

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

**Modul 10**



**Nama : Muhammad Rizal Nurfirdaus**

**NIM : 20230810088**

**Kelas : TINFC-2023-04**

**TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS KUNINGAN**

## Praktikum

### 1. Praktikum 1 : Translasi 3D Kubus

Source Code:

```
# Praktikum 1
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Fungsi untuk membuat objek kubus
def create_cube(length):
    # Titik-titik sudut dari kubus
    x = np.array([0, length, length, 0, 0, length, length, 0])
    y = np.array([0, 0, length, length, 0, 0, length, length])
    z = np.array([0, 0, 0, 0, length, length, length, length])
    return x, y, z

# Fungsi untuk menerapkan translasi pada objek
def translate_cube(x, y, z, tx, ty, tz):
    # Matriks translasi
    translation_matrix = np.array([tx, ty, tz])

    # Mengubah koordinat
    translated_x = x + translation_matrix[0]
    translated_y = y + translation_matrix[1]
    translated_z = z + translation_matrix[2]

    return translated_x, translated_y, translated_z

# Mengatur parameter objek
length = 3 # panjang sisi kubus

# Input translasi dari pengguna
tx = float(input("Masukkan nilai translasi pada sumbu X: "))
ty = float(input("Masukkan nilai translasi pada sumbu Y: "))
tz = float(input("Masukkan nilai translasi pada sumbu Z: "))

# Membuat objek kubus
x, y, z = create_cube(length)

# Menerapkan translasi
translated_x, translated_y, translated_z = translate_cube(x, y, z, tx,
ty, tz)

# Visualisasi objek
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
```

```

# Menggambar objek asli
ax.plot_trisurf(x, y, z, color='cyan', alpha=0.5, label='Asli')

# Menggambar objek yang sudah ditranslasi
ax.plot_trisurf(translated_x, translated_y, translated_z,
                 color='red', alpha=0.5, label='Translasi')

# Menambahkan label dan legenda
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Translasi 3D Objek Kubus dengan Input Dinamis')

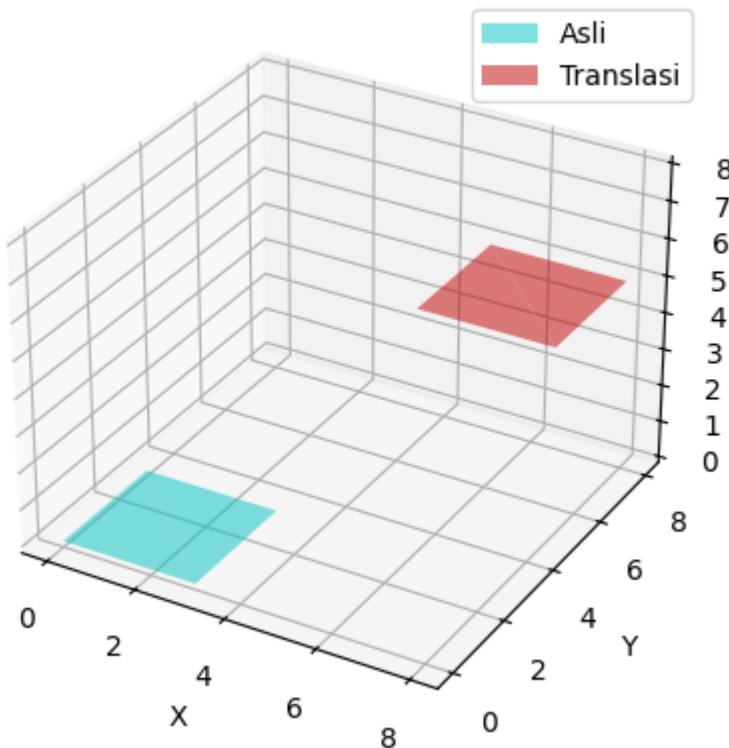
# Menghindari error legenda
handles, labels = ax.get_legend_handles_labels()
ax.legend(handles, labels)

# Menampilkan grafik
plt.show()

```

Hasil Run:

Translasi 3D Objek Kubus dengan Input Dinamis



Analisis: Pada Praktikum 1, tujuan utama adalah memvisualisasikan proses translasi pada objek kubus tiga dimensi dengan input dinamis pada sumbu X, Y, dan Z. Namun, berdasarkan hasil visualisasi yang diperoleh, objek yang ditampilkan tidak membentuk kubus secara utuh, melainkan hanya berupa satu bidang datar berbentuk persegi. Hal

ini terjadi karena metode visualisasi yang digunakan (`plot_trisurf`) hanya menggambar permukaan dari sekumpulan titik, bukan membangun seluruh sisi kubus yang seharusnya terdiri dari enam bidang. Meskipun secara perhitungan matematis translasi telah diterapkan dengan benar ditandai oleh perpindahan posisi objek asli ke posisi baru tanpa perubahan ukuran atau orientasi visualisasi yang dihasilkan tidak merepresentasikan bentuk kubus 3D secara lengkap. Dengan demikian, praktikum ini berhasil menunjukkan konsep translasi secara matematis, tetapi belum berhasil dalam menampilkan bentuk geometris kubus yang sesungguhnya.

## 2. Praktikum 2 : Penskalaan 3D Kubus

Source Code:

```
# Praktikum 2
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d.art3d import Poly3DCollection

# Fungsi untuk membuat objek limas
def create_pyramid(base_length, height):
    # Titik-titik sudut dari limas
    base_x = np.array([0, base_length, base_length, 0])
    base_y = np.array([0, 0, base_length, base_length])
    base_z = np.zeros(4) # z = 0 untuk dasar

    # Titik puncak limas
    apex = np.array([[base_length / 2, base_length / 2, height]])

    # Menggabungkan semua titik
    x = np.concatenate((base_x, apex[:, 0]))
    y = np.concatenate((base_y, apex[:, 1]))
    z = np.concatenate((base_z, apex[:, 2]))

    return x, y, z, apex

# Fungsi untuk menerapkan penskalaan pada objek
def scale_pyramid(x, y, z, scale_x, scale_y, scale_z):
    # Matriks penskalaan
    scale_matrix = np.array([scale_x, scale_y, scale_z])

    # Mengubah koordinat
    scaled_x = x * scale_matrix[0]
    scaled_y = y * scale_matrix[1]
    scaled_z = z * scale_matrix[2]

    return scaled_x, scaled_y, scaled_z

# Mengatur parameter objek
base_length = 4    # panjang sisi dasar limas
height = 5         # tinggi limas
```

```
# Input penskalaan dari pengguna
scale_x = float(input("Masukkan faktor penskalaan pada sumbu X: "))
scale_y = float(input("Masukkan faktor penskalaan pada sumbu Y: "))
scale_z = float(input("Masukkan faktor penskalaan pada sumbu Z: "))

# Membuat objek limas
x, y, z, apex = create_pyramid(base_length, height)

# Menerapkan penskalaan
scaled_x, scaled_y, scaled_z = scale_pyramid(x, y, z, scale_x, scale_y,
scale_z)

# Visualisasi objek
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Menggambar objek asli
ax.add_collection3d(
    Poly3DCollection([list(zip(x, y, z))], color='cyan', alpha=0.5,
label='Asli')
)

# Menggambar objek yang sudah diskala
ax.add_collection3d(
    Poly3DCollection([list(zip(scaled_x, scaled_y, scaled_z))],
color='red', alpha=0.5, label='Skala')
)

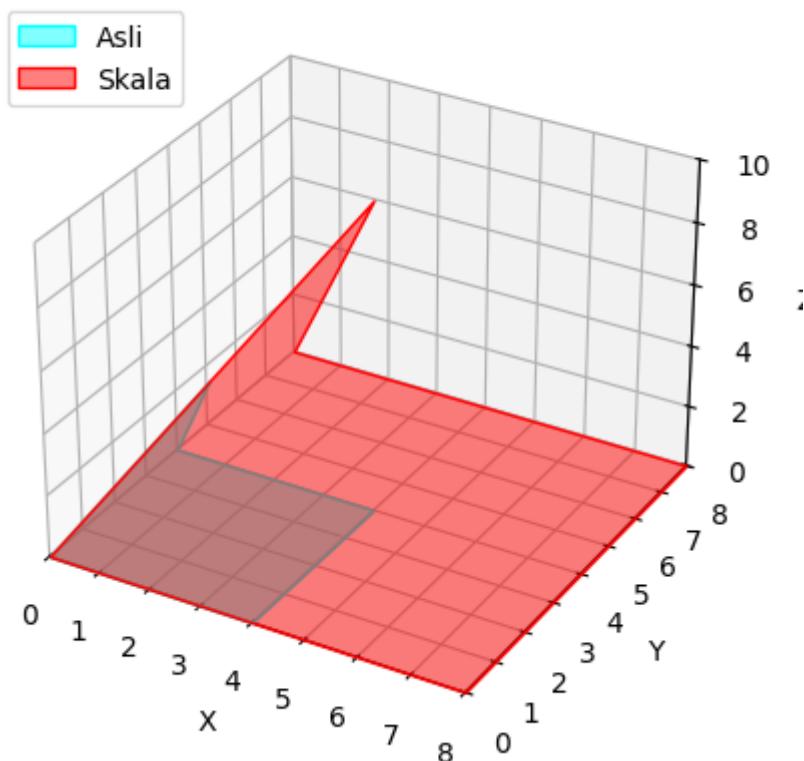
# Menambahkan label dan legenda
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Penskalaan 3D Objek Limas dengan Input Dinamis')
ax.legend(loc='upper left')

# Menentukan batas sumbu
ax.set_xlim([0, base_length * scale_x])
ax.set_ylim([0, base_length * scale_y])
ax.set_zlim([0, height * scale_z])

# Menampilkan grafik
plt.show()
```

Hasil Run:

### Penskalaan 3D Objek Limas dengan Input Dinamis



Analisis: Pada Praktikum 2, praktikum bertujuan untuk menerapkan transformasi penskalaan pada objek limas tiga dimensi dengan faktor skala berbeda pada sumbu X, Y, dan Z. Akan tetapi, hasil visualisasi menunjukkan bahwa objek yang ditampilkan tidak menyerupai limas, melainkan hanya berupa satu bidang poligon yang tidak membentuk bangun ruang secara utuh. Hal ini disebabkan oleh cara penggambaran objek yang menggabungkan seluruh titik koordinat menjadi satu poligon menggunakan Poly3DCollection, tanpa memisahkan alas dan sisi-sisi segitiga limas. Meskipun proses penskalaan telah diterapkan dengan benar secara matematis terlihat dari perubahan ukuran objek sesuai faktor skala yang diberikan visualisasi tidak mampu merepresentasikan struktur limas tiga dimensi yang seharusnya memiliki satu alas dan empat sisi tegak. Oleh karena itu, praktikum ini berhasil menunjukkan konsep penskalaan koordinat, namun belum berhasil menampilkan bentuk limas 3D yang sesuai secara visual.

## LATIHAN/TUGAS

1. Buatlah kode program sederhana untuk menampilkan *shear* untuk objek persegi panjang berikut:

Jawab :

Modul Translasi dan Penskalaan Objek 3D

Pada praktikum modul transformasi objek 3D ini, penulis mempelajari dua jenis transformasi geometri, yaitu translasi dan penskalaan. Kedua praktikum tersebut memberikan pemahaman yang lebih jelas mengenai bagaimana suatu objek tiga dimensi dapat dimanipulasi secara matematis dan divisualisasikan menggunakan koordinat dalam ruang 3D.

### Praktikum 1: Translasi 3D Kubus

Pada praktikum pertama, dilakukan translasi terhadap sebuah objek kubus dalam ruang tiga dimensi. Translasi dilakukan dengan menambahkan vektor translasi pada setiap titik sudut kubus. Dari praktikum ini dapat dipahami bahwa translasi tidak mengubah bentuk, ukuran, maupun orientasi kubus, melainkan hanya memindahkan posisi kubus ke lokasi yang baru sesuai dengan vektor translasi yang diberikan.

Melalui visualisasi hasil translasi, terlihat jelas bahwa jarak antar titik sudut kubus tetap sama sebelum dan sesudah transformasi. Hal ini membuktikan bahwa translasi merupakan transformasi yang bersifat isometrik. Praktikum ini membantu penulis memahami konsep pergeseran objek 3D secara lebih konkret, terutama dalam kaitannya dengan penerapan koordinat dan sumbu X, Y, dan Z.

### Praktikum 2: Penskalaan 3D Kubus

Pada praktikum kedua, dilakukan penskalaan terhadap objek 3D berupa limas atau kubus. Penskalaan dilakukan dengan mengalikan setiap koordinat titik objek dengan faktor skala tertentu. Dari hasil praktikum ini dapat diamati bahwa ukuran objek berubah sesuai dengan nilai faktor skala yang digunakan.

Jika faktor skala lebih besar dari satu, objek menjadi lebih besar, sedangkan jika faktor skala kurang dari satu, objek menjadi lebih kecil. Selain itu, apabila faktor skala pada masing-masing sumbu berbeda, maka bentuk objek dapat mengalami perubahan proporsi. Praktikum ini memberikan pemahaman bahwa penskalaan tidak hanya memengaruhi ukuran objek, tetapi juga dapat memengaruhi proporsi dan tampilan visual objek dalam ruang 3D.

### Kesimpulan

Berdasarkan kedua praktikum tersebut, dapat disimpulkan bahwa translasi dan penskalaan merupakan transformasi dasar yang sangat penting dalam pemodelan objek 3D. Translasi berfungsi untuk mengatur posisi objek, sedangkan penskalaan digunakan untuk mengatur ukuran objek. Melalui praktikum ini, penulis memperoleh pemahaman yang lebih baik mengenai konsep transformasi 3D serta penerapannya dalam grafika komputer dan pemodelan objek tiga dimensi.