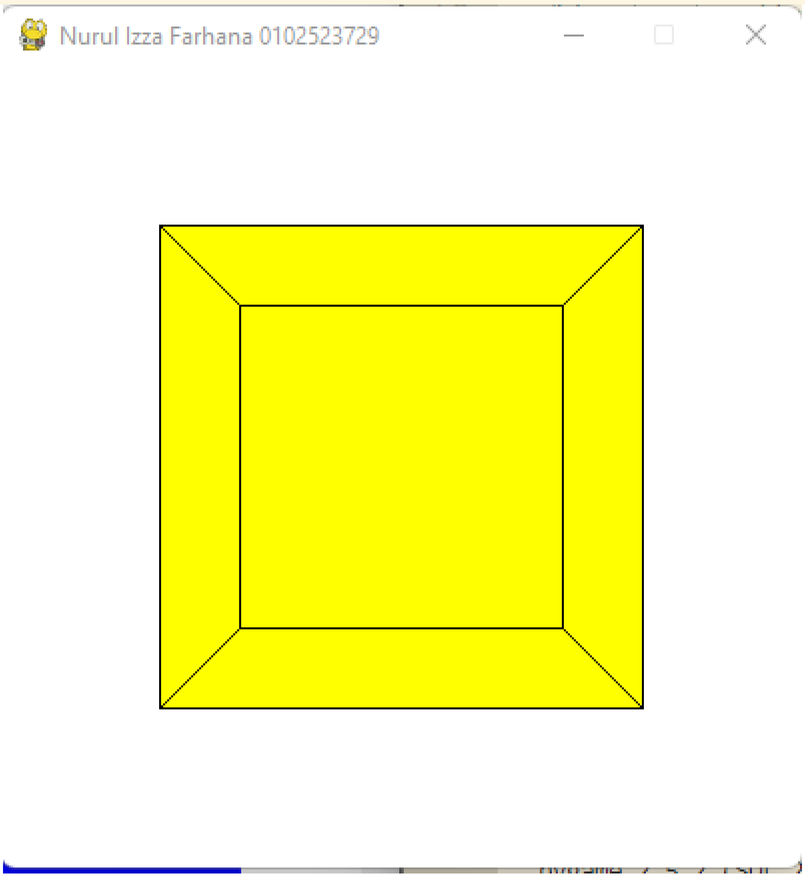
MODUL 5: 3D Vertex

Latihan 5_1: Kubus 3D

```
import pygame
from pygame.locals import *
from OpenGL.GL import *
from OpenGL.GLU import *
import numpy as np
import math


def plotfunc():
    squares = [
        [[-1,-1,1], [1,-1,1], [1,1,1], [-1,1,1]],
        [[1,-1,-1], [1,1,-1], [-1,1,-1],[-1,-1,-1]],
        [[-1,1,-1], [-1,1,1], [1,1,1],[1,1,-1]],

        [[-1,-1,-1], [1,-1,-1], [1,-1,1], [-1,-1,1]],
        [[1,-1,-1], [1,1,-1], [1,1,1],[1,-1,1]],
        [[-1,-1,-1], [-1,-1,1], [-1,1,1],[-1,1,-1]],
    ]

    glClear (GL_COLOR_BUFFER_BIT)
    glColor3f (0.0, 0.0, 0.0)
    glPointSize (1.0)
    for square in squares:
        glColor3f (1, 1, 0)
        glBegin(GL_POLYGON)
        for point in square:
            glVertex3f (point[0], point[1], point[2])
        glEnd()
        glColor3f (0, 0, 0)
        glBegin (GL_LINE_LOOP)
        for point in square:
            glVertex3f (point [0], point [1], point [2])
        glEnd()
    glFlush ()

def main():
    pygame.init()
    display = (400,400)
    pygame.display.set_caption('Nurul Izza Farhana 0102523729')
    pygame.display.set_mode(display, DOUBLEBUF|OPENGL)
    gluPerspective(45, (display[0]/display[1]), 0.1, 50.0)
    glTranslatef(0.0,0.0, -5)
    glClearColor(1.0, 1.0, 1.0, 1.0)
    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)
        plotfunc()
        pygame.display.flip()
        pygame.time.wait(10)

main()
```

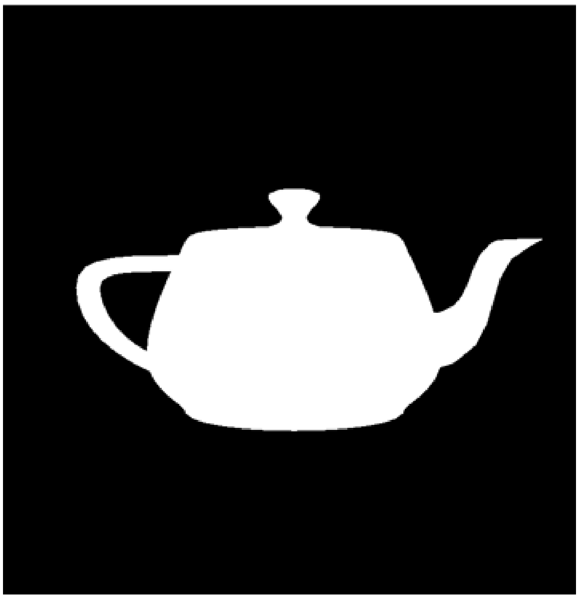


Latihan3_2: Bangunan_Sederhana

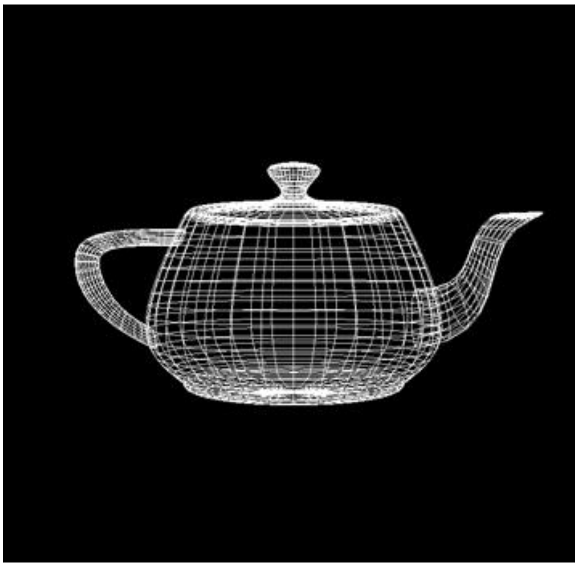
1. Gantilah objek kubus 3D dengan objek 3D menggunakan fungsi yang telah disediakan oleh GLUT:

```
import sys
import pygame
from screeninfo import get_monitors
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
from pygame.locals import *

def solidTeapot():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glutSolidTeapot(1.0)
    glutSwapBuffers()

def wireTeapot():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glutWireTeapot(1.0)
    glutSwapBuffers()

def solidCube():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glutSolidCube(1.0)
    glutSwapBuffers()

def wireCube():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glutWireCube(1.0)
    glutSwapBuffers()

def reshape(width, height):
    glViewport(0, 0, width, height)

    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45.0, width / height, 0.1, 100.0)
    glMatrixMode(GL_MODELVIEW)
    glLoadIdentity()
    gluLookAt(0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0)

def get_display_size():
    primary_monitor = get_monitors()[0]
    width = primary_monitor.width
    height = primary_monitor.height
    return [width, height]

def get_window_size(scale=1):
    display_size = get_display_size()
    size = round(display_size[0]*scale)
    if round(display_size[1]*scale) > size :
        size = round(display_size[1]*scale)
    return size

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)

    scale = 0.4
    size = get_window_size(scale)
    glutInitWindowSize(size,size)
    glutCreateWindow(b"Hana")
    glEnable(GL_DEPTH_TEST)
    glutDisplayFunc(solidCube)
    glutDisplayFunc(wireCube)
    glutDisplayFunc(wireTeapot)
    glutDisplayFunc(solidTeapot)
    glutReshapeFunc(reshape)
    glutMainLoop()

main()
```

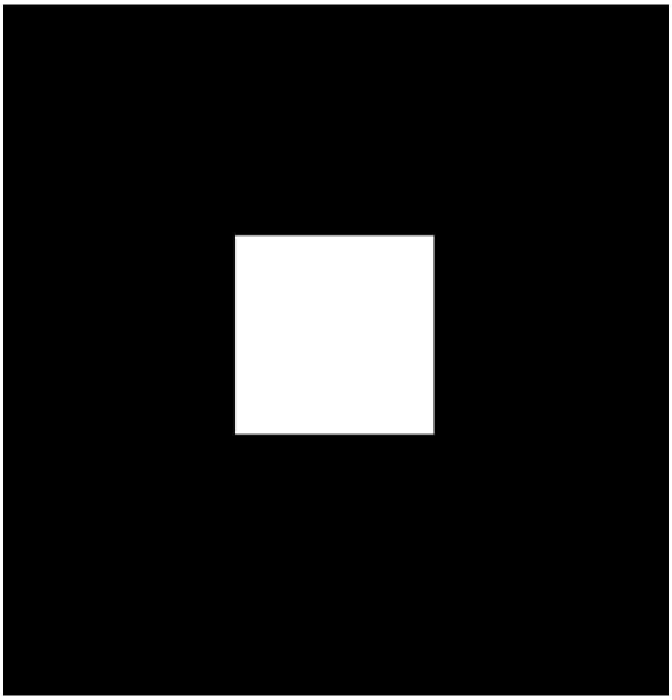
a. SolidTeapot



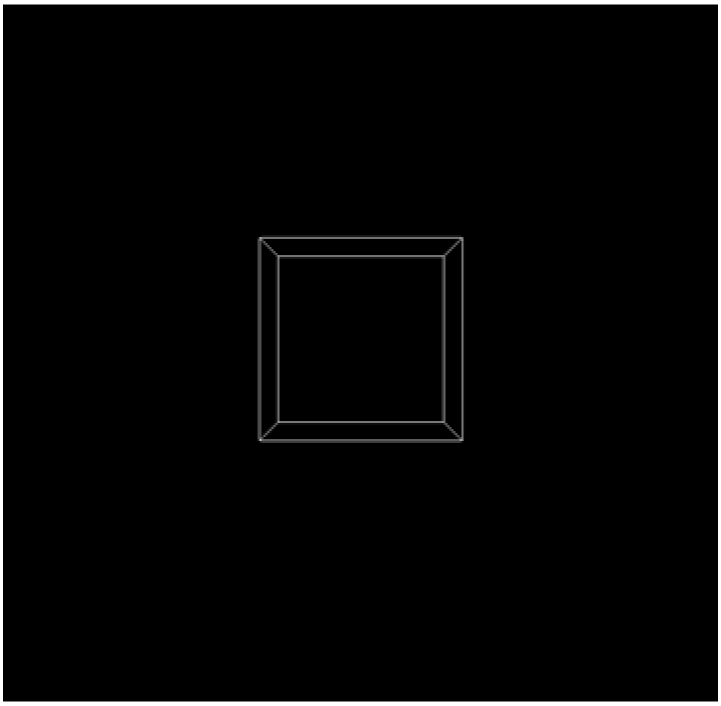
b. WireTeapot



c. **SolidCube**



d. **WireCube**



2. Kesimpulan dari latihan pada Modul 6

Dapat membuat object dengan menggunakan fungsi yang disediakan oleh module GLUT.

TUGAS

Buatlah sebuah bangunan yang menggunakan dua atau lebih bangun dasar yang ada (contoh: balok, limas, atau prisma)

```
from screeninfo import get_monitors
from OpenGL.GL import *
from OpenGL.GLUT import *
from OpenGL.GLU import *

def draw_cube(x, y, z, size):
    glPushMatrix()
    glTranslatef(x, y, z)
    glutSolidCube(size)
    glPopMatrix()

def draw_sphere(x, y, z, radius, slices=30, stacks=30):
    glPushMatrix()
    glTranslatef(x, y, z)
    glutSolidSphere(radius, slices, stacks)
    glPopMatrix()

def display():
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT)
    glLoadIdentity()
    gluLookAt(5, 5, 5, 0, 0, 0, 0, 1, 0)

    glColor3f(1.0, 0.0, 0.0)  # warna merah
    draw_cube(-1, 0, 0, 1)

    glColor3f(0.0, 0.0, 1.0)  # warna biru
    draw_cube(1, 0, 0, 1)

    glColor3f(0.0, 1.0, 0.0)  # warna hijau
    draw_sphere(0,0,0, 0.5)

    glutSwapBuffers()

def reshape(width, height):
    glViewport(0, 0, width, height)
    glMatrixMode(GL_PROJECTION)
    glLoadIdentity()
    gluPerspective(45, (width / height), 0.1, 100.0)
    glMatrixMode(GL_MODELVIEW)

def get_display_size():
    primary_monitor = get_monitors()[0]
    width = primary_monitor.width
    height = primary_monitor.height
    return [width, height]

def get_window_size(scale=1):
    display_size = get_display_size()
    size = round(display_size[0]*scale)
    if round(display_size[1]*scale) > size :
        size = round(display_size[1]*scale)
    return size


def main():
    scale = 0.4
    size = get_window_size(scale)

    glutInitWindowSize(size,size)
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH)
    glutCreateWindow(b"Hana")
    glEnable(GL_DEPTH_TEST)

    glutDisplayFunc(display)
    glutReshapeFunc(reshape)
    glutMainLoop()

main()
```