

LAPORAN GRAFIKA KOMPUTER

PEMBUATAN OBJEK PLANET 3D



UNESA
Universitas Negeri Surabaya

Oleh:

Muhammad Didik Wahyudi 20051397040

DIPLOMA IV MANAJAMEN INFORMATIKA

FAKULTAS VOKASI

UNIVERSITAS NEGERI SURABAYA

2022

PEMBUATAN OBJEK PLANET 3D

Pada OpenGL untuk membuat objek 3D yang saya buat, memerlukan fungsi yang telah disediakan oleh OpenGL, fungsi-fungsi tersebut adalah:

```
# warna dan variabel lain untuk memudahkan modifikasi
planetColor = (0.0, 0.6, 0.7)
partialDiskColor=(0.7, 0.4, 0.0)

# memulai solid Drawing
def draw(slices, stacks):

    t = time.time() % 1000
    angle = t*90
    quad = gluNewQuadric()

    glColor3fv(planetColor)
```

- **planetColor** digunakan untuk warna planet yang akan di buat.
- **partialDiskColor** partial disk menunjukkan bahwa objek dapat menghasilkan suatu hal yang sama atau ganda (overlapping).
- **Slices** fungsi slice mengembalikan objek slice yang dapat digunakan untuk memotong objek sequence
- **Stacks** atau tumpukan adalah sebuah kumpulan data yang diletakkan di atas data yang lain.
- **t** untuk menampilkan fungsi time dengan cara import time. Fungsinya untuk mendapatkan waktu saat ini dalam milidetik
- **angle** adalah derajat putaran, dikalikan 90 pada fungsi time.
- **quad** fungsi yang digunakan untuk menyelesaikan integral atau quadrature dengan menggunakan metode simpson dalam analisis numerik.
- **glColor3fv** fungsi untuk mengatur warna saat ini dari array nilai warna yang sudah ada.

```

# planet
glPushMatrix()
glTranslatef(1.8, 1.5, -6)
glRotatef(45, 1, 0, 0)
glRotatef(angle, 0, 0, 1)
gluSphere(quad, 2, slices, stacks) # quads, radius, slices, stacks
glPopMatrix()

# space station
glColor3fv(partialDiskColor)
glPushMatrix()
glTranslatef(4.4, 1.5, -6)
# glRotatef(-60, 1, 0.2, 0)
glRotatef(-angle, 0, 0, 1)
gluPartialDisk(quad, 0.5, 1, slices, stacks, 0, 270) # quad, inner, outer, slices, loops, start angle, sweep angle
glPopMatrix()

# Cincin
glColor3f(1.0, 0.8, 0.2) # mengontrol warna cincin
glPushMatrix()
glTranslatef(1.8, 1.5, -6)
# glRotatef(-60, 1, 0.2, 0)
glRotatef(-angle, 0, 1, 1)
gluDisk(quad, 2.2, 2.7, slices, stacks) # quad, inner, outer, slices, loops
glPopMatrix()

# glColor3f(0.7, 0.5, 0.0) # Jika ingin mengubah warna cincin lainnya.
glPushMatrix()
glTranslatef(1.8, 1.5, -6)
glRotatef(-60, 1, 0.2, 0) # memutar dengan angle
glRotatef(-angle, 1, 0, 1) # putaran cincin terus menerus.
gluDisk(quad, 2.2, 2.7, slices, stacks) # quad, inner, outer, slices, loops
glPopMatrix()

```

- **glPushMatrix**

Fungsi glPushMatrix mendorong tumpukan matriks saat ini satu per satu, menduplikasi matriks saat ini. Artinya, setelah panggilan glPushMatrix , matriks di bagian atas tumpukan identik dengan yang di bawahnya.

- **glTranslatef**

Jika mode matriks GL_MODELVIEW atau GL_PROJECTION, semua objek yang digambar setelah glTranslatef dipanggil diterjemahkan. Menggunakan glPushMatrix dan glPopMatrix untuk menyimpan dan memulihkan sistem koordinat yang tidak diterjemahkan.

- **glPopMatrix**

Fungsi glPopMatrix memunculkan tumpukan matriks saat ini, mengganti matriks saat ini dengan yang di bawahnya pada tumpukan. Awalnya, masing-masing tumpukan berisi satu matriks, matriks identitas.

- **glRotatef**

Fungsi glRotatef menghitung matriks yang melakukan rotasi berlawanan arah jarum jam derajat sudut tentang vektor dari asal melalui titik (x, y, z). semua objek yang digambar setelah glRotatef dipanggil akan diputar.

- **gluSphere (untuk membuat planet dan ufo)**

Fungsi gluSphere menggambar bola radius yang diberikan berpusat di sekitar asal. Bola dibagi di sekitar sumbu z menjadi irisan dan sepanjang sumbu z menjadi tumpukan (mirip dengan garis bujur dan lintang).

- **gluPartialDisk**

Fungsi gluPartialDisk merender disk parsial pada bidang $z = 0$. Disk parsial mirip dengan disk penuh, kecuali bahwa hanya subset disk dari startAngle melalui startAngle + sweepAngle + disertakan (di mana 0 derajat berada di sepanjang sumbu y positif, 90 derajat berada di sepanjang sumbu x positif, 180 derajat berada di sepanjang sumbu y negatif, dan 270 derajat berada di sepanjang sumbu x negatif).

- **gluDisk (untuk membuat cincin)**

Fungsi gluDisk merender disk pada bidang $z = 0$. Disk memiliki radius outerRadius, dan berisi lubang melingkar konsentris dengan radius innerRadius. Jika innerRadius adalah 0, maka tidak ada lubang yang dihasilkan.

- **gluCylinder (untuk membuat ufo)**

Fungsi gluCylinder menggambar silinder yang berorientasi di sepanjang sumbu z. Dasar silinder ditempatkan pada $z = 0$, dan bagian atas pada zheight = .

- **gluNewQuadric (untuk membuat transformasi bentuk planet)**

Fungsi gluNewQuadric membuat dan mengembalikan pointer ke objek quadric baru. Lihat objek ini saat memanggil fungsi penyajian dan kontrol quadric. Nilai pengembalian nol berarti tidak ada cukup memori untuk dialokasikan ke objek.

- **gluQuadricDrawStyle (untuk membuat mode x-ray)**

Fungsi gluQuadricDrawStyle menentukan gaya gambar untuk quadric yang dirender dengan quadObject.

Sebagai tambahan, OpenGL mendukung lighting untuk mengaktifkan dan menonaktifkan perhitungan pencahayaan, panggil glEnable dan glDisable dengan argumen GL_LIGHTING. Pencahayaan awalnya dinonaktifkan. Saat diaktifkan, sumber cahaya yang diaktifkan berkontribusi pada perhitungan pencahayaan. Saya menggunakan light_ambient, diffuse, specular dan position untuk mengatur pencahayaan.

```
# mengatur light
light_ambient = [0.0, 0.0, 0.0, 1.0]
light_diffuse = [1.0, 1.0, 1.0, 1.0]
light_specular = [1.0, 1.0, 1.0, 1.0]
light_position = [2.0, 5.0, 5.0, 0.0]

mat_ambient = [0.7, 0.7, 0.7, 1.0]
mat_diffuse = [0.8, 0.8, 0.8, 1.0]
mat_specular = [1.0, 1.0, 1.0, 1.0]
high shininess = [100.0]
```

PROSES TRANSFORMASI PLANET 3D

Aplikasi pembuatan berupa transformasi planet 3D. Dengan melakukan beberapa pembentukkan objek-objek seperti planet, cincin, dan ufo. kemudian di gabung menjadi satu bagian . Adapun proses seperti transformasi bentuk, pergerakan ufo dan Penempatan arah kamera dari objek. Untuk mengontrol objek transformasi planet ada tombol seperti :

- **Tekan PANAHA ATAS atau PANAHA BAWAH untuk transformasi bentuk objek .**
- **A,W,S,D untuk bergerak di sekitar area objek.**
- **Scroll Mouse untuk zoom in dan out.**
- **X untuk masuk ke mode X-ray [wireframe mode].**

Ada juga beberapa proses manipulasi yang dilakukan pada objek transformasi planet 3D yaitu :

- **Initialize Game**

Pada tahapan ini kita melakukan inisialisasi atau deklarasi variabel dan objek yang dibutuhkan.

```

slices = 7
stacks = 7

xrayButton = 1
xyz_pos = 0.1
movingDown = False
z_pos = -8

pygame.init()
display = (800, 600)
pygame.display.set_caption("OpenGL --> Transformasi Planet 3D")
pygame.display.set_mode(display, DOUBLEBUF | OPENGL)

glMatrixMode(GL_PROJECTION)
gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)

glMatrixMode(GL_MODELVIEW)
glTranslatef(-2.0, -1.0, -5)

```

Fungsi slices dan stack digunakan untuk transformasi dari objek. Lalu ada xrayButton digunakan untuk memasuki mode xray dan selanjutnya ada display untuk mengatur resolusi display layar.

Fungsi glMatrixMode menentukan matriks mana yang merupakan matriks saat ini.

Tumpukan matriks yang merupakan target untuk operasi matriks berikutnya. Parameter mode GL_PROJECTION menerapkan operasi matriks berikutnya ke tumpukan matriks proyeksi.

Parameter mode GL_MODELVIEW menerapkan operasi matriks berikutnya ke tumpukan matriks modelview.

- **Event Loop**

Event loop berfungsi untuk mengecek event apa saja yang terjadi di dalam game. Misal saat mouse discroll dan saat salah satu keyboard ditekan

.

```

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        pygame.quit()
        quit()
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_UP:
            slices += 1
            stacks += 1
        if event.key == pygame.K_DOWN and (slices > 3 and stacks > 3):
            slices -= 1
            stacks -= 1
        if event.key == pygame.K_x:
            xrayButton = xrayButton+1

        # mengontrol view
        if event.key == pygame.K_a:
            glTranslatef(0.5, 0, 0)
        if event.key == pygame.K_d:
            glTranslatef(-0.5, 0, 0)

        if event.key == pygame.K_w:
            glTranslatef(0, -0.5, 0)
        if event.key == pygame.K_s:
            glTranslatef(0, 0.5, 0)
    if event.type == pygame.MOUSEBUTTONDOWN:
        if event.button == 4:
            glTranslatef(0, 0, 0.5)
        if event.button == 5:
            glTranslatef(0, 0, -0.5)

```

Fungsi dari pygame.QUIT apabila ditekan tombol x, maka program akan berhenti. apabila ada tombol keyboard yang ditekan, maka akan dilakukan fungsi pygame.KEYDOWN dan apabila jika ditekan tombol panah atas dan bawah maka akan melakukan transformasi bentuk dari objek.

Dan selanjutnya untuk masuk ke mode xray dengan cara menekan tombol X pada keyboard. Untuk mengatur objek bergerak di sekitar area dengan menekan tombol W maka objek akan mengarah kebawah, tombol S mengarah keatas, tombol D menyamping ke kiri dan tombol A menyamping ke kanan. Mengatur view dengan cara scroll mouse keatas maka objek akan zoom in dan scroll mouse kebawah objek akan zoom out.

- **Game Loop**

Game loop akan mengulang terus menerus selama game-nya berjalan.

```
# Pergerakan UFO
if(movingDown):
    if(xyz_pos > -4):
        xyz_pos = xyz_pos - 0.04
    else:
        movingDown = False
else:
    if(xyz_pos < 8):
        xyz_pos = xyz_pos + 0.04
    else:
        movingDown = True
```

Objek ufo bergerak otomatis secara vertikal keatas dan kebawah.

- Clear Screen dan Draw the game object

```
# glRotatef(1, 3, 1, 1)
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
glColor3f(1.0, 1.0, 3.0)

# memulai menggambar stasiun luar angkasa
if(xrayButton % 2):
    draw(slices, stacks)
else:
    xrayDraw(slices, stacks)
UFO(xyz_pos)

# background color (warna angkasa)
glClearColor(0.0, 0.0, 0.12, 1)
glEnable(GL_CULL_FACE)
glCullFace(GL_BACK)

glEnable(GL_DEPTH_TEST)
glDepthFunc(GL_LEQUAL)

glEnable(GL_LIGHT0)
glEnable(GL_NORMALIZE)
glEnable(GL_COLOR_MATERIAL)
glEnable(GL_LIGHTING)

glLightfv(GL_LIGHT0, GL_AMBIENT, light_ambient)
glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse)
glLightfv(GL_LIGHT0, GL_SPECULAR, light_specular)
glLightfv(GL_LIGHT0, GL_POSITION, light_position)

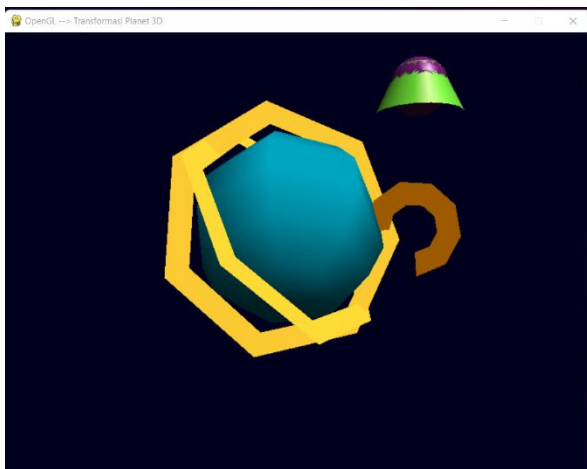
glMaterialfv(GL_FRONT, GL_AMBIENT, mat_ambient)
glMaterialfv(GL_FRONT, GL_DIFFUSE, mat_diffuse)
glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular)
glMaterialfv(GL_FRONT, GL_SHININESS, high_shininess)

pygame.display.flip()
pygame.time.wait(10)
```

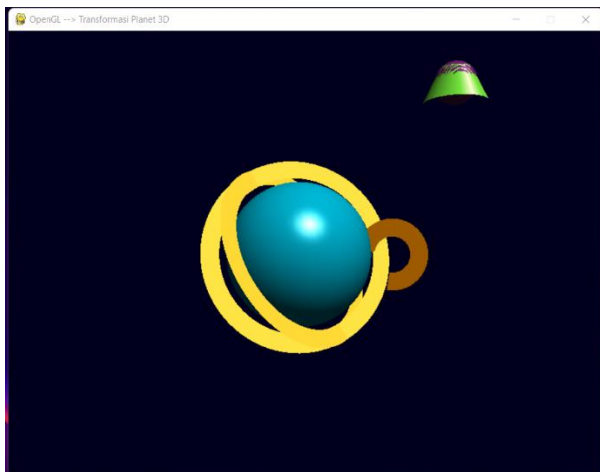
fungsi `glClear` untuk menghapus semua display dan fungsi `pygame.time.wait` menunggu 10ms sebelum looping lagi. Pada tahapan draw the game object menggambar mode xray pada objek.

SCREENSHOT HASIL OUTPUT OBJEK PLANET 3D

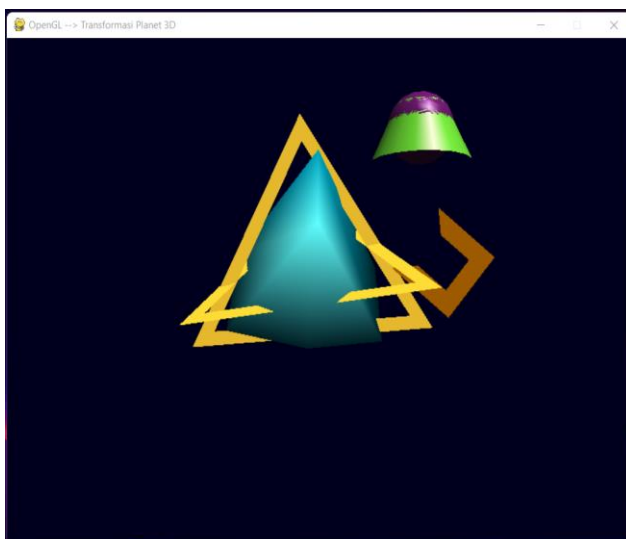
1. Awal start



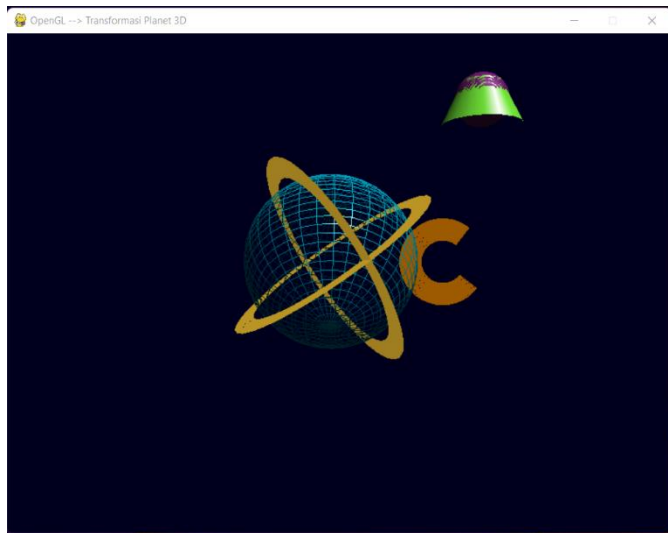
2. Jika ditekan tombol panah atas secara terus menerus



3. Jika ditekan tombol panah bawah



4. Masuk ke mode xray dengan menekan tombol X pada keyboard



5. Mengatur posisi object dengan tombol W,A,S,D dan scroll mouse untuk zoom in dan zoom out

