang dicerminkan terhadap sumbu X.

Source Code:

```
import matplotlib.pyplot as plt

# Titik-titik belah ketupat asli (posisi di atas sumbu X)
x_asli = [0, 2, 0, -2, 0]
y_asli = [3, 1, -1, 1, 3]

# Pencerminan terhadap sumbu X
x_cermin = x_asli
y_cermin = [-y for y in y_asli]

plt.figure(figsize=(6, 6))

# Gambar belah ketupat asli
plt.fill(x_asli, y_asli, color='gray', alpha=0.7, label='Belah Ketupat Asli')

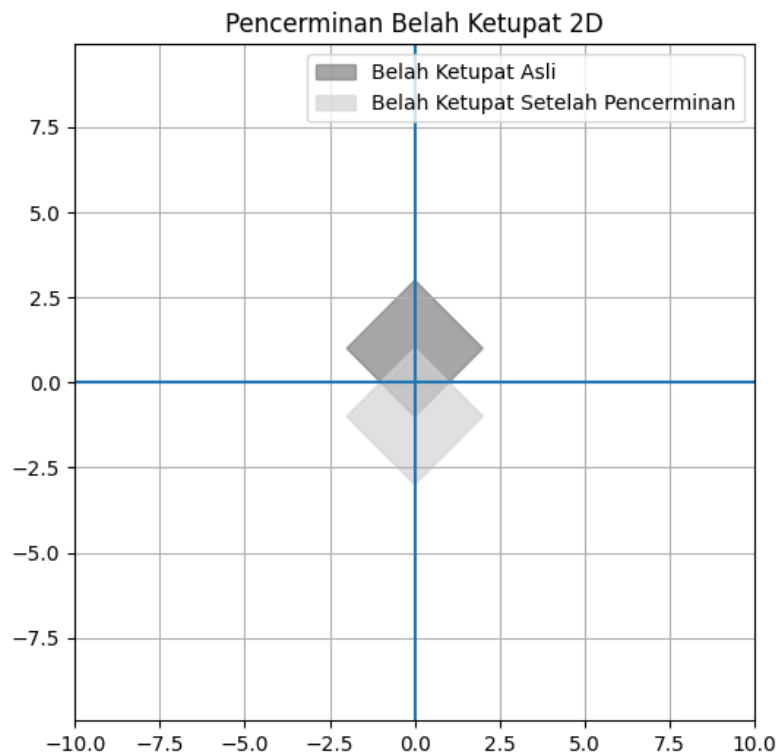
# Gambar hasil pencerminan
plt.fill(x_cermin, y_cermin, color='lightgray', alpha=0.7,
        label='Belah Ketupat Setelah Pencerminan')

# Sumbu koordinat
plt.axhline(0)
plt.axvline(0)

# Pengaturan tampilan
plt.grid(True)
plt.axis('equal')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.legend()
plt.title('Pencerminan Belah Ketupat 2D')
```

```
plt.show()
```

Hasil Run :



Analisis: Program Pencerminan belah ketupat terhadap sumbu X merupakan salah satu transformasi geometri dua dimensi yang memetakan setiap titik  $(x, y)$  menjadi  $(x, -y)$ . Transformasi ini tidak mengubah bentuk maupun ukuran belah ketupat, melainkan hanya mengubah posisinya secara vertikal terhadap sumbu X.

Pada hasil visualisasi, belah ketupat asli berada di atas sumbu X, sedangkan belah ketupat hasil pencerminan berada di bawah sumbu X dengan jarak yang sama. Kedua objek tampak simetris dan berpotongan tepat pada sumbu X, yang menunjukkan bahwa proses pencerminan telah dilakukan dengan benar.

Penggunaan sistem koordinat kartesius mempermudah analisis karena sumbu X dan sumbu Y berfungsi sebagai acuan utama dalam melihat perubahan posisi objek. Skala yang sama pada kedua sumbu menjaga proporsi belah ketupat sehingga tidak terjadi distorsi bentuk.

Secara keseluruhan, pencerminan terhadap sumbu X hanya memengaruhi tanda koordinat vertikal tanpa mengubah karakteristik geometris objek. Visualisasi ini membantu memahami konsep simetri dan transformasi geometri pada objek 2D secara jelas dan sistematis.

