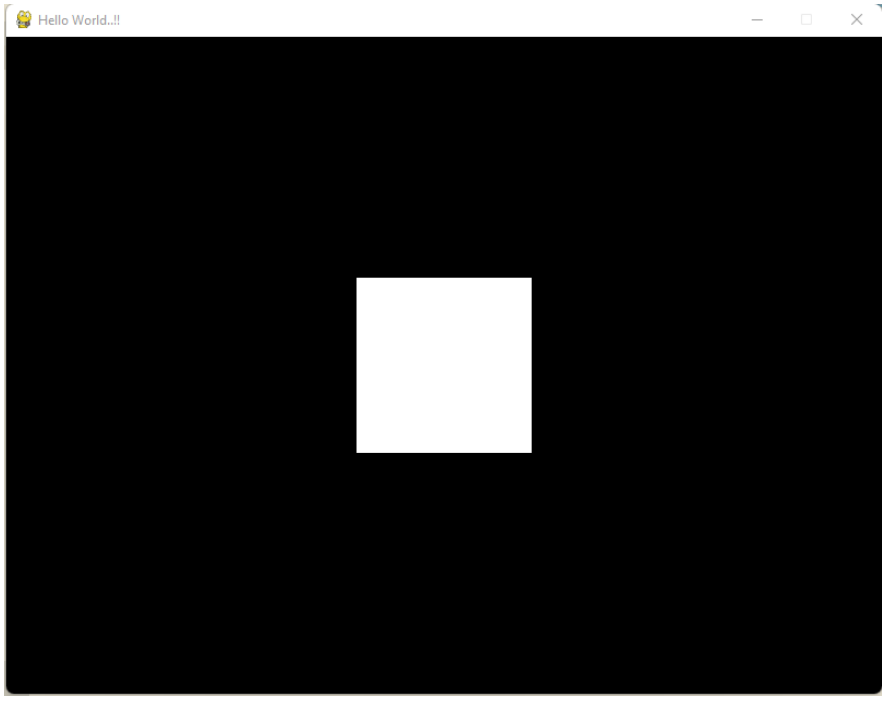
MODUL 1

Latihan1_1. GLUT “Hello World” & Gambar Segiempat

Source code:

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Cube():
    glBegin(GL_QUADS)
    glVertex3f(-0.5, -0.5, 0.5)
    glVertex3f(0.5, -0.5, 0.5)
    glVertex3f(0.5, 0.5, 0.5)
    glVertex3f(-0.5, 0.5, 0.5)
    glEnd()

def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!!')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Cube()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```

Output program:

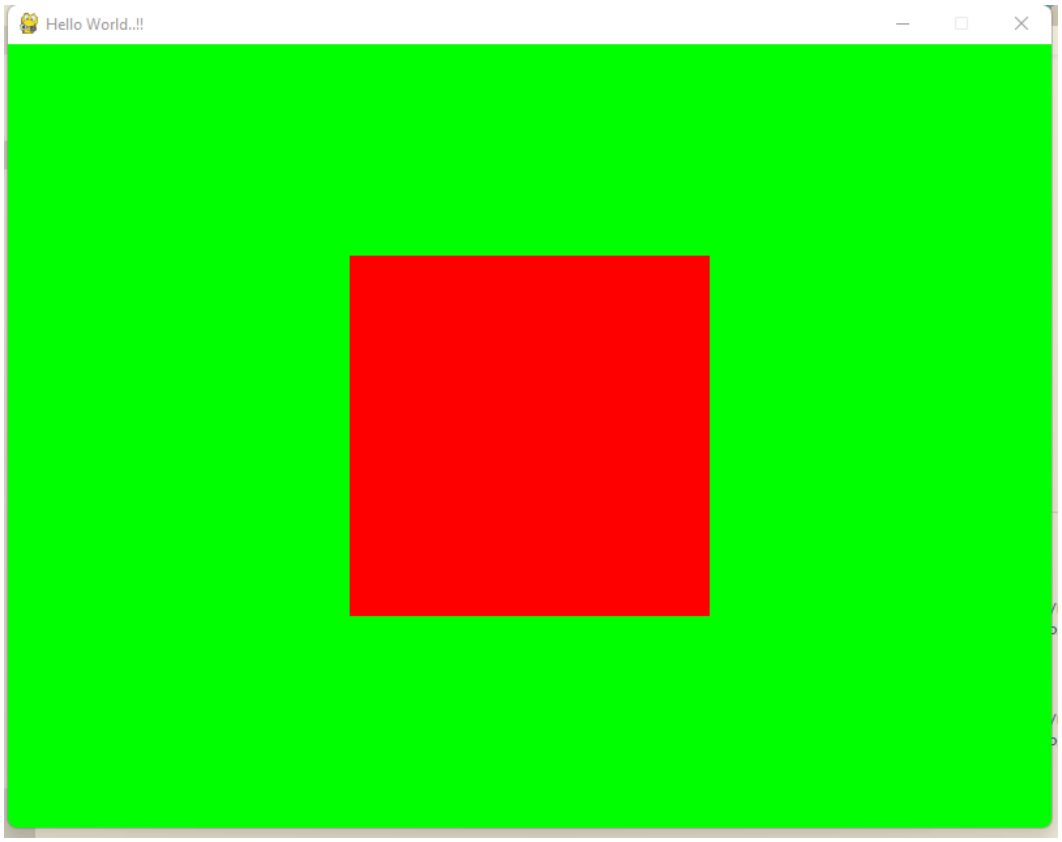


Tugas:

1. Ganti warna background dengan mengganti nilai glClearColor:
   Hijau → glClearColor (0.0, 1.0, 0.0, 1.0)
2. Ganti warna object (segi empat) dengan mengganti nilai dalam glColor3f
   Merah → glColor3f (1.0, 0.0, 0.0)

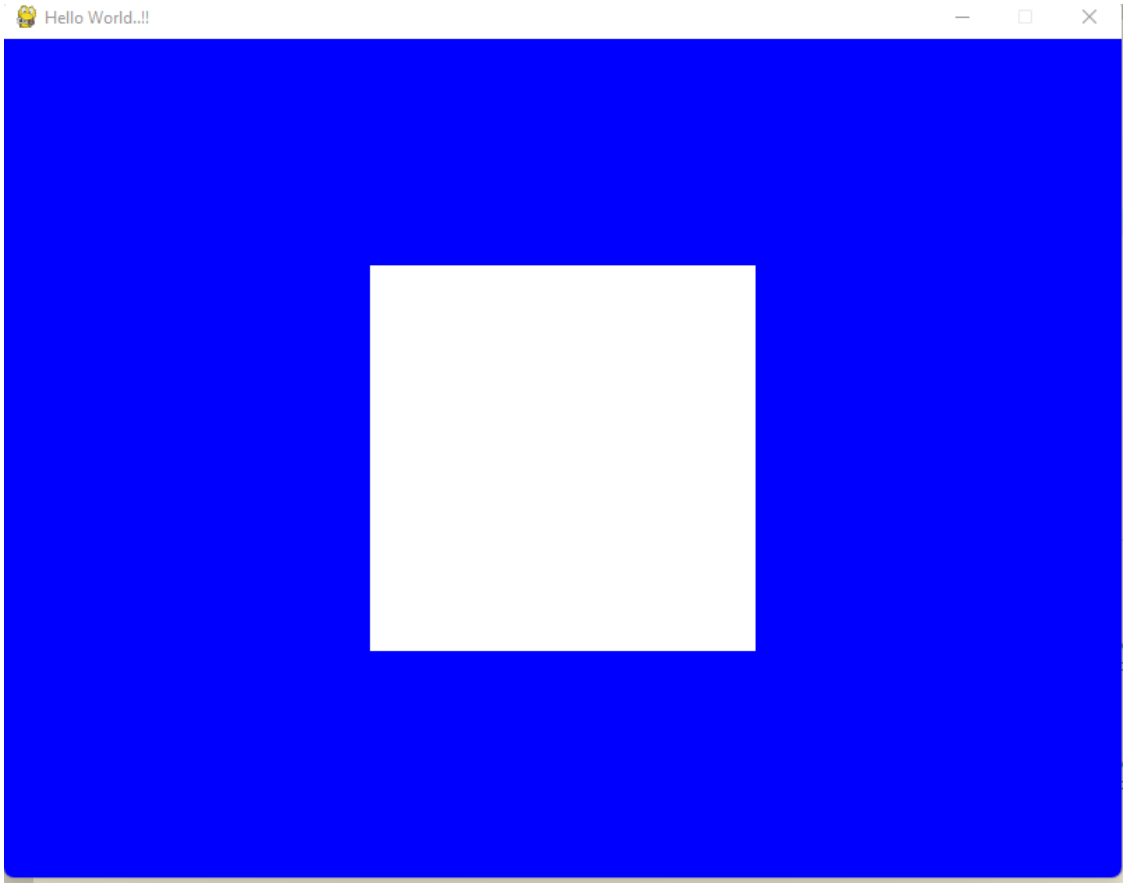
```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Cube():
    glBegin(GL_QUADS)
    glVertex3f(-0.8, -0.8, 0.8)
    glVertex3f(0.8, -0.8, 0.8)
    glVertex3f(0.8, 0.8, 0.8)
    glVertex3f(-0.8, 0.8, 0.8)
    glEnd()

def init():
    glClearColor (0.0, 1.0, 0.0, 1.0);   #background colour hijau
    glColor3f(1.0, 0.0, 0.0);            #object colour red
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();                    #load identity
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection
```

```
def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!!')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Cube()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```



Ganti warna background dengan mengganti nilai glClearColor:
Biru → glClearColor (0.0, 0.0, 1.0, 1.0)
Ganti warna object (segi empat) dengan mengganti nilai dalam glColor3f
Putih → glColor3f (1.0, 1.0, 1.0)

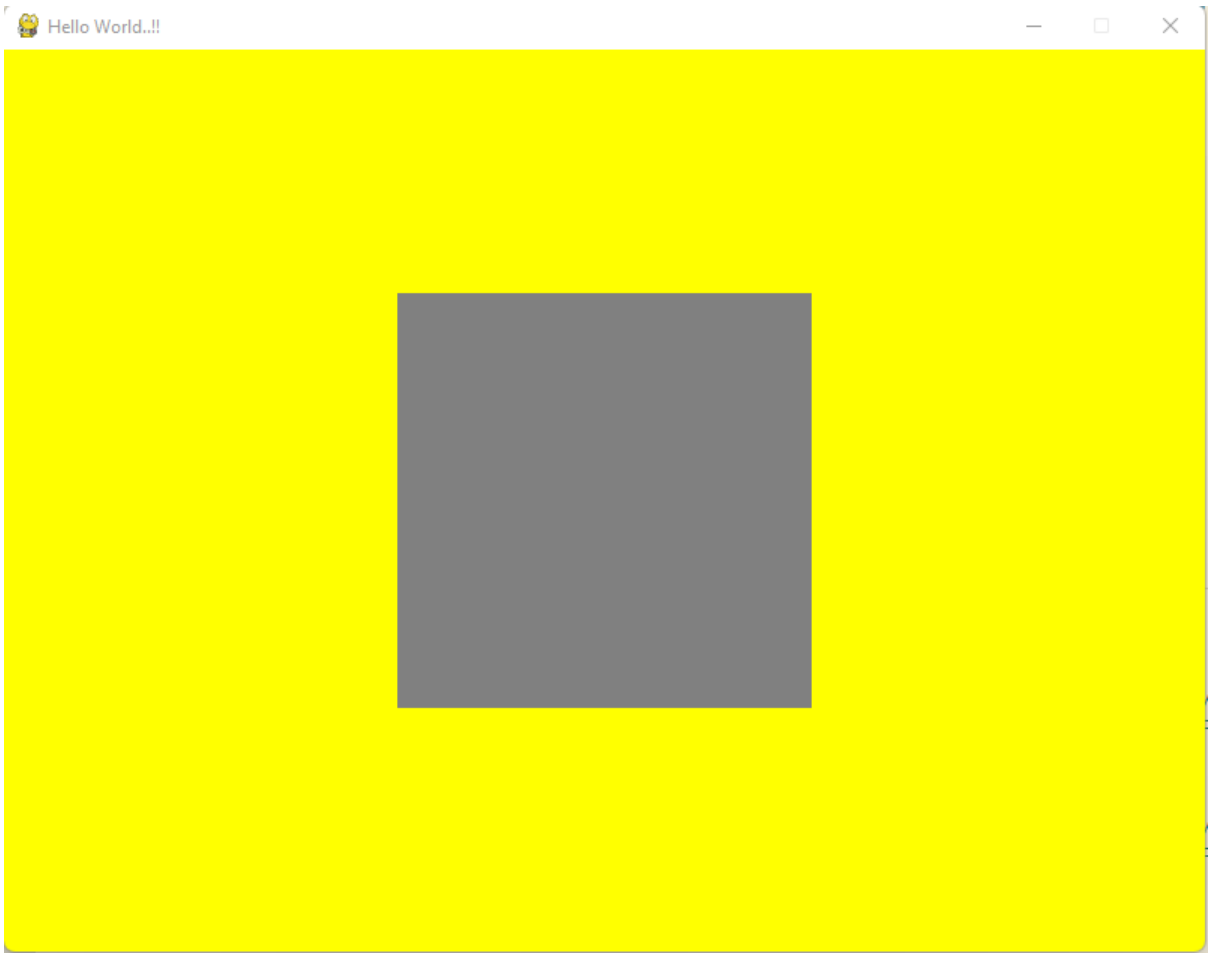
```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Cube():
    glBegin(GL_QUADS)
    glVertex3f(-0.8, -0.8, 0.8)
    glVertex3f(0.8, -0.8, 0.8)
    glVertex3f(0.8, 0.8, 0.8)
    glVertex3f(-0.8, 0.8, 0.8)
    glEnd()

def init():
    glClearColor(0.0, 0.0, 1.0, 1.0); #Set background color to blue
    glColor3f(1.0, 1.0, 1.0);          #set object color to white
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();                    #load identity
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection


def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!!')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Cube()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```



Ganti warna background dengan mengganti nilai glClearColor:
Kuning → glClearColor (1.0, 1.0, 0.0, 1.0)
Ganti warna object (segi empat) dengan mengganti nilai dalam glColor3f
Abu-Abu → glColor3f (0.5, 0.5, 0.5)

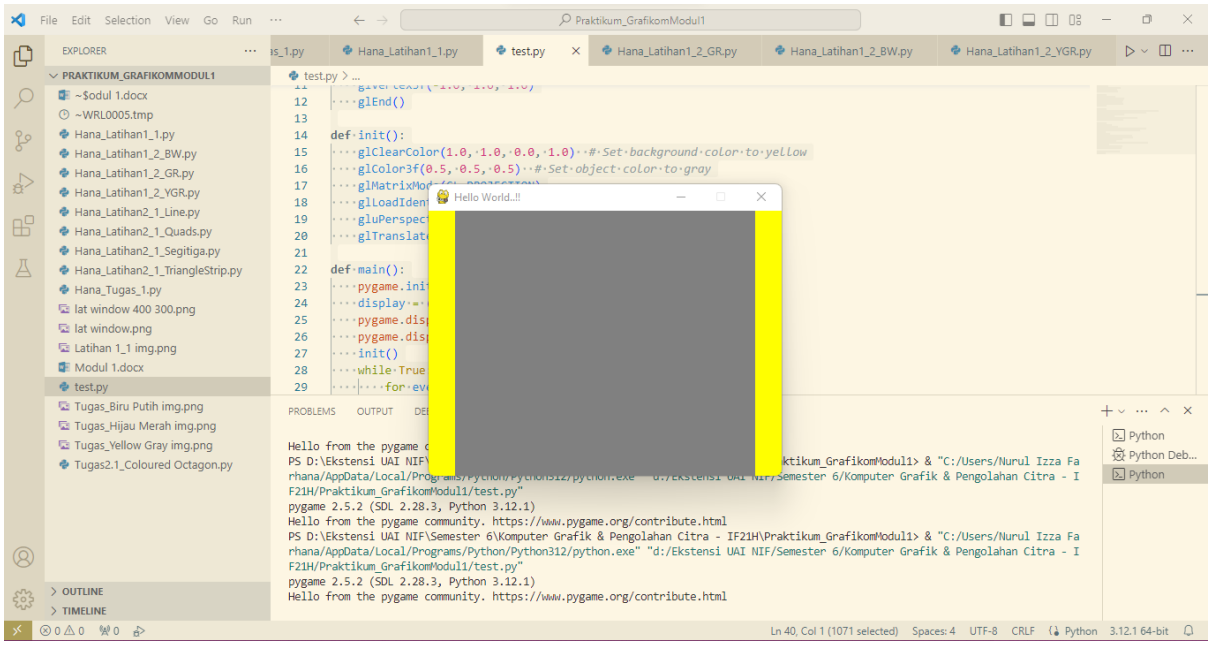
```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Cube():
    glBegin(GL_QUADS)
    glVertex3f(-0.8, -0.8, 0.8)
    glVertex3f(0.8, -0.8, 0.8)
    glVertex3f(0.8, 0.8, 0.8)
    glVertex3f(-0.8, 0.8, 0.8)
    glEnd()

def init():
    glClearColor(1.0, 1.0, 0.0, 1.0); # Set background color to yellow
    glColor3f(0.5, 0.5, 0.5);          # Set object color to gray
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();                    #load identity
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection
```

```
def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!!')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Cube()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```

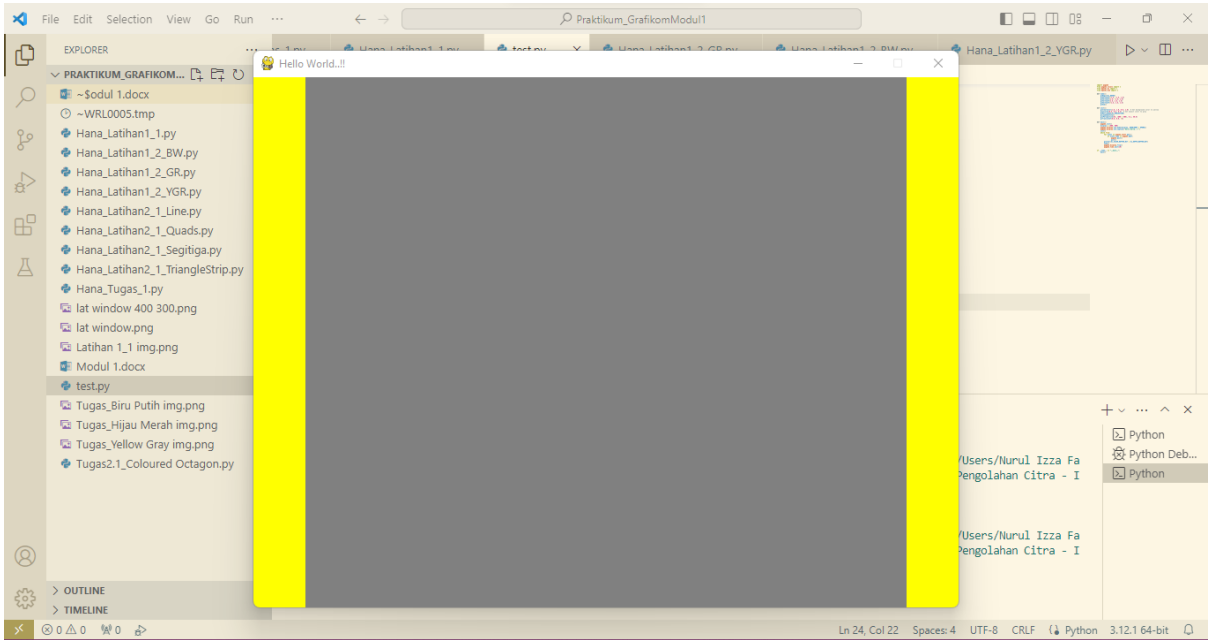


3. Ganti ukuran object dengan mengganti nilai vertex pada object QL_QUADS
4. Perbesar ukuran window dengan mengganti nilai pada glutInitWindowSize
5. Ganti nilai pada glutInitWindowPosition, lihat dan perhatikan perubahan yang terjadi pada Window

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Cube():
    glBegin(GL_QUADS)
    glVertex3f(-1.6, -1.6, 1.6)
    glVertex3f(1.6, -1.6, 1.6)
    glVertex3f(1.6, 1.6, 1.6)
    glVertex3f(-1.6, 1.6, 1.6)
    glEnd()

def init():
    glClearColor(1.0, 1.0, 0.0, 1.0)  # Set background color to yellow
    glColor3f(0.5, 0.5, 0.5)  # Set object color to gray
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45, (800 / 600), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)

def main():
    pygame.init()
    display = (400, 300)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!!')
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Cube()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```



Display Window ukuran 400, 600



Display Window ukuran 800, 600

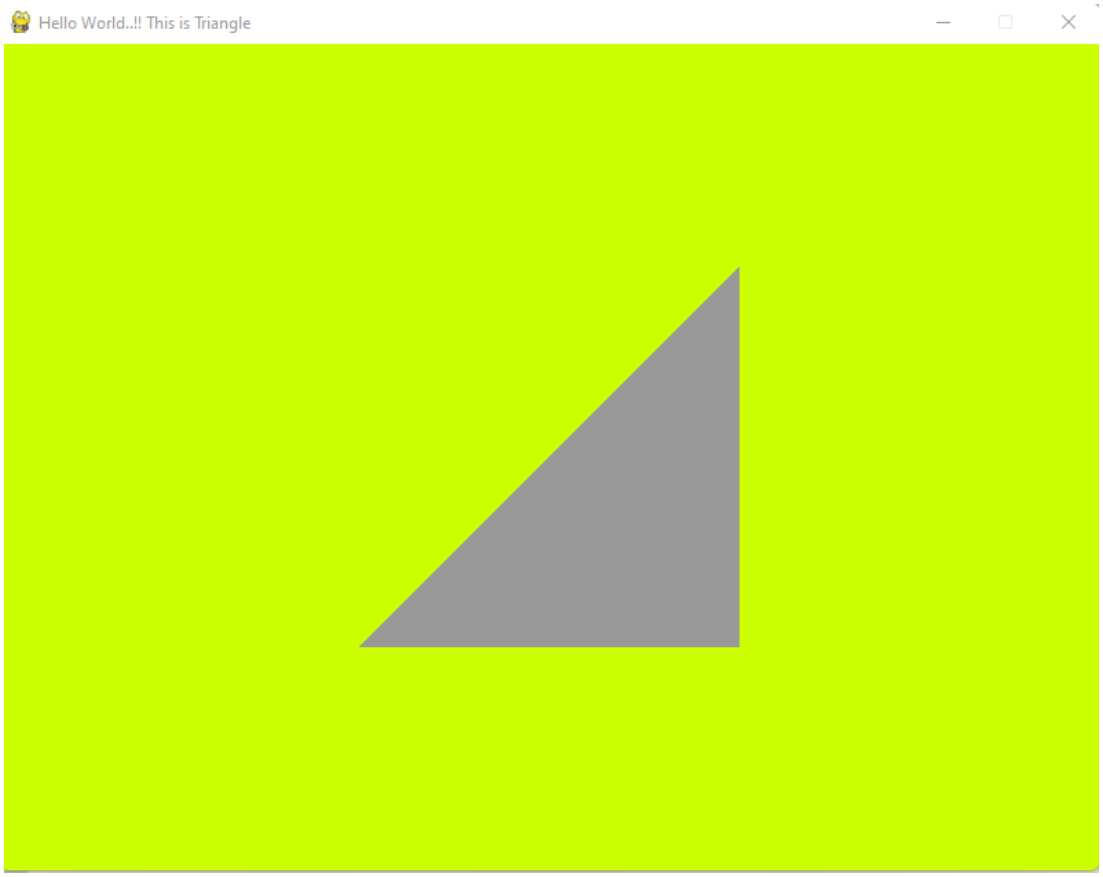
6. Kesimpulan

   Dalam latihan ini, kita memahami dasar penggunaan Pygame dan PyOpenGL untuk membuat dan menampilkan objek 3D sederhana dalam window. Fungsi-fungsi dasar seperti `glClearColor`, `glColor3f`, `glVertex2f`, `glutInitWindowSize`, dan `glutInitWindowPosition` memiliki peran krusial. `glClearColor` digunakan untuk mengubah warna background window, `glColor3f` untuk merubah warna objek, `glVertex2f` untuk mengubah koordinat objek, dan `glutInitWindowSize` dan `glutInitWindowPosition` untuk mengatur ukuran dan posisi window. Kesimpulannya, latihan ini memberikan dasar yang kuat untuk eksplorasi lebih lanjut dalam pengembangan grafis dengan Python dan OpenGL.

MODUL 2: Open GL Primitives

Ketiklah kode program berikut menggunakan notepad, simpan dengan nama Nama_latihan2_1.py Latihan 2_1. Segitiga

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Triangle():
    glBegin(GL_TRIANGLES)
    glVertex3f(-0.8, -0.8, 0.8)
    glVertex3f(0.8, -0.8, 0.8)
    glVertex3f(0.8, 0.8, 0.8)
    glVertex3f(-0.8, 0.8, 0.8)
    glEnd()

def init():
    glClearColor (0.8, 1.0, 0.0, 1.0);   #background colour
    glColor3f(0.6, 0.6, 0.6);            #object colour
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();                    #load identity
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection


def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!! This is Triangle')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Triangle()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```



Tugas:

1. Buatlah program untuk membuat jenis openGL primitives yang lain, dengan mengganti PRIMITIVES (GL_TRIANGLES) dengan primitives yang lain,yaitu:
   - GL_LINES

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Line():
    glBegin(GL_LINES)
    glVertex3f(-0.8, -0.8, 0.8)
    glVertex3f(0.8, -0.8, 0.8)
    glVertex3f(0.8, 0.8, 0.8)
    glVertex3f(-0.8, 0.8, 0.8)
    glEnd()

def init():
    glClearColor (0.8, 1.0, 0.0, 1.0);   #background colour
    glColor3f(0.6, 0.6, 0.6);            #object colour
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();                    #load identity
```

```
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection


def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!! This is Line')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
    glTranslatef(0.0, 0.0, -5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Line()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```

- GL_TRIANGLESTRIP

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def TriangleStrip():
glBegin(GL_TRIANGLE_STRIP)
glVertex3f(-0.8, -0.8, 0.8)
glVertex3f(0.8, -0.8, 0.8)
glVertex3f(0.8, 0.8, 0.8)
glVertex3f(-0.8, 0.8, 0.8)
glEnd()

def init():
glClearColor (0.8, 1.0, 0.0, 1.0);   #background colour
glColor3f(0.6, 0.6, 0.6);            #object colour
glMatrixMode (GL_PROJECTION);
glLoadIdentity();                    #load identity
glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection


def main():
pygame.init()
```

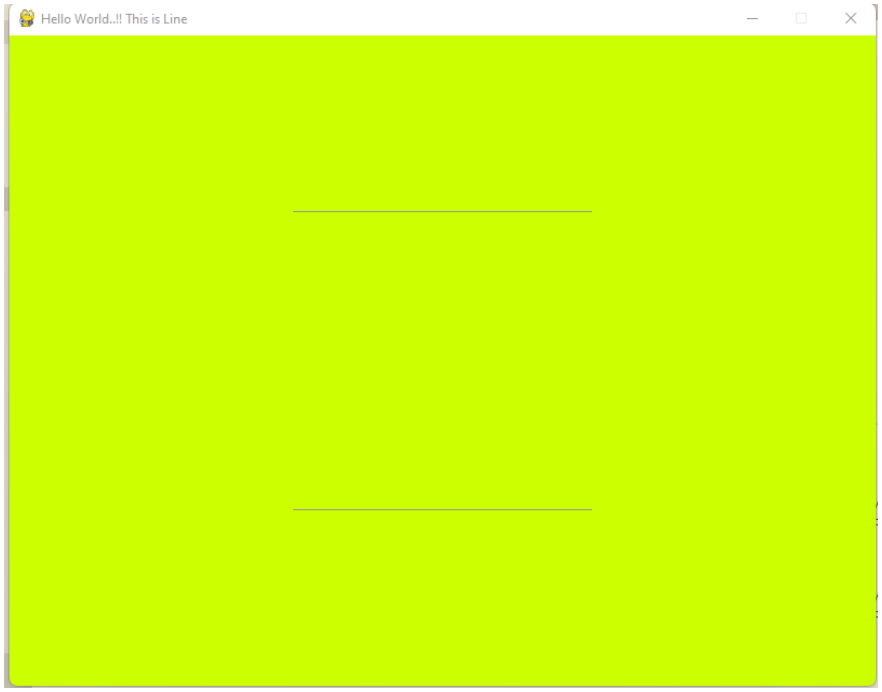
```
display = (800, 600)
pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
pygame.display.set_caption('Hello World..!!')
gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
glTranslatef(0.0, 0.0, -5)
init()
while True:
for event in pygame.event.get():
if event.type == pygame.QUIT:
pygame.quit()
quit()
glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
TriangleStrip()
pygame.display.flip()
pygame.time.wait(10)

if __name__ == "__main__":
main()
```

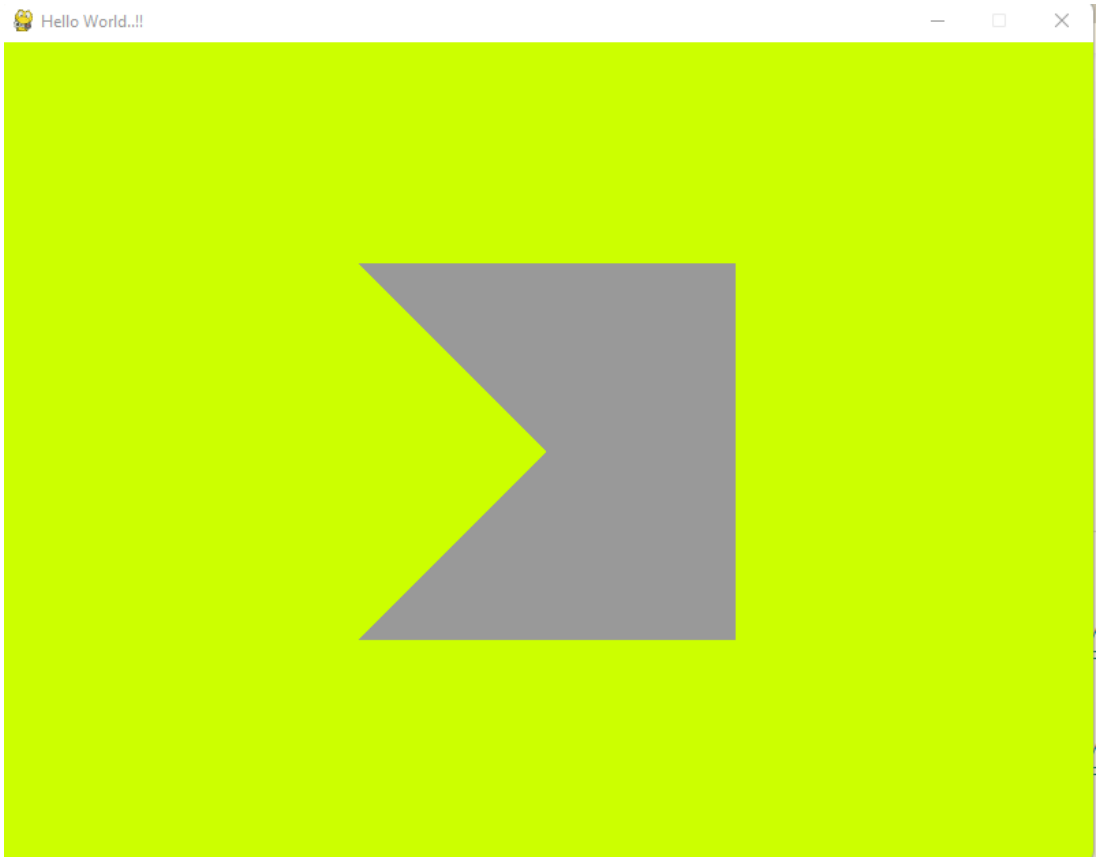
- GL_QUADS

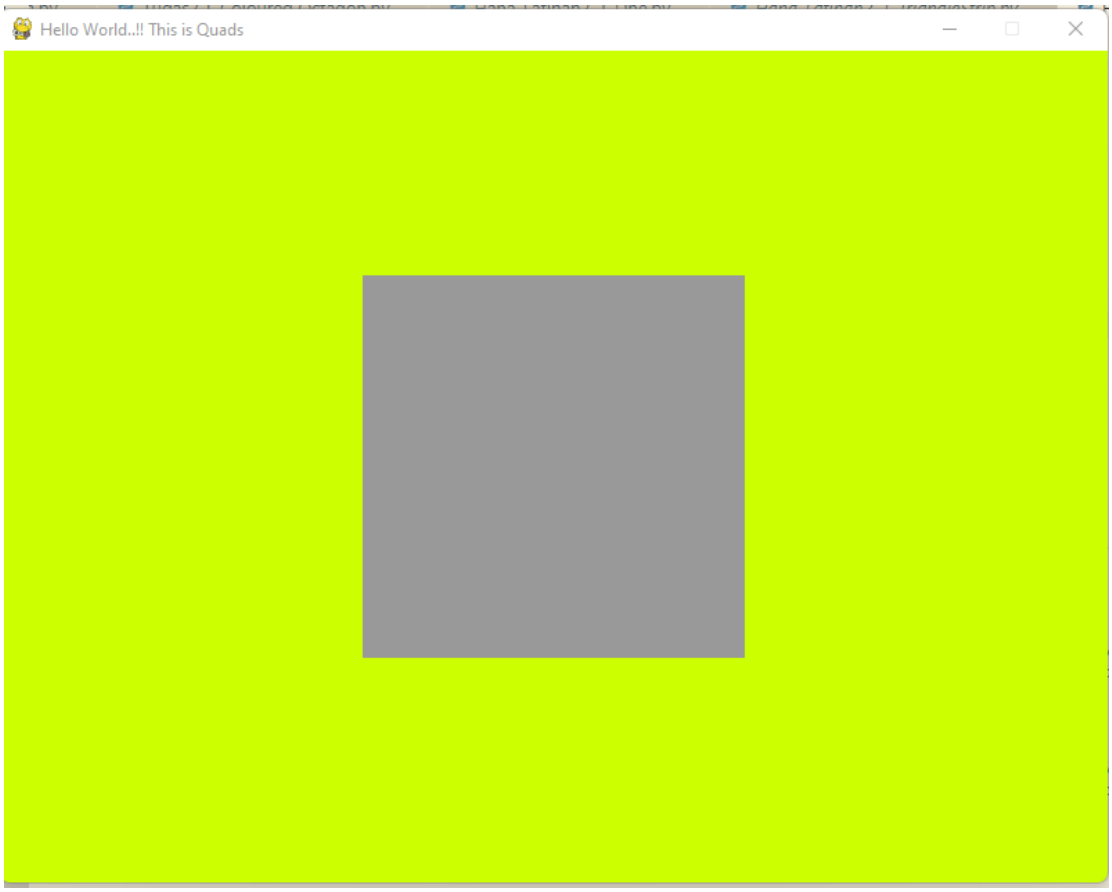
```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Quad():
    glBegin(GL_QUADS)
    glVertex3f(-0.8, -0.8, 0.8)
    glVertex3f(0.8, -0.8, 0.8)
    glVertex3f(0.8, 0.8, 0.8)
    glVertex3f(-0.8, 0.8, 0.8)
    glEnd()

def init():
    glClearColor (0.8, 1.0, 0.0, 1.0);   #background colour
    glColor3f(0.6, 0.6, 0.6);            #object colour
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();                    #load identity
    glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);   #projection


def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Hello World..!! This is Quads')
    gluPerspective(45, (display[0] / display[1]), 0.1, 50.0)
```

```
    glTranslatef(0.0, 0.0, -5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
        Quad()
        pygame.display.flip()
        pygame.time.wait(10)

if __name__ == "__main__":
    main()
```

2. Perhatikan urutan dari vertex untuk setiap jenis OpenGL Geometric Primitive, lampirkan output program.



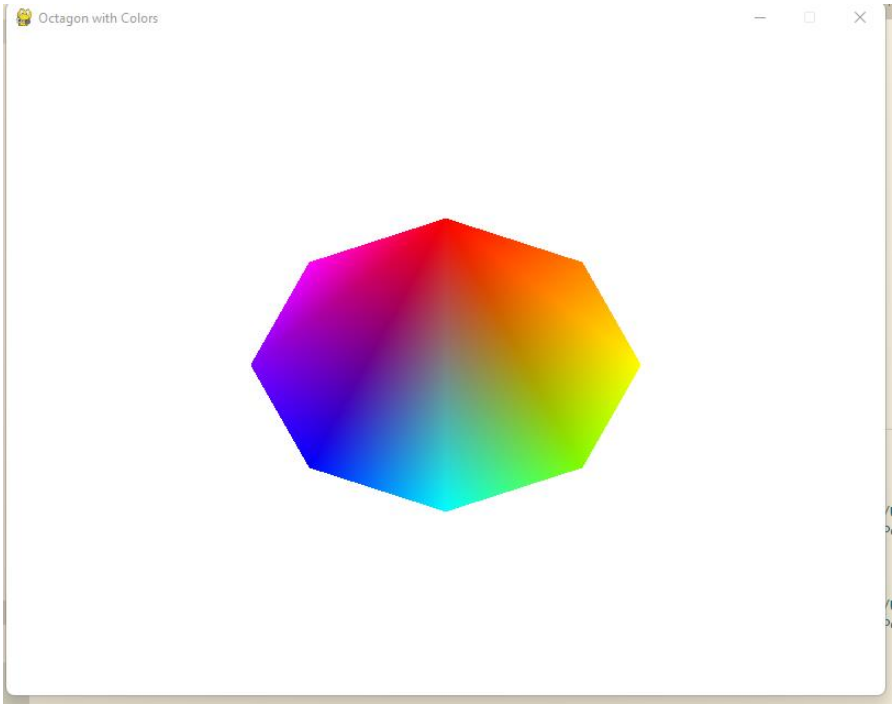

Hello World..!!


Hello World..!! This is Quads

3. Buatlah program untuk menghasilkan segi delapan berwarna, dengan menambahkan fungsi glcolor3f() simpan dengan nama Tugas2.1.

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *

def Octagon():
    glBegin(GL_POLYGON)
    glColor3f(1.0, 0.0, 0.0)  # Merah
    glVertex2f(0.0, 1.0)
    glColor3f(1.0, 0.5, 0.0)  # Oranye
    glVertex2f(0.7, 0.7)
    glColor3f(1.0, 1.0, 0.0)  # Kuning
    glVertex2f(1.0, 0.0)
    glColor3f(0.5, 1.0, 0.0)  # Hijau
    glVertex2f(0.7, -0.7)
    glColor3f(0.0, 1.0, 1.0)  # Cyan
    glVertex2f(0.0, -1.0)
    glColor3f(0.0, 0.0, 1.0)  # Biru
    glVertex2f(-0.7, -0.7)
    glColor3f(0.5, 0.0, 1.0)  # Ungu
    glVertex2f(-1.0, 0.0)
    glColor3f(1.0, 0.0, 1.0)  # Magenta
    glVertex2f(-0.7, 0.7)
    glEnd()

def init():
    glClearColor(1.0, 1.0, 1.0, 1.0)  # Set background color to white
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluOrtho2D(-1.5, 1.5, -1.5, 1.5)  # Projection

def main():
    pygame.init()
    display = (800, 600)
    pygame.display.set_mode(display, DOUBLEBUF | OPENGL)
    pygame.display.set_caption('Octagon with Colors')
    gluOrtho2D(-1.5, 1.5, -1.5, 1.5)
    init()
    while True:
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                pygame.quit()
                quit()
        glClear(GL_COLOR_BUFFER_BIT)
        Octagon()
```

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

if __name__ == "__main__":
    main()
```



4. Kesimpulan dari Modul 2

   Pemahaman konsep dasar dalam pembuatan objek grafis menggunakan titik, garis, dan poligon. Ketika membuat objek, perlu memperhatikan urutan dari setiap vertex agar bentuk objek yang dihasilkan sesuai.