UTS Grafika Komputer

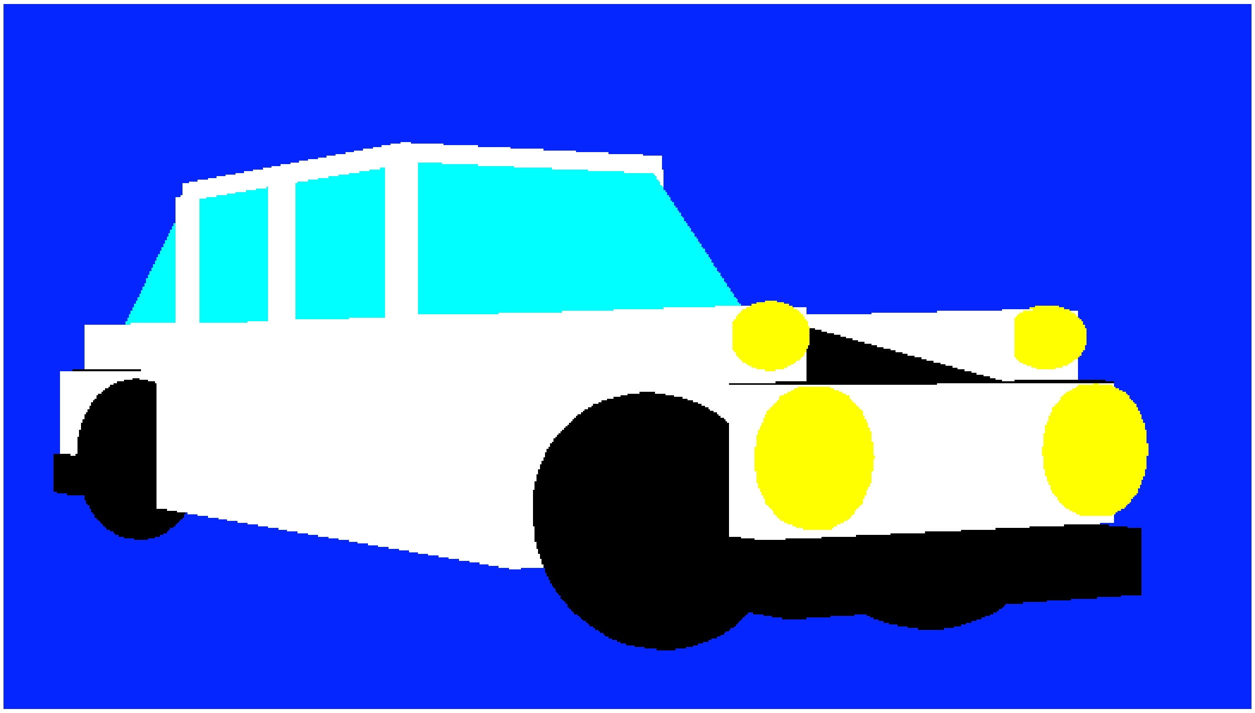
Anggota Kelompok:

1. ALvin Sutanto / C14200099
2. Hans Jeremy/ C14200034
3. Toni Ariyanto / C14200121
4. Reiner Julio / C14200136

Link github : https://github.com/ToniAriyanto/grafkomUTS.git

Link Video : https://youtu.be/KjFGrmfF_6M

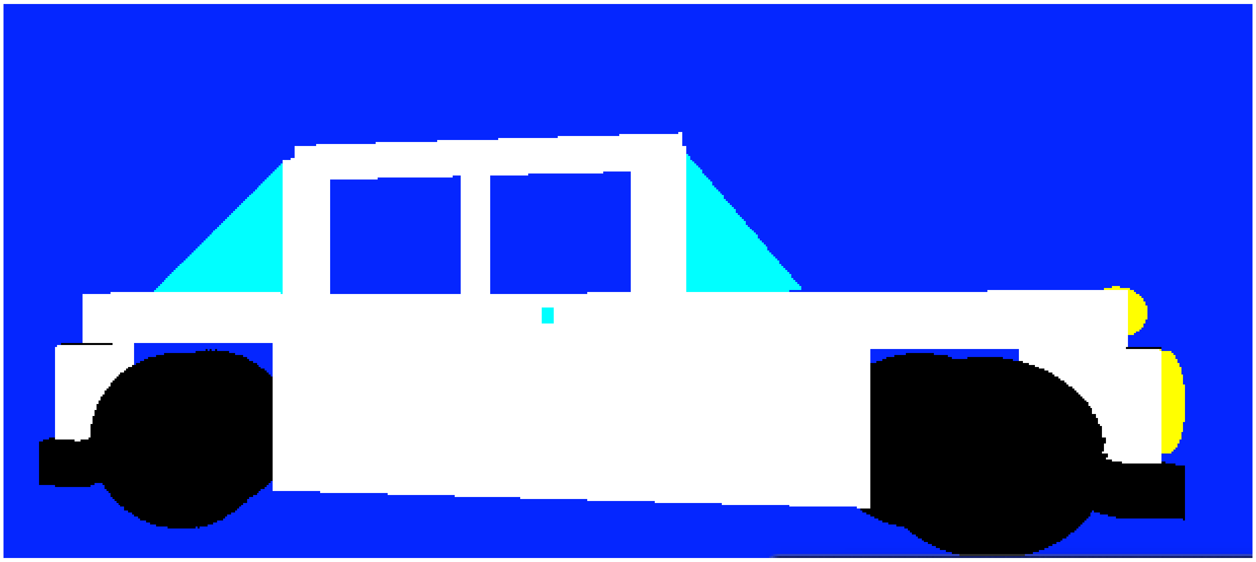
Object Toni



Terdiri dari:
4 torus - Ban
16 rectangle/Box
3 cube modif
2 sphere sebagai lampu
2 cylinder/tabung sebagai lampu foglamp

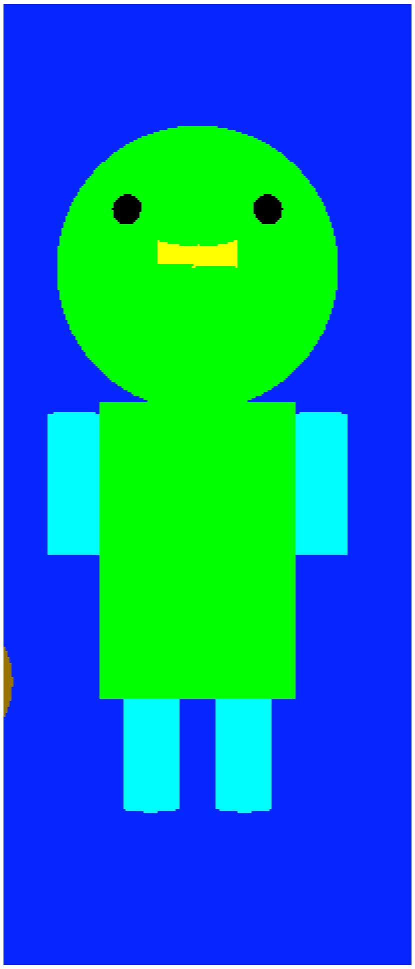
Disini saya membuat mobil. Saya menggunakan rectangle,box,circle,cylinder,torus,sphere, dan kubus yang di modif menjadi bentuk segitiga

Di objek mobil ini,saya dapat melakukan rotasi pada sumbu sendiri dan juga dapat rotasi terhadap sumbu pusat,bisa juga melakukan translasi untuk maju mundur terhadap sumbu x,y dan juga Z ,saya juga bisa membuka kaca mobil



Dengan cara mengscale

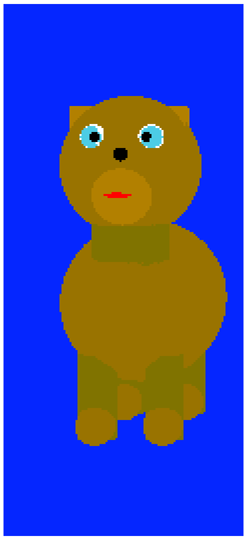
Objek Hans



Terdiri dari
Kepala yang terbuat dari sphere
Badan yang terbuat dari cuboid
2 buah mata yang terbuat dari sphere
2 buah tangan yang terbuat dari cylinder
2 buah kaki yang terbuat dari cylinder
Hidung yang terbuat dari cube modif

Transformasi:
Menggunakan tombol arrow up and down untuk maju mundur
Menggunakan tombol arrow left and right untuk bergerak ke kiri dan kanan
Menggunakan tombol angka 0 untuk rotate clockwise
Menggunakan tombol angka 9 untuk rotate counterclockwise
Setelah translasi maju mundur kiri kanan, objek masih dapat rotate di sumbunya sendiri

Object Alvin - Hewan



Bagian Kepala

Terdiri dari

- Kepala besar = terbuat dari sebuah sphere
- 2 buah telinga = terbuat dari 2 buah cuboid yang dimasukkan ke dalam bagian kepala sehingga berbentuk telinga
- 2 bola mata putih = terbuat dari 2 buah sphere
- 2 bola mata biru = terbuat dari 2 buah sphere
- 2 pupil mata hitam = terbuat dari 2 buah sphere
- Hidung = terbuat dari 1 buah sphere
- Bagian luar mulut = terbuat dari 1 buah sphere
- Bagian lidah = terbuat dari sphere yang ditipiskan
- Leher = terbuat dari silinder/tabung

Bagian tubuh

- Tubuh = terbuat dari sphere yang tarik ke belakang sehingga berbentuk lonjong

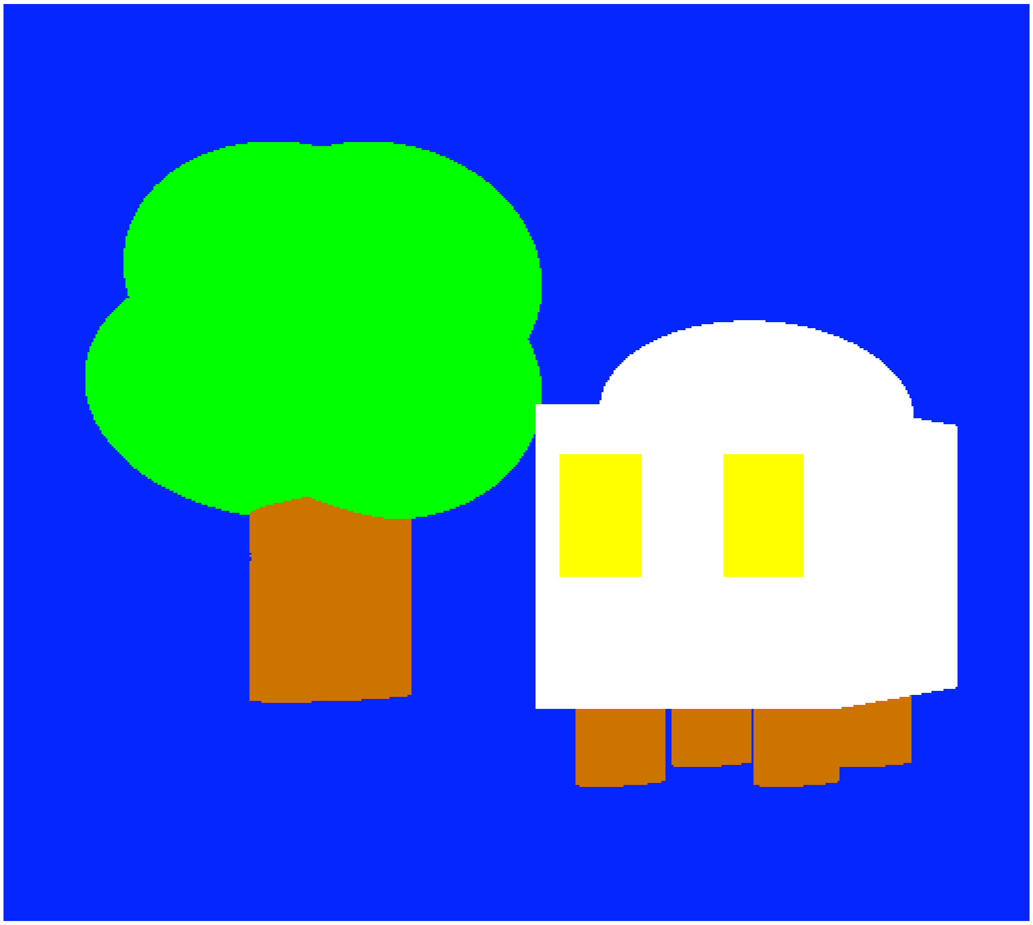
Bagian tiap kaki

- Lengan kaki = terbuat dari silinder
- Telapak kaki = terbuat dari sphere

Transformasi dari hewan tersebut

- Menggunakan scaling sehingga dapat menyesuaikan ukuran hewan dengan lingkungan di sekitarnya
- Menggunakan translasi dengan menekan IJKL untuk bergerak maju, mundur, ke kanan, dan ke kiri
- Menggunakan rotasi dengan menekan X untuk berputar counter clockwise
- Setelah melakukan translasi, hewan masih dapat berotasi sesuai titik pusat hewan sehingga hewan masih dapat berputar sesuai titik pusatnya di manapun

Object Reiner - Rumah dan Pohon



Di objek bangunan rumah ini,bangunan bisa melakukan rotasi pada sumbu sendiri dan bisa rotasi terhadap sumbu pusat. Selain itu bangunan yang saya buat bisa melakukan translasi untuk maju mundur terhadap sumbu x, y dan z.
BANGUNAN

Bagian Atap :

- Bagian atap saya buat menggunakan setengah sphere

Bagian Bangunan :

- Bangunannya saya buat menggunakan cuboid
- Jendela nya menggunakan rectangle

Bagian Penahan :

- Penahan rumah atau kaki yang menahan rumahnya menggunakan cylinder

POHON

- Bagian daun menggunakan sphere
- Bagian batang pohon menggunakan cylinder

Draft code

```
import Engine.*;
import Engine.Object;
import org.joml.Vector2f;
import org.joml.Vector3f;
import org.joml.Vector4f;
import org.lwjgl.opengl.GL;


import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;


import static org.lwjgl.glfw.GLFW.*;
import static org.lwjgl.opengl.GL11.glClearColor;
import static org.lwjgl.opengl.GL30.*;


public class Main {
    private Window window =
            new Window
                    (1920,  1080,"Hello World");


    private List<Vector3f> controlPoints = new ArrayList<>();


    public Main(Vector3f... points) {
        for (Vector3f point : points) {
            controlPoints.add(point);
        }
    }
    private ArrayList<Object> objects
```

```
        = new ArrayList<>();
private ArrayList<Object> objectsRectangle
        = new ArrayList<>();

private ArrayList<Object> objectAlvin
        = new ArrayList<>();

private ArrayList<Object> objectHans
        = new ArrayList<>();
private ArrayList<Object> objectToni
        = new ArrayList<>();

private ArrayList<Object> objectTanganHans
        = new ArrayList<>();

private ArrayList<Object> objectKakiHans
        = new ArrayList<>();

private ArrayList<Object> objectMata
        = new ArrayList<>();
private ArrayList<Object> objectBan
        = new ArrayList<>();

private ArrayList<Object> environment = new ArrayList<>();

private ArrayList<Object> objectKepalaAlvin
        = new ArrayList<>();

private ArrayList<Object> objectKaki1Alvin
        = new ArrayList<>();

private ArrayList<Object> objectKaki2Alvin
        = new ArrayList<>();
```

```
private ArrayList<Object> objectKaki3Alvin
        = new ArrayList<>();

private ArrayList<Object> objectKaki4Alvin
        = new ArrayList<>();

private ArrayList<Object> objectPohon
        = new ArrayList<>();

private ArrayList<Object> objectReiner
        = new ArrayList<>();

private ArrayList<Object> objectAtap
        = new ArrayList<>();

private ArrayList<Object> objectJendela
        = new ArrayList<>();

private ArrayList<Object> objectKaki
        = new ArrayList<>();

private ArrayList<Object> objectsPointsControl
        = new ArrayList<>();

private MouseInput mouseInput;

private boolean shouldTranslate = true;

float lastFrameTime = 0;
double rotationSpeed = 1;
int countDegree = 0;

Projection projection = new Projection(window.getWidth(),window.getHeight());
```

```
    Camera camera = new Camera();
    public void init() {
        window.init();
        GL.createCapabilities();
        mouseInput = window.getMouseInput();
        camera.setPosition(0, 0, 1);
//        camera.setRotation((float) Math.toRadians(0.0f), (float) Math.toRadians(-15.0f));
        //code
//        objects.add(new Object2d(
//            Arrays.asList(
//                //shaderFile lokasi menyesuaikan objectnya
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/scene.vert"
//                , GL_VERTEX_SHADER),
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/scene.frag"
//                , GL_FRAGMENT_SHADER)
//            ),
//            new ArrayList<>(
//                List.of(
//                    new Vector3f(0.0f,0.5f,0.0f),
//                    new Vector3f(-0.5f,-0.5f,0.0f),
//                    new Vector3f(0.5f,-0.5f,0.0f)
//                )
//            ),
//            new Vector4f(0.0f,1.0f,1.0f,1.0f)
//        ));
//        objects.add(new Object(
//            Arrays.asList(
//                //shaderFile lokasi menyesuaikan objectnya
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/" +
//                    "sceneWithVerticesColor.vert"
//                        , GL_VERTEX_SHADER),
//                new ShaderProgram.ShaderModuleData
//                    ("resources/shaders/" +
//                    "sceneWithVerticesColor.frag"
//                            , GL_FRAGMENT_SHADER)
//        ),
//        new ArrayList<>(
//                List.of(
//                    new Vector3f(0.0f,0.5f,0.0f),
//                    new Vector3f(-0.5f,-0.5f,0.0f),
//                    new Vector3f(0.5f,-0.5f,0.0f)
//                )
//            ),
//        new ArrayList<>(
//            List.of(
//                new Vector3f(1.0f,0.0f,0.0f),
//                new Vector3f(0.0f,1.0f,0.0f),
//                new Vector3f(0.0f,0.0f,1.0f)
//            )
//        )
//        ));
//        objectsRectangle.add(new Rectangle(
//            Arrays.asList(
//                //shaderFile lokasi menyesuaikan objectnya
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/scene.vert"
//                , GL_VERTEX_SHADER),
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/scene.frag"
//                , GL_FRAGMENT_SHADER)
//            ),
//            new ArrayList<>(
//                List.of(
//                    new Vector3f(0.0f,0.0f,0.0f),
//                    new Vector3f(0.5f,0.0f,0.0f),
//                    new Vector3f(0.0f,0.5f,0.0f),
//                    new Vector3f( 0.5f,0.5f,0.0f)
//                )
//            ),
//            new Vector4f(0.0f,1.0f,1.0f,1.0f),
//            Arrays.asList(0,1,2,1,2,3)
//
//        ));
//        objectsPointsControl.add(new Object(
//            Arrays.asList(
//                //shaderFile lokasi menyesuaikan objectnya
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/scene.vert"
//                , GL_VERTEX_SHADER),
//                new ShaderProgram.ShaderModuleData
//                ("resources/shaders/scene.frag"
//                , GL_FRAGMENT_SHADER)
//            ),
//          new ArrayList<>(),
//          new Vector4f(0.0f,1.0f,1.0f,1.0f)
//      ));
//      objects.add(new Cuboid(
//              Arrays.asList(
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
//              ),
//              new ArrayList<>(),
//              new Vector4f(0.0f,1.0f,0.0f,1.0f),
//              Arrays.asList(0.0f,0.0f,0.0f),
//              0.125f,
//              0.125f,
//              0.125f,
//              36,
//              18
//      ));
////        objects.get(0).translateObject(0.5f,0.0f,0.0f);
//      objects.get(0).scaleObject(2f,2f,2f);
//
//      objects.get(0).getChildObject().add(new Sphere(
//              Arrays.asList(
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
//              ),
//              new ArrayList<>(),
//              new Vector4f(0.0f,1.0f,0.0f,1.0f),
//              Arrays.asList(0.0f,0.0f,0.0f),
//              0.125f,
//              0.125f,
//              0.125f,
//              36,
//              18
//      ));
//      objects.get(0).getChildObject().get(0).translateObject(0.25f,0.0f,0.0f);
////        objects.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(0.25f,0.0f,0.0f));
//
//      objects.get(0).getChildObject().add(new Cylinder(
////                Arrays.asList(
////                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
////                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
////            ),
////            new ArrayList<>(),
////            new Vector4f(0.0f,1.0f,1.0f,1.0f),
////            Arrays.asList(0.0f,0.0f,0.0f),
////            0.125f,
////            0.125f,
////            0.125f,
////            36,
////            18
////        ));
//      objects.get(0).getChildObject().get(1).translateObject(0.5f,0.0f,0.0f);
//      objects.get(0).getChildObject().get(1).scaleObject(0.5f,0.5f,0.5f);
////        objects.get(0).getChildObject().get(1).setCenterPoint(Arrays.asList(0.5f,0.0f,0.0f));
//
//      objects.get(0).getChildObject().get(1).getChildObject().add(new Sphere(
//              Arrays.asList(
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
//              ),
//              new ArrayList<>(),
//              new Vector4f(0.0f,1.0f,0.0f,1.0f),
//              Arrays.asList(0.0f,0.0f,0.0f),
//              0.125f,
//              0.125f,
//              0.125f,
//              36,
//              18
//      ));
//      objects.get(0).getChildObject().get(1).getChildObject().get(0).scaleObject(0.5f,0.5f,0.5f);
//
objects.get(0).getChildObject().get(1).getChildObject().get(0).translateObject(0.5f,-0.1f,0.0f);
//
objects.get(0).getChildObject().get(1).getChildObject().get(0).setCenterPoint(Arrays.asList(0.5f,-
0.1f,0.0f));


//      environment.add(new Cuboid(
//              Arrays.asList(
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
//              ),
//              new ArrayList<>(),
//              new Vector4f(0.0f, 1.0f, 0.0f, 1.0f),
//              Arrays.asList(0f,0.5f,-0.5f),
//              0.25f,
//              0.25f,
//              0.25f,
//              36,
//              18
//      ));


//      objectAlvin.add(new Sphere(
//              Arrays.asList(
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
//                      new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
//              ),
//              new ArrayList<>(),
//              new Vector4f(0.0f,1.0f,0.0f,1.0f),
//              Arrays.asList(0.0f,0.0f,0.0f),
//              0.125f,
//              0.125f,
//              0.125f,
//              36,
//              18
//      ));


//      objectTanganHans.get(0).rotateObject((float)Math.toRadians(0.5f),0.0f,0.0f,1.0f);


//      POHON




        objectPohon.add(new Cylinder(
                Arrays.asList(
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
                ),
                new ArrayList<>(),
                new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
                Arrays.asList(-0.73f,0.0f,0.0f),
                0.035f,
                0.035f,
                0.035f,
                7200,
                36,0.15f




        ));


        objectPohon.add(new Sphere(
                Arrays.asList(
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
                ),
                new ArrayList<>(),
                new Vector4f(0.0f,1.0f,0.0f,0.0f),
                Arrays.asList(-0.75f,0.09f,0.0f),
                0.085f,
                0.07f,
                0.085f,
                500,
                1000
        ));


        objectPohon.add(new Sphere(
                Arrays.asList(
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
```

```
            ),
            new ArrayList<>(),
            new Vector4f(0.0f,1.0f,0.0f,0.0f),
            Arrays.asList(-0.7f,0.09f,0.0f),
            0.07f,
            0.07f,
            0.07f,
            500,
            1000
        ));


        objectPohon.add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f,1.0f,0.0f,0.0f),
            Arrays.asList(-0.7f,0.15f,0.0f),
            0.07f,
            0.07f,
            0.07f,
            500,
            1000
        ));


        objectPohon.add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f,1.0f,0.0f,0.0f),
            Arrays.asList(-0.75f,0.15f,0.0f),
            0.07f,
            0.07f,
            0.07f,
            500,
            1000
        ));


        //BANGUNAN


        objectReiner.add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f,1.0f,1.0f,1.0f),
            Arrays.asList(0.0f,0.0f,0.0f),
            0.15f,
            0.15f,
            0.15f,
            36,
            18
        ));


        objectReiner.get(0).translateObject(-0.5f,0.0f,0.0f);
        objectReiner.get(0).setCenterPoint(Arrays.asList(-0.5f,0.0f,0.0f));


        objectReiner.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.02f,
                0.15f, 1.0f,
                0.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.0f,
            0.0f,
            0.0f,
            36,
            18
        ));
        objectReiner.get(0).getChildObject().get(0).translateObject(-0.5f, 0.0f, 0.0f);
        objectReiner.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.5f, -0.5f, 0.0f));


        //JENDELA


        objectReiner.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.02f,
                0.15f, 1.0f,
                0.0f),
            Arrays.asList(-0.46f,0.02f,0.079f),
            0.0f,
            0.0f,
            0f,
            36,
            18
        ));


        objectReiner.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0.0f, 0.0f),
            Arrays.asList(-0.46f,0.02f,0.079f),
            0.04f,
            0.06f,
            0f,
            36,
            18
        ));


        objectReiner.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0.0f, 0.0f),
            Arrays.asList(-0.54f,0.02f,0.079f),
            0.04f,
            0.06f,
            0f,
            36,
            18
        ));
        objectReiner.get(0).getChildObject().get(0).translateObject(-0.46f, 0.0f, 0.0f);
        objectReiner.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.46f, -0.46f,
0.0f));
        objectReiner.get(0).getChildObject().get(1).translateObject(-0.54f, 0.0f, 0.0f);
        objectReiner.get(0).getChildObject().get(1).setCenterPoint(Arrays.asList(-0.54f, -0.54f,
0.0f));


        //ATAP


        objectReiner.add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f,1.0f,1.0f,1.0f),
            Arrays.asList(0.0f,0.0f,0.0f),
            0.075f,
            0.05f,
            0.075f,
            500,
            1000
        ));
        objectReiner.get(1).translateObject(-0.5f,0.075f,0.0f);
        objectReiner.get(1).setCenterPoint(Arrays.asList(-0.5f,0.075f,0.0f));


        objectReiner.get(1).getChildObject().add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert", GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag", GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f,1.0f,1.0f,1.0f),
            Arrays.asList(-0.5f,0.075f,0.0f),
            0.075f,
            0.05f,
            0.075f,
            500,
            1000
        ));
        objectReiner.get(1).getChildObject().get(0).translateObject(0.0f, 0.0f, 0.0f);
        objectReiner.get(1).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.5f, -0.5f, 0.0f));


        //KAKI


        objectKaki.add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert", GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag", GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
            Arrays.asList(0.0f,0.0f,0.0f),
            0.02f,
            0.02f,
            0.02f,
            7200,
            36,0.05f




        ));
          objectKaki.get(0).translateObject(-0.55f,-0.09f,0.044f);
          objectKaki.get(0).setCenterPoint(Arrays.asList(-0.55f,-0.09f,0.44f));


        objectKaki.get(0).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
            Arrays.asList(0.0f,0.0f,0.0f),
            0.02f,
            0.02f,
            0.02f,
            7200,
            36,0.05f
        ));


        objectKaki.get(0).getChildObject().get(0).translateObject(-0.46f,-0.09f, 0.044f);
        objectKaki.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.46f, -0.09f,
    0.044f));


        objectKaki.get(0).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
            Arrays.asList(0.0f,0.0f,0.0f),
            0.02f,
                0.02f,
                0.02f,
                7200,
                36,0.05f
        ));


        objectKaki.get(0).getChildObject().get(1).translateObject(-0.46f,-0.09f, -0.044f);
        objectKaki.get(0).getChildObject().get(1).setCenterPoint(Arrays.asList(-0.46f, -0.09f,
    -0.044f));


        objectKaki.get(0).getChildObject().add(new Cylinder(
                Arrays.asList(
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
                ),
                new ArrayList<>(),
                new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
                Arrays.asList(0.0f,0.0f,0.0f),
                0.02f,
                0.02f,
                0.02f,
                7200,
                36,0.05f
        ));


        objectKaki.get(0).getChildObject().get(2).translateObject(-0.55f,-0.09f, -0.044f);
        objectKaki.get(0).getChildObject().get(2).setCenterPoint(Arrays.asList(-0.55f, -0.09f,
    -0.044f));


        /*objectReiner.add(new Cylinder(
                Arrays.asList(
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
                ),
                new ArrayList<>(),
                new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
            Arrays.asList(-0.68f,-0.1f,0.0f),
            0.02f,
            0.02f,
            0.02f,
            7200,
            36,0.04f
        ));


        objectReiner.get(2).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.8f, 0.45f, 0.0f, 1.0f),
            Arrays.asList(-0.46f,-0.1f,0.0f),
            0.02f,
            0.02f,
            0.02f,
            7200,
            36,0.05f
        ));


        objectReiner.add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f,1.0f,1.0f,1.0f),
            Arrays.asList(-0.50f,-0.078f,0.0f),
            0.3f,
            0.01f,
            0.28f,
            36,
            18
        ));*/
        //obj 1


        objectHans.add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.15f, 0.0f),
            0.075f,
            0.075f,
            0.075f,
            1200,
            1200
        ));


        objectHans.get(0).getChildObject().add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.0075f,
            0.0075f,
            0.0075f,
            36,
            18
        ));


        objectHans.get(0).getChildObject().add(new Sphere(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.0075f,
            0.0075f,
            0.0075f,
            36,
            18
        ));


        objectHans.get(0).getChildObject().add(new Cuboidmodiv(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.02f,
            0.01f,
            0.0075f,
            36,
            18
        ));
        objectHans.get(0).getChildObject().get(0).translateObject(0.035f, 0.17f, 0.065f);
        objectHans.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(0.035f, 0.17f,
    0.065f));
        objectHans.get(0).getChildObject().get(1).translateObject(-0.035f, 0.17f, 0.065f);
        objectHans.get(0).getChildObject().get(1).setCenterPoint(Arrays.asList(-0.035f, 0.17f,
    0.065f));
        objectHans.get(0).getChildObject().get(2).scaleObject(2f,2f,2f);
        objectHans.get(0).getChildObject().get(2).translateObject(0.0f, 0.15f, 0.075f);
        objectHans.get(0).getChildObject().get(2).setCenterPoint(Arrays.asList(0.0f, 0.15f, 0.075f));


        objectHans.add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 0.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.1f,
            0.15f,
            0.1f,
            36,
            18
        ));


        objectHans.get(1).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.015f,
            0.015f,
            0.015f,
            1000,
            3600, 0.075f
        ));
        objectHans.get(1).getChildObject().get(0).translateObject(-0.025f, -0.1f, 0.0f);
        objectHans.get(1).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.025f, -0.025f,
    0.0f));


        objectHans.get(1).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.015f,
            0.015f,
            0.015f,
            1000,
            3600, 0.075f
        ));
        objectHans.get(1).getChildObject().get(1).translateObject(0.025f, -0.1f, 0.0f);
        objectHans.get(1).getChildObject().get(1).setCenterPoint(Arrays.asList(0.025f, -0.025f,
    0.0f));


        objectHans.get(1).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.015f,
            0.015f,
            0.015f,
            1000,
            3600, 0.075f
        ));
        objectHans.get(1).getChildObject().get(2).translateObject(0.065f, 0.035f, 0.0f);
        objectHans.get(1).getChildObject().get(2).setCenterPoint(Arrays.asList(0.065f, 0.2f, 0.0f));


        objectHans.get(1).getChildObject().add(new Cylinder(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.015f,
            0.015f,
            0.015f,
            1000,
            3600, 0.075f
        ));
        objectHans.get(1).getChildObject().get(3).translateObject(-0.065f, 0.035f, 0.0f);
        objectHans.get(1).getChildObject().get(3).setCenterPoint(Arrays.asList(-0.065f, 0.2f, 0.0f));




        //obj1
        //base badan mobil
        objectToni.add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f,0f,0f),
            0.25f,
            0.15f,
            0.4f,
            36,
            18
        ));
        //obj2
        objectToni.get(0).getChildObject().add(new Cuboidmodiv(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.2f,
            0.1f,
            0.075f,
            36,
            18
        ));
        //obj 3
        objectToni.get(0).getChildObject().add(new Cuboidmodiv(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0f, 0f, 0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.15f,
            0.04f,
            0.15f,
            36,
            18
        ));
        //obj 4
        //penyambung
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.001f,
            0.02f,
            36,
            18
        ));
        //panjangan bemper obj 5
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.05f,
            0.02f,
            36,
            18
        ));
        //panjangan bumper obj6
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.04f,
            0.02f,
            36,
            18
        ));
    //bagian bumper
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0f, 0f, 0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.035f,
            0.04f,
            36,
            18
        ));
        //bumper


        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.05f,
            0.04f,
            0.15f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.05f,
            0.04f,
            0.15f,
            36,
            18
        ));
        //pilar mobil
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.22f,
            0.11f,
            0.02f,
            36,
            18
        ));
        //kaca mobil
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.2f,
            0.1f,
            0.11f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.22f,
            0.11f,
            0.02f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.2f,
            0.1f,
            0.11f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.22f,
            0.11f,
            0.02f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new MirrorCube(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.2f,
            0.1f,
            0.1f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.22f,
            0.04f,
            0.15f,
            36,
            18
        ));
    //obj 4
        //penyambung
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.001f,
            0.02f,
            36,
            18
        ));
        //panjangan bemper obj 5
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.05f,
            0.02f,
            36,
            18
        ));
        //panjangan bumper obj6
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 1.0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.04f,
            0.02f,
            36,
            18
        ));
    //bagian bumper
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0f, 0f, 0f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.25f,
            0.035f,
            0.04f,
            36,
            18
        ));
        objectToni.get(0).getChildObject().add(new Cuboid(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(10f, 10f, 10f, 1.0f),
            Arrays.asList(0f, 0f, 0f),
            0.22f,
            0.01f,
            0.27f,
            36,
            18
        ));


        objectBan.add(new Torus(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.035f,
            0.035f,
            0.035f,
            360,
            18
        ));
        objectBan.add(new Torus(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.035f,
            0.035f,
            0.035f,
            360,
            18
        ));
        objectBan.add(new Torus(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.035f,
            0.035f,
            0.035f,
            360,
            18
        ));
        objectBan.add(new Torus(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.0f, 0.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.035f,
            0.035f,
            0.035f,
            360,
            18
        ));
        objectToni.get(0).getChildObject().add(new CylinderHorizontal(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.035f,
            0.035f,
            0.035f,
            360,
            18,0.0075f
        ));
        objectToni.get(0).getChildObject().add(new CylinderHorizontal(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.035f,
            0.035f,
            0.035f,
            360,
            18,0.0075f
        ));
        objectToni.get(0).getChildObject().add(new halfCircle(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.0175f,
            0.0175f,
            0.0175f,
            180,
            180
        ));
        objectToni.get(0).getChildObject().add(new halfCircle(
            Arrays.asList(
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
    GL_VERTEX_SHADER),
                new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
    GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(1.0f, 1.0f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f, 0.0f),
            0.0175f,
            0.0175f,
            0.0175f,
            180,
            180
        ));
        objectToni.get(0).translateObject(0.5f, -0.05f, 0f);
        objectToni.get(0).setCenterPoint(Arrays.asList(0.5f, -0.05f, 0f));


        objectToni.get(0).getChildObject().get(0).translateObject(0.0f, 0.127f, 0.125f);
        objectToni.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(0.0f, 0.127f, 0.125f));


        objectToni.get(0).getChildObject().get(1).translateObject(0.0f, 0.05475f, 0.275f);
        objectToni.get(0).getChildObject().get(1).setCenterPoint(Arrays.asList(0.0f, 0.05475f,
    0.275f));


        objectToni.get(0).getChildObject().get(2).translateObject(0.0f, 0.036f, 0.358f);
        objectToni.get(0).getChildObject().get(2).setCenterPoint(Arrays.asList(0.0f, 0.036f, 0.358f));


        objectToni.get(0).getChildObject().get(3).translateObject(0.0f, 0.01f, 0.358f);
        objectToni.get(0).getChildObject().get(3).setCenterPoint(Arrays.asList(0.0f, 0.01f, 0.358f));


        objectToni.get(0).getChildObject().get(4).translateObject(0.0f, -0.02f, 0.358f);
        objectToni.get(0).getChildObject().get(4).setCenterPoint(Arrays.asList(0.0f, -0.02f, 0.358f));


        objectToni.get(0).getChildObject().get(5).translateObject(0.0f, -0.059f, 0.362f);
        objectToni.get(0).getChildObject().get(5).setCenterPoint(Arrays.asList(0.0f, -0.059f,
    0.362f));
```

```
        objectToni.get(0).getChildObject().get(6).translateObject(0.1f, 0.05475f, 0.275f);
        objectToni.get(0).getChildObject().get(6).setCenterPoint(Arrays.asList(0.1f, 0.05475f,
0.275f));


        objectToni.get(0).getChildObject().get(7).translateObject(-0.1f, 0.05475f, 0.275f);
        objectToni.get(0).getChildObject().get(7).setCenterPoint(Arrays.asList(-0.1f, 0.05475f,
0.275f));


        objectToni.get(0).getChildObject().get(8).translateObject(0.0f, 0.125f, 0.077f);
        objectToni.get(0).getChildObject().get(8).setCenterPoint(Arrays.asList(0.0f, 0.125f, 0.077f));


        objectToni.get(0).getChildObject().get(9).translateObject(0.0f, 0.127f, 0.01f);
        objectToni.get(0).getChildObject().get(9).setCenterPoint(Arrays.asList(0.0f, 0.127f, 0.01f));


        objectToni.get(0).getChildObject().get(10).translateObject(0.0f, 0.125f, -0.055f);
        objectToni.get(0).getChildObject().get(10).setCenterPoint(Arrays.asList(0.0f, 0.125f,
-0.055f));


        objectToni.get(0).getChildObject().get(11).translateObject(0.0f, 0.127f, -0.12f);
        objectToni.get(0).getChildObject().get(11).setCenterPoint(Arrays.asList(0.0f, 0.127f,
-0.12f));


        objectToni.get(0).getChildObject().get(12).translateObject(0.0f, 0.125f, -0.185f);
        objectToni.get(0).getChildObject().get(12).setCenterPoint(Arrays.asList(0.0f, 0.125f,
-0.185f));


        objectToni.get(0).getChildObject().get(13).translateObject(0.0f, 0.128f, -0.245f);
        objectToni.get(0).getChildObject().get(13).setCenterPoint(Arrays.asList(0.0f, 0.128f,
-0.245f));


        objectToni.get(0).getChildObject().get(14).translateObject(0.0f, 0.0575f, -0.275f);
        objectToni.get(0).getChildObject().get(14).setCenterPoint(Arrays.asList(0.0f, 0.0575f,
-0.275f));
```

```
    objectToni.get(0).getChildObject().get(15).translateObject(0.0f, 0.036f, -0.358f);
    objectToni.get(0).getChildObject().get(15).setCenterPoint(Arrays.asList(0.0f, 0.036f,
-0.358f));


    objectToni.get(0).getChildObject().get(16).translateObject(0.0f, 0.01f, -0.358f);
    objectToni.get(0).getChildObject().get(16).setCenterPoint(Arrays.asList(0.0f, 0.01f,
-0.358f));


    objectToni.get(0).getChildObject().get(17).translateObject(0.0f, -0.02f, -0.358f);
    objectToni.get(0).getChildObject().get(17).setCenterPoint(Arrays.asList(0.0f, -0.02f,
-0.358f));


    objectToni.get(0).getChildObject().get(18).translateObject(0.0f, -0.059f, -0.362f);
    objectToni.get(0).getChildObject().get(18).setCenterPoint(Arrays.asList(0.0f, -0.059f,
-0.362f));


    objectToni.get(0).getChildObject().get(19).translateObject(0.0f, 0.185f, -0.05f);
    objectToni.get(0).getChildObject().get(19).setCenterPoint(Arrays.asList(0.0f, 0.185f,
-0.05f));


    objectToni.get(0).getChildObject().get(20).translateObject(-0.1f, 0f, 0.375f);
    objectToni.get(0).getChildObject().get(20).setCenterPoint(Arrays.asList(0.0f, 0.185f,
0.375f));


    objectToni.get(0).getChildObject().get(21).translateObject(0.1f, 0f, 0.375f);
    objectToni.get(0).getChildObject().get(21).setCenterPoint(Arrays.asList(0.0f, 0.185f,
0.375f));


    objectToni.get(0).getChildObject().get(22).translateObject(0.1f, 0.06f, 0.35f);
    objectToni.get(0).getChildObject().get(22).setCenterPoint(Arrays.asList(0.0f, 0.1f, 0.35f));


    objectToni.get(0).getChildObject().get(23).translateObject(-0.1f, 0.06f, 0.35f);
    objectToni.get(0).getChildObject().get(23).setCenterPoint(Arrays.asList(-0.1f, 0.1f, 0.35f));
    objectBan.get(0).translateObject(-0.275f, -0.09f, 0.4f);
    objectBan.get(0).setCenterPoint(Arrays.asList(-0.275f, -0.09f, 0.4f));
    objectBan.get(0).rotateObject((float) Math.toRadians(90f), 0.0f, 1.0f, 0.0f);


    objectBan.get(1).translateObject(-0.275f, -0.09f, 0.6f);
    objectBan.get(1).setCenterPoint(Arrays.asList(0.1f, -0.175f, -0.4f));
    objectBan.get(1).rotateObject((float) Math.toRadians(90f), 0.0f, 1.0f, 0.0f);


    objectBan.get(2).translateObject(0.275f, -0.09f, 0.4f);
    objectBan.get(2).setCenterPoint(Arrays.asList(-0.275f, -0.09f, 0.4f));
    objectBan.get(2).rotateObject((float) Math.toRadians(90f), 0.0f, 1.0f, 0.0f);


    objectBan.get(3).translateObject(0.275f, -0.09f, 0.6f);
    objectBan.get(3).setCenterPoint(Arrays.asList(0.1f, -0.175f, -0.4f));
    objectBan.get(3).rotateObject((float) Math.toRadians(90f), 0.0f, 1.0f, 0.0f);


    //kepala
    objectKepalaAlvin.add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.6f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f, 0.0f, 0.0f),
        0.125f,
        0.125f,
        0.125f,
        720,
        720
    ));


    objectKepalaAlvin.get(0).translateObject(-0.5f, 0.0f, 0.0f);
    objectKepalaAlvin.get(0).setCenterPoint(Arrays.asList(0.0f, 0.0f, 0.0f));
    objectKepalaAlvin.get(0).scaleObject(0.3f,0.3f,0.3f);
    //telinga kanan


    objectKepalaAlvin.get(0).getChildObject().add(new Cuboid(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.7f, 0.5f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.07f,
        0.07f,
        0.07f,
        36,
        18
    ));
    objectKepalaAlvin.get(0).getChildObject().get(0).translateObject(-0.435f, 0.07f, -0.005f);
    objectKepalaAlvin.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(0.065f, 0.07f,
-0.005f));
    objectKepalaAlvin.get(0).getChildObject().get(0).scaleObject(0.3f,0.3f,0.3f);




    //telinga kiri


    objectKepalaAlvin.get(0).getChildObject().add(new Cuboid(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.7f, 0.5f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.07f,
        0.07f,
        0.07f,
        36,
        18
    ));
    objectKepalaAlvin.get(0).getChildObject().get(1).translateObject(-0.57f, 0.07f, -0.005f);
    objectKepalaAlvin.get(0).getChildObject().get(1).setCenterPoint(Arrays.asList(-0.07f, 0.07f,
-0.005f));
    objectKepalaAlvin.get(0).getChildObject().get(1).scaleObject(0.3f,0.3f,0.3f);


    //pupil kanan


    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0f, 0f, 0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.00922f,
        0.00922f,
        0.00922f,
        720,
        720
    ));
    objectKepalaAlvin.get(0).getChildObject().get(2).translateObject(-0.455f, 0.05f, 0.12f);
    objectKepalaAlvin.get(0).getChildObject().get(2).setCenterPoint(Arrays.asList(0.045f, 0.05f,
0.12f));
    objectKepalaAlvin.get(0).getChildObject().get(2).scaleObject(0.3f,0.3f,0.3f);


    //pupil kiri


    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0f, 0f, 0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.00922f,
        0.00922f,
        0.00922f,
        720,
        720
    ));
    objectKepalaAlvin.get(0).getChildObject().get(3).translateObject(-0.545f, 0.05f, 0.12f);
    objectKepalaAlvin.get(0).getChildObject().get(3).setCenterPoint(Arrays.asList(-0.045f,
0.05f, 0.12f));
    objectKepalaAlvin.get(0).getChildObject().get(3).scaleObject(0.3f,0.3f,0.3f);
    //mata kanan
    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(1f, 1f, 1f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.0222f,
        0.0222f,
        0.0222f,
        720,
        720
    ));


    objectKepalaAlvin.get(0).getChildObject().get(4).translateObject(-0.45f, 0.05f, 0.1f);
    objectKepalaAlvin.get(0).getChildObject().get(4).setCenterPoint(Arrays.asList(0.05f, 0.05f,
0.1f));
    objectKepalaAlvin.get(0).getChildObject().get(4).scaleObject(0.3f,0.3f,0.3f);
    //mata biru kiri




    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.3f, 0.8f, 0.9f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.02f,
        0.02f,
        0.02f,
        720,
        720
    ));


    objectKepalaAlvin.get(0).getChildObject().get(5).translateObject(-0.551f, 0.05f, 0.105f);
    objectKepalaAlvin.get(0).getChildObject().get(5).setCenterPoint(Arrays.asList(-0.051f,
0.05f, 0.105f));
    objectKepalaAlvin.get(0).getChildObject().get(5).scaleObject(0.3f,0.3f,0.3f);


    //mata biru kanan




    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.3f, 0.8f, 0.9f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.02f,
        0.02f,
        0.02f,
        720,
        720
    ));


    objectKepalaAlvin.get(0).getChildObject().get(6).translateObject(  -0.449f, 0.05f, 0.105f);
    objectKepalaAlvin.get(0).getChildObject().get(6).setCenterPoint(Arrays.asList( 0.051f,
0.05f, 0.105f));
    objectKepalaAlvin.get(0).getChildObject().get(6).scaleObject(0.3f,0.3f,0.3f);
    //mata kiri
    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(1f, 1f, 1f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.0222f,
        0.0222f,
        0.0222f,
        720,
        720
    ));
    objectKepalaAlvin.get(0).getChildObject().get(7).translateObject(  -0.55f, 0.05f, 0.1f);
    objectKepalaAlvin.get(0).getChildObject().get(7).setCenterPoint(Arrays.asList(-0.05f, 0.05f,
0.1f));
    objectKepalaAlvin.get(0).getChildObject().get(7).scaleObject(0.3f,0.3f,0.3f);


    //dasar mulut
    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.7f, 0.5f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.052f,
        0.052f,
        0.052f,
        720,
        720
    ));
    objectKepalaAlvin.get(0).getChildObject().get(8).translateObject(  -0.5f,-0.05f, 0.1f);
    objectKepalaAlvin.get(0).getChildObject().get(8).setCenterPoint(Arrays.asList(0f,-0.05f,
0.1f));
    objectKepalaAlvin.get(0).getChildObject().get(8).scaleObject(0.3f,0.3f,0.3f);
    //hidung
    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0f, 0f, 0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.0122f,
        0.0122f,
        0.0122f,
        720,
        720
    ));
    objectKepalaAlvin.get(0).getChildObject().get(9).translateObject(-0.5f, 0.02f, 0.12f);
    objectKepalaAlvin.get(0).getChildObject().get(9).setCenterPoint(Arrays.asList(0.0f, 0.02f,
0.12f));
    objectKepalaAlvin.get(0).getChildObject().get(9).scaleObject(0.3f,0.3f,0.3f);




    //lidah


    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(1f, 0f, 0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.0222f,
        0.0042f,
        0.0422f,
            720,
            720
    ));


    objectKepalaAlvin.get(0).getChildObject().get(10).rotateObject(-0.05f,1f,0f,0f);
    objectKepalaAlvin.get(0).getChildObject().get(10).translateObject(  -0.5f, -0.05f, 0.15f);
    objectKepalaAlvin.get(0).getChildObject().get(10).setCenterPoint(Arrays.asList(0.0f, 0.01f,
0.15f));
    objectKepalaAlvin.get(0).getChildObject().get(10).scaleObject(0.3f,0.3f,0.3f);




    //badan


    objectKepalaAlvin.get(0).getChildObject().add(new Sphere(
            Arrays.asList(
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                    new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
            ),
            new ArrayList<>(),
            new Vector4f(0.6f, 0.45f, 0.0f, 1.0f),
            Arrays.asList(0.0f, 0.0f,0.0f),
            0.15f,
            0.15f,
            0.3f,
            720,
            720
    ));


    objectKepalaAlvin.get(0).getChildObject().get(11).translateObject(  -0.5f, -0.25f, -0.18f);
    objectKepalaAlvin.get(0).getChildObject().get(11).setCenterPoint(Arrays.asList(0.0f, -0.25f,
-0.18f));
    objectKepalaAlvin.get(0).getChildObject().get(11).scaleObject(0.3f,0.3f,0.3f);


    //leher
```

```
        objectKepalaAlvin.get(0).getChildObject().add(new Cylinder(
                Arrays.asList(
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
                ),
                new ArrayList<>(),
                new Vector4f(0.5f, 0.45f, 0.0f, 1.0f),
                Arrays.asList(0f,0f,0f),
                0.07f,
                0.07f,
                0.07f,
                7200,
                36,0.15f




        ));
        objectKepalaAlvin.get(0).getChildObject().get(12).translateObject(  -0.5f,-0.1f,-0f);
        objectKepalaAlvin.get(0).getChildObject().get(12).setCenterPoint(Arrays.asList(0f,-0.1f,-0f));
        objectKepalaAlvin.get(0).getChildObject().get(12).scaleObject(0.3f,0.3f,0.3f);




        //kaki 1


        objectKaki1Alvin.add(new Cylinder(
                Arrays.asList(
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
                        new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
                ),
                new ArrayList<>(),
                new Vector4f(0.5f, 0.45f, 0.0f, 1.0f),
                Arrays.asList(0.0f,0.0f,0.0f),
                0.035f,
                0.035f,
                0.035f,
                7200,
        36,0.15f




    ));


    objectKaki1Alvin.get(0).translateObject(-0.56f,-0.41f,-0f);
    objectKaki1Alvin.get(0).setCenterPoint(Arrays.asList(-0.06f,-0.41f,-0f));
    objectKaki1Alvin.get(0).scaleObject(0.3f,0.3f,0.3f);


    //kaki 2


    objectKaki2Alvin.add(new Cylinder(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.5f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.035f,
        7200,
        36,0.15f








    ));


    objectKaki2Alvin.get(0).translateObject(-0.44f,-0.41f,-0f);
    objectKaki2Alvin.get(0).setCenterPoint(Arrays.asList(0.06f,-0.41f,-0f));
    objectKaki2Alvin.get(0).scaleObject(0.3f,0.3f,0.3f);


    //kaki 4


    objectKaki4Alvin.add(new Cylinder(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.5f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.035f,
        7200,
        36,0.15f




    ));
    objectKaki4Alvin.get(0).translateObject(-0.56f,-0.41f,-0.35f);
    objectKaki4Alvin.get(0).setCenterPoint(Arrays.asList(-0.06f,-0.41f,-0.35f));
    objectKaki4Alvin.get(0).scaleObject(0.3f,0.3f,0.3f);
//
    //kaki 3


    objectKaki3Alvin.add(new Cylinder(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.5f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.035f,
        7200,
        36,0.15f




    ));
    objectKaki3Alvin.get(0).translateObject(-0.44f, -0.41f, -0.35f);
    objectKaki3Alvin.get(0).setCenterPoint(Arrays.asList(0.06f, -0.41f, -0.35f));
    objectKaki3Alvin.get(0).scaleObject(0.3f,0.3f,0.3f);


    //telapak kaki 1


    objectKaki4Alvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.6f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.045f,
        720,
        720
    ));


    objectKaki4Alvin.get(0).getChildObject().get(0).translateObject(-0.56f, -0.46f, -0.31f);
    objectKaki4Alvin.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.06f, -0.46f,
-0.31f));
    objectKaki4Alvin.get(0).getChildObject().get(0).scaleObject(0.3f,0.3f,0.3f);
//      objectKaki4Alvin.get(1).getChildObject().get(1).translateObject(0.25f,0.0f,0.0f);
//      objectKaki4Alvin.get(1).getChildObject().get(1).setCenterPoint(Arrays.asList(-0.06f,
-0.46f, -0.35f));
```

```
    //telapak kaki 2


    objectKaki3Alvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.6f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.045f,
        720,
        720
    ));


    objectKaki3Alvin.get(0).getChildObject().get(0).translateObject(-0.44f, -0.46f, -0.31f);
    objectKaki3Alvin.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(0.06f, -0.46f,
-0.31f));
    objectKaki3Alvin.get(0).getChildObject().get(0).scaleObject(0.3f,0.3f,0.3f);
//
    //telapak kaki 3
//
    objectKaki1Alvin.get(0).getChildObject().add(new  Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.6f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.045f,
        720,
        720
    ));
    objectKaki1Alvin.get(0).getChildObject().get(0).translateObject(-0.56f, -0.46f, 0.02f);
    objectKaki1Alvin.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(-0.06f, -0.46f,
0.02f));
    objectKaki1Alvin.get(0).getChildObject().get(0).scaleObject(0.3f,0.3f,0.3f);


    //telapak kaki 4


    objectKaki2Alvin.get(0).getChildObject().add(new Sphere(
        Arrays.asList(
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.vert",
GL_VERTEX_SHADER),
            new ShaderProgram.ShaderModuleData("resources/shaders/scene.frag",
GL_FRAGMENT_SHADER)
        ),
        new ArrayList<>(),
        new Vector4f(0.6f, 0.45f, 0.0f, 1.0f),
        Arrays.asList(0.0f,0.0f,0.0f),
        0.035f,
        0.035f,
        0.045f,
        720,
        720
    ));


    objectKaki2Alvin.get(0).getChildObject().get(0).translateObject(-0.44f, -0.46f, 0.02f);
    objectKaki2Alvin.get(0).getChildObject().get(0).setCenterPoint(Arrays.asList(0.06f, -0.46f,
0.02f));
    objectKaki2Alvin.get(0).getChildObject().get(0).scaleObject(0.3f,0.3f,0.3f);


  }




  public void input() {
    if (window.isKeyPressed(GLFW_KEY_W)) {
      countDegree++;
```

```
        // Get the center point of the first object


        //rotasi terhadap matahari
        //objectHans.get(0).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f)
        objectToni.get(0).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(1).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(2).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(3).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(4).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(5).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(6).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(7).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(8).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(9).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(10).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(11).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(12).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(13).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(14).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(15).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(16).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(17).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(18).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(19).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
//          objectToni.get(20).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);


        objectBan.get(0).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
        objectBan.get(1).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
        objectBan.get(2).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
        objectBan.get(3).rotateObject((float) Math.toRadians(1f),0.0f,1.0f,0.0f);
```

```
//            for(Object child:objects.get(0).getChildObject()){
//                List<Float> temp = new ArrayList<>(child.getCenterPoint());
//                //rotasi terhadap sumbu masing-masing planet
//                child.translateObject(temp.get(0)*-1,temp.get(1)*-1,temp.get(2)*-1);
//                child.rotateObject((float) Math.toRadians(0.5f),0.0f,1.0f,0.0f);
//                child.translateObject(temp.get(0)*1,temp.get(1)*1,temp.get(2)*1);
//                for(Object y:objects.get(0).getChildObject().get(1).getChildObject()){
//                    //rotasi terhadap bumi
//                    List<Float> temp1 = new
ArrayList<>(objects.get(0).getChildObject().get(1).getCenterPoint());
//                    y.translateObject(temp1.get(0)*-1,temp1.get(1)*-1,temp1.get(2)*-1);
//                    y.rotateObject((float) Math.toRadians(0.5f),0.0f,1.0f,0.0f);
//                    y.translateObject(temp1.get(0)*1,temp1.get(1)*1,temp1.get(2)*1);
//                    //rotasi terhadap sumbunya sendiri
//                    temp1 = new
ArrayList<>(objects.get(0).getChildObject().get(1).getChildObject().get(0).getCenterPoint());
//                    y.translateObject(temp1.get(0)*-1,temp1.get(1)*-1,temp1.get(2)*-1);
//                    y.rotateObject((float) Math.toRadians(0.5f),0.0f,1.0f,1.0f);
//                    y.translateObject(temp1.get(0)*1,temp1.get(1)*1,temp1.get(2)*1);
//                }
//                child.rotateObject((float) Math.toRadians(0.5f),0.0f,1.0f,1.0f);
//            }
       }
       if(window.isKeyPressed(GLFW_KEY_Z)){
           objectToni.get(0).getChildObject().get(9).scaleObject(0.22f,
                   0.09f,
                   0.02f);
           objectToni.get(0).getChildObject().get(11).scaleObject(0.22f,
                   0.09f,
                   0.02f);



       }
       if(window.isKeyPressed(GLFW_KEY_G)){
           objectReiner.get(0).translateObject(0.0f,0.0f,0.0025f);
           objectReiner.get(1).translateObject(0.0f,0.0f,0.0025f);
           objectKaki.get(0).translateObject(0.0f,0.0f,0.0025f);
       }
```

```
if(window.isKeyPressed(GLFW_KEY_B)){
   objectReiner.get(0).translateObject(0.0f,0.0f,-0.0025f);
   objectReiner.get(1).translateObject(0.0f,0.0f,-0.0025f);
   objectKaki.get(0).translateObject(0.0f,0.0f,-0.0025f);
}


if(window.isKeyPressed(GLFW_KEY_N)){
   objectReiner.get(0).translateObject(0.0025f,0.0f,0.0f);
   objectReiner.get(1).translateObject(0.0025f,0.0f,0.0f);
   objectKaki.get(0).translateObject(0.0025f,0.0f,0.0f);
}


if(window.isKeyPressed(GLFW_KEY_V)){
   objectReiner.get(0).translateObject(-0.0025f,0.0f,0.0f);
   objectReiner.get(1).translateObject(-0.0025f,0.0f,0.0f);
   objectKaki.get(0).translateObject(-0.0025f,0.0f,0.0f);
}


if(window.isKeyPressed(GLFW_KEY_UP)){
   objectHans.get(0).translateObject(0.0f,0.0f,0.0025f);
   objectHans.get(1).translateObject(0.0f,0.0f,0.0025f);
}
if(window.isKeyPressed(GLFW_KEY_DOWN)){
   objectHans.get(0).translateObject(0.0f,0.0f,-0.0025f);
   objectHans.get(1).translateObject(0.0f,0.0f,-0.0025f);
}


if(window.isKeyPressed(GLFW_KEY_RIGHT)){
   objectHans.get(0).translateObject(0.0025f,0.0f,0.0f);
   objectHans.get(1).translateObject(0.0025f,0.0f,0.0f);
}
if(window.isKeyPressed(GLFW_KEY_LEFT)){
   objectHans.get(0).translateObject(-0.0025f,0.0f,0.0f);
   objectHans.get(1).translateObject(-0.0025f,0.0f,0.0f);
}


if(window.isKeyPressed(GLFW_KEY_8)){
   objectToni.get(0).translateObject(0.0f,0.0f,0.0025f);
```

```
    objectBan.get(0).translateObject(0.0f,0.0f,0.0025f);
    objectBan.get(1).translateObject(0.0f,0.0f,0.0025f);
    objectBan.get(2).translateObject(0.0f,0.0f,0.0025f);
    objectBan.get(3).translateObject(0.0f,0.0f,0.0025f);


}
if(window.isKeyPressed(GLFW_KEY_2))


{
    objectToni.get(0).translateObject(0.0f,0.0f,-0.0025f);
    objectBan.get(0).translateObject(0.0f,0.0f,-0.0025f);
    objectBan.get(1).translateObject(0.0f,0.0f,-0.0025f);
    objectBan.get(2).translateObject(0.0f,0.0f,-0.0025f);
    objectBan.get(3).translateObject(0.0f,0.0f,-0.0025f);
}


if(window.isKeyPressed(GLFW_KEY_6)){
    objectToni.get(0).translateObject(0.0025f,0.0f,0.0f);


    objectBan.get(0).translateObject(0.0025f,0.0f,0.0f);
    objectBan.get(1).translateObject(0.0025f,0.0f,0.0f);
    objectBan.get(2).translateObject(0.0025f,0.0f,0.0f);
    objectBan.get(3).translateObject(0.0025f,0.0f,0.0f);
}


if(window.isKeyPressed(GLFW_KEY_4)){
    objectToni.get(0).translateObject(-0.0025f,0.0f,0.0f);


    objectBan.get(0).translateObject(-0.0025f,0.0f,0.0f);
    objectBan.get(1).translateObject(-0.0025f,0.0f,0.0f);
    objectBan.get(2).translateObject(-0.0025f,0.0f,0.0f);
    objectBan.get(3).translateObject(-0.0025f,0.0f,0.0f);
}
```

```
if (window.isKeyPressed(GLFW_KEY_T)) {


    objectToni.get(0).rotateObject((float) Math.toRadians(1f),0.0f,-1.0f,0.0f);



    objectBan.get(0).rotateObject((float) Math.toRadians(1f),0.0f,-1.0f,0.0f);
    objectBan.get(1).rotateObject((float) Math.toRadians(1f),0.0f,-1.0f,0.0f);
    objectBan.get(2).rotateObject((float) Math.toRadians(1f),0.0f,-1.0f,0.0f);
    objectBan.get(3).rotateObject((float) Math.toRadians(1f),0.0f,-1.0f,0.0f);



}
if (window.isKeyPressed(GLFW_KEY_F11)){
    List<Float> temp1 = new ArrayList<>(objectToni.get(0).getCenterPoint());
    objectToni.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectToni.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, 1.0f, 0.0f);
    objectToni.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);



    //List<Float> temp22 = new ArrayList<>(objectBan.get(0).getCenterPoint());
    objectBan.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectBan.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, 1.0f, 0.0f);
    objectBan.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);
    //List<Float> temp23 = new ArrayList<>(objectBan.get(1).getCenterPoint());
    objectBan.get(1).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectBan.get(1).rotateObject((float) Math.toRadians(5f), 0.0f, 1.0f, 0.0f);
    objectBan.get(1).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


    //List<Float> temp24 = new ArrayList<>(objectBan.get(2).getCenterPoint());
    objectBan.get(2).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectBan.get(2).rotateObject((float) Math.toRadians(5f), 0f, 1.0f, 0.0f);
    objectBan.get(2).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


    //List<Float> temp25 = new ArrayList<>(objectBan.get(3).getCenterPoint());
    objectBan.get(3).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectBan.get(3).rotateObject((float) Math.toRadians(5f), 0f, 1.0f, 0.0f);
    objectBan.get(3).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


}


if (window.isKeyPressed(GLFW_KEY_0)){


    List<Float> temp1 = new ArrayList<>(objectHans.get(0).getCenterPoint());
    objectHans.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectHans.get(0).rotateObject((float) Math.toRadians(2f), 0.0f, 1.0f, 0.0f);
    objectHans.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


    List<Float> temp2 = new ArrayList<>(objectHans.get(1).getCenterPoint());
    objectHans.get(1).translateObject(temp2.get(0) * -1, temp2.get(1) * -1, temp2.get(2) * -1);
    objectHans.get(1).rotateObject((float) Math.toRadians(2f), 0.0f, 1.0f, 0.0f);
    objectHans.get(1).translateObject(temp2.get(0) * 1, temp2.get(1) * 1, temp2.get(2) * 1);


}


if (window.isKeyPressed(GLFW_KEY_9)){
    List<Float> temp1 = new ArrayList<>(objectHans.get(0).getCenterPoint());
    objectHans.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) * -1);
    objectHans.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, -1.0f, 0.0f);
```

```
      objectHans.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


      List<Float> temp2 = new ArrayList<>(objectHans.get(1).getCenterPoint());
      objectHans.get(1).translateObject(temp2.get(0) * -1, temp2.get(1) * -1, temp2.get(2) * -1);
      objectHans.get(1).rotateObject((float) Math.toRadians(5f), 0.0f, -1.0f, 0.0f);
      objectHans.get(1).translateObject(temp2.get(0) * 1, temp2.get(1) * 1, temp2.get(2) * 1);
    }


    if (window.isKeyPressed(GLFW_KEY_COMMA)){
      List<Float> temp1 = new ArrayList<>(objectReiner.get(0).getCenterPoint());
      objectReiner.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) *
-1);
      objectReiner.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, 1.0f, 0.0f);
      objectReiner.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


      List<Float> temp2 = new ArrayList<>(objectReiner.get(1).getCenterPoint());
      objectReiner.get(1).translateObject(temp2.get(0) * -1, temp2.get(1) * -1, temp2.get(2) *
-1);
      objectReiner.get(1).rotateObject((float) Math.toRadians(5f), 0.0f, 1.0f, 0.0f);
      objectReiner.get(1).translateObject(temp2.get(0) * 1, temp2.get(1) * 1, temp2.get(2) * 1);


      List<Float> temp3 = new ArrayList<>(objectKaki.get(0).getCenterPoint());
      objectKaki.get(0).translateObject(temp2.get(0) * -1, temp2.get(1) * -1, temp2.get(2) * -1);
      objectKaki.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, 1.0f, 0.0f);
      objectKaki.get(0).translateObject(temp2.get(0) * 1, temp2.get(1) * 1, temp2.get(2) * 1);
    }


    if (window.isKeyPressed(GLFW_KEY_M)){
      List<Float> temp1 = new ArrayList<>(objectReiner.get(0).getCenterPoint());
      objectReiner.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2) *
-1);
      objectReiner.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, -1.0f, 0.0f);
      objectReiner.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) * 1);


      List<Float> temp2 = new ArrayList<>(objectReiner.get(1).getCenterPoint());
      objectReiner.get(1).translateObject(temp2.get(0) * -1, temp2.get(1) * -1, temp2.get(2) *
-1);
      objectReiner.get(1).rotateObject((float) Math.toRadians(5f), 0.0f, -1.0f, 0.0f);
```

```
            objectReiner.get(1).translateObject(temp2.get(0) * 1, temp2.get(1) * 1, temp2.get(2) * 1);


            List<Float> temp3 = new ArrayList<>(objectKaki.get(0).getCenterPoint());
            objectKaki.get(0).translateObject(temp2.get(0) * -1, temp2.get(1) * -1, temp2.get(2) * -1);
            objectKaki.get(0).rotateObject((float) Math.toRadians(5f), 0.0f, -1.0f, 0.0f);
            objectKaki.get(0).translateObject(temp2.get(0) * 1, temp2.get(1) * 1, temp2.get(2) * 1);
        }


        if (window.isKeyPressed(GLFW_KEY_I)){
//             List<Float> temp1 = new ArrayList<>(objectKaki1Alvin.get(0).getCenterPoint());
//             objectKaki1Alvin.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1,
temp1.get(2) * -1);
//             objectKaki1Alvin.get(0).rotateObject((float) Math.toRadians(1f), 0.0f, 0.0f, 1.0f);
//             objectKaki1Alvin.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2)
* 1);


            objectKepalaAlvin.get(0).translateObject(0.0f,0.0f,0.01f);
            objectKaki1Alvin.get(0).translateObject(0.0f,0.0f,0.01f);
            objectKaki2Alvin.get(0).translateObject(0.0f,0.0f,0.01f);
            objectKaki3Alvin.get(0).translateObject(0.0f,0.0f,0.01f);
            objectKaki4Alvin.get(0).translateObject(0.0f,0.0f,0.01f);
        }


        if(window.isKeyPressed(GLFW_KEY_K)){
            objectKepalaAlvin.get(0).translateObject(0.0f,0.0f,-0.01f);
            objectKaki1Alvin.get(0).translateObject(0.0f,0.0f,-0.01f);
            objectKaki2Alvin.get(0).translateObject(0.0f,0.0f,-0.01f);
            objectKaki3Alvin.get(0).translateObject(0.0f,0.0f,-0.01f);
            objectKaki4Alvin.get(0).translateObject(0.0f,0.0f,-0.01f);
        }


        if(window.isKeyPressed(GLFW_KEY_L)){
            objectKepalaAlvin.get(0).translateObject(0.01f,0.0f,0.0f);
            objectKaki1Alvin.get(0).translateObject(0.01f,0.0f,0.0f);
            objectKaki2Alvin.get(0).translateObject(0.01f,0.0f,0.0f);
            objectKaki3Alvin.get(0).translateObject(0.01f,0.0f,0.0f);
            objectKaki4Alvin.get(0).translateObject(0.01f,0.0f,0.0f);
        }
        if(window.isKeyPressed(GLFW_KEY_J)){
            objectKepalaAlvin.get(0).translateObject(-0.01f,0.0f,0.0f);
            objectKaki1Alvin.get(0).translateObject(-0.01f,0.0f,0.0f);
            objectKaki2Alvin.get(0).translateObject(-0.01f,0.0f,0.0f);
            objectKaki3Alvin.get(0).translateObject(-0.01f,0.0f,0.0f);
            objectKaki4Alvin.get(0).translateObject(-0.01f,0.0f,0.0f);
        }


        if (window.isKeyPressed(GLFW_KEY_X)) {




//            for (int i = 0; i < objectKepalaAlvin.size(); i++){
//
//                objectKepalaAlvin.get(i).rotateObject(0.1f, 0.0f,1.0f,0.0f);
//
//            }


            List<Float> temp1 = new ArrayList<>(objectKepalaAlvin.get(0).getCenterPoint());
            objectKepalaAlvin.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1,
temp1.get(2) * -1);
            objectKepalaAlvin.get(0).rotateObject((float) Math.toRadians(1f), 0.0f, 1.0f, 0.0f);
            objectKepalaAlvin.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2)
* 1);





            objectKaki1Alvin.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2)
* -1);
            objectKaki1Alvin.get(0).rotateObject((float) Math.toRadians(1f), 0.0f, 1.0f, 0.0f);
            objectKaki1Alvin.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) *
1);


            objectKaki2Alvin.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2)
* -1);
            objectKaki2Alvin.get(0).rotateObject((float) Math.toRadians(1f), 0.0f, 1.0f, 0.0f);
            objectKaki2Alvin.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) *
1);
```

```
        objectKaki3Alvin.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2)
* -1);
        objectKaki3Alvin.get(0).rotateObject((float) Math.toRadians(1f), 0.0f, 1.0f, 0.0f);
        objectKaki3Alvin.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) *
1);


        objectKaki4Alvin.get(0).translateObject(temp1.get(0) * -1, temp1.get(1) * -1, temp1.get(2)
* -1);
        objectKaki4Alvin.get(0).rotateObject((float) Math.toRadians(1f), 0.0f, 1.0f, 0.0f);
        objectKaki4Alvin.get(0).translateObject(temp1.get(0) * 1, temp1.get(1) * 1, temp1.get(2) *
1);
    }
//          // Get the center point of the first object
//          List<Float> centerPoint = objects.get(0).getCenterPoint();
//
//          // Translate the object by a fixed amount
//          float dx = 0.001F;
//          float dy = 0.001F;
//
//          // Translate the object by the offset
//          objects.get(0).translateObject(dx, dy, 0F);
//
//          // You can also apply the translation to the child objects if necessary
//          // for(Object child : objects.get(0).getChildObject()){
//          //     child.translateObject(dx, dy, 0);
//          // }
//      }
//      if (window.isKeyPressed(GLFW_KEY_R)) {
//          // Get the first object in the list
//          Object objectToRotate = objects.get(0);
//
//          float angle = 0.1F;  // in radians
//          float axisX = 0.0F;
//          float axisY = 0.0F;
//          float axisZ = 1.0F;
//
//
//          // Apply the rotation to the object
//          objectToRotate.rotateObject(angle, axisX, axisY, axisZ);
//
//          // You can also apply the rotation to the child objects if necessary
//            // for(Object child : objectToRotate.getChildObject()){
//            //     child.rotateObject(angle, axisX, axisY, axisZ);
//            // }
//        }
//        if (window.isKeyPressed(GLFW_KEY_S)) {
//            // Get the first object in the list
//            Object objectToRotate = objects.get(0);
//
//            // Define the rotation angle and axis
//            float angle = 0.05F;  // in radians
//            float axisX = 0.0F;
//            float axisY = 0.0F;
//            float axisZ = 1.0F;
//
//            // Apply the rotation to the object
//            objectToRotate.rotateObject(angle, axisX, axisY, axisZ);
//
//            // You can also apply the rotation to the child objects if necessary
//            // for(Object child : objectToRotate.getChildObject()){
//            //     child.rotateObject(angle, axisX, axisY, axisZ);
//            // }
//        }
```

```
//
//        if(mouseInput.isLeftButtonPressed()){
//            Vector2f pos = mouseInput.getCurrentPos();
////            System.out.println("x : "+pos.x+" y : "+pos.y);
//            pos.x = (pos.x - (window.getWidth())/2.0f) /
//                    (window.getWidth()/2.0f);
//            pos.y = (pos.y - (window.getHeight())/2.0f) /
//                    (-window.getHeight()/2.0f);
//            //System.out.println("x : "+pos.x+" y : "+pos.y);
//
//            if((!(pos.x > 1 || pos.x < -0.97) && !(pos.y > 0.97 || pos.y < -1))){
//                System.out.println("x : "+pos.x+" y : "+pos.y);
////                objectsPointsControl.get(0).addVertices(new Vector3f(pos.x,pos.y,0));
//
//            }
//        }
```

```
    }
    public void loop(){
        while (window.isOpen()) {
            window.update();
            glClearColor(0.02f,
                    0.15f, 1.0f,
                    0.0f);
            GL.createCapabilities();
            input();


            //code
            for(Object object: objects){
                object.draw(camera,projection);
            }
            for(Object object: objectsRectangle){
                object.draw(camera,projection);
            }
//            for(Object object: objectsPointsControl){
//                object.drawLine();
//            }
            for(Object object: environment){
                object.draw(camera,projection);
            }


            for(Object object: objectHans){
                object.draw(camera,projection);
            }




            for(Object object: objectKakiHans){
                object.draw(camera,projection);
            }
            for(Object object: objectToni){
                object.draw(camera,projection);
            }
            for(Object object: objectBan){
                object.draw(camera,projection);
            }
```

```
for (Object object : objectKepalaAlvin) {
    object.draw(camera, projection);
}


for (Object object : objectKaki3Alvin) {
    object.draw(camera, projection);
}


for (Object object : objectKaki4Alvin) {
    object.draw(camera, projection);
}


for (Object object : objectKaki1Alvin) {
    object.draw(camera, projection);
}


for (Object object : objectKaki2Alvin) {
    object.draw(camera, projection);
}


for(Object object: objectPohon){
    object.draw(camera,projection);
}


for(Object object: objectReiner){
    object.draw(camera,projection);
}


for(Object object: objectJendela){
    object.draw(camera,projection);
}


for(Object object: objectAtap){
    object.draw(camera,projection);
```

```
        }


        for(Object object: objectKaki){
            object.draw(camera,projection);
        }






        // Restore state
        glDisableVertexAttribArray(0);


        // Poll for window events.
        // The key callback above will only be
        // invoked during this call.
        glfwPollEvents();





    }
}
public Vector3f calculateBezierPoint(float t) {
    List<Vector3f> points = new ArrayList<>(controlPoints);


    while (points.size() > 1) {
        for (int j = 0; j < points.size() - 1; j++) {
            points.set(j, lerp(points.get(j), points.get(j + 1), t));
        }
        points.remove(points.size() - 1);
    }


    return points.get(0);
}
private Vector3f lerp(Vector3f a, Vector3f b, float t) {
    float x = a.x + t * (b.x - a.x);
    float y = a.y + t * (b.y - a.y);
    float z = a.z + t * (b.z - a.z);
```

```
        return new Vector3f(x, y, z);
    }




    public void run() {


        init();
        loop();


        // Terminate GLFW and
        // free the error callback
        glfwTerminate();
        glfwSetErrorCallback(null).free();
    }
    public static void main(String[] args) {
        new Main().run();
        Main curve = new Main(
                new Vector3f(1, 1, 2),
                new Vector3f(1, 2, 3),
                new Vector3f(4, 5, 6),
                new Vector3f(7, 8, 9)
        );


        for (float t = 0; t <= 1; t += 0.1f) {
            Vector3f point = curve.calculateBezierPoint(t);
            System.out.println(point);
        }
    }


    public void stopTranslation(){
        shouldTranslate = false;
    }
}



private void createTorus(List<Float> centerPoint){
        vertices.clear();
        for(float i = 0;i<360;i+=(360f/sectorCount)){
            double radI = degToRad(i);
            for(float j = 0; j < 360; j+=(360f/stackCount)){
                double radJ = degToRad(j);
                Float x = (float) ((centerPoint.get(0) + (radiusX +
radiusY*Math.cos(radJ))*Math.cos(radI)));
                Float y = (float) ((centerPoint.get(1) + (radiusX +
radiusY*Math.cos(radJ))*Math.sin(radI)));
                Float z = (float) (centerPoint.get(2) + radiusZ*Math.sin(radJ));
                vertices.add(new Vector3f(x,y,z));
            }
        }
    }


 public void halfCircle(){
        float pi = (float)Math.PI;

        float sectorStep = 2 * (float)Math.PI / sectorCount;
        float stackStep = (float)Math.PI / stackCount;
        float sectorAngle, StackAngle, x, y, z;

        for (int i = 0; i <= stackCount/2; ++i)
        {
            StackAngle = pi / 2 - i * stackStep;
            x = radiusX * (float)Math.cos(StackAngle);
            y = radiusY * (float)Math.cos(StackAngle);
            z = radiusZ * (float)Math.sin(StackAngle);

            for (int j = 0; j <= sectorCount; ++j)
            {
                sectorAngle = j * sectorStep;
                Vector3f temp_vector = new Vector3f();
                temp_vector.x = centerPoint.get(0) + x * (float)Math.cos(sectorAngle);
                temp_vector.y = centerPoint.get(1) + y * (float)Math.sin(sectorAngle);
                temp_vector.z = centerPoint.get(2) + z;
                vertices.add(temp_vector);
            }
        }
    }

public void createSphere(){
        float pi = (float)Math.PI;
        float sectorStep = 2 * (float)Math.PI / sectorCount;
        float stackStep = (float)Math.PI / stackCount;
        float sectorAngle, StackAngle, x, y, z;

        for (int i = 0; i <= stackCount; ++i)
        {
            StackAngle = pi / 2 - i * stackStep;
            x = radiusX * (float)Math.cos(StackAngle);
            y = radiusY * (float)Math.cos(StackAngle);
            z = radiusZ * (float)Math.sin(StackAngle);

            for (int j = 0; j <= sectorCount; ++j)
            {
                sectorAngle = j * sectorStep;
                Vector3f temp_vector = new Vector3f();
                temp_vector.x = centerPoint.get(0) + x * (float)Math.cos(sectorAngle);
                temp_vector.y = centerPoint.get(1) + y * (float)Math.sin(sectorAngle);
                temp_vector.z = centerPoint.get(2) + z;
                vertices.add(temp_vector);
            }
        }
    }

public Cuboidmodiv(List<ShaderModuleData> shaderModuleDataList, List<Vector3f> vertices,
Vector4f color, List<Float> centerPoint, Float radiusX, Float radiusY, float radiusZ, int
stackCount, int sectorCount) {
        super(shaderModuleDataList, vertices, color, centerPoint, radiusX, radiusY);
        this.radiusZ = radiusZ;
        this.stackCount = stackCount;
        this.sectorCount = sectorCount;
        createBox();
        setupVAOVBO();
    }
    public void createBox(){
        Vector3f temp = new Vector3f();
        ArrayList<Vector3f> tempVertices = new ArrayList<>();
        //TITIK 1
        temp.x = centerPoint.get(0) - radiusX / 2.0f;
        temp.y = centerPoint.get(1) + radiusY / 2.0f;
        temp.z = centerPoint.get(2) - radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 2
        temp.x = centerPoint.get(0) + radiusX / 2.0f;
        temp.y = centerPoint.get(1) + radiusY / 2.0f;
        temp.z = centerPoint.get(2) - radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 3
        temp.x = centerPoint.get(0) + radiusX / 2.0f;
        temp.y = centerPoint.get(1) - radiusY / 2.0f;
        temp.z = centerPoint.get(2) - radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 4
        temp.x = centerPoint.get(0) - radiusX / 2.0f;
        temp.y = centerPoint.get(1) - radiusY / 2.0f;
        temp.z = centerPoint.get(2) - radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 5
        temp.x = centerPoint.get(0) - radiusX / 2.0f;
        temp.y = centerPoint.get(1) + radiusY / 2.0f;
        temp.z = centerPoint.get(2) + radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 6
        temp.x = centerPoint.get(0) + radiusX / 2.0f;
        temp.y = centerPoint.get(1) + radiusY / 2.0f;
        temp.z = centerPoint.get(2) + radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 7
        temp.x = centerPoint.get(0) + radiusX / 2.0f;
        temp.y = centerPoint.get(1) - radiusY / 2.0f;
        temp.z = centerPoint.get(2) + radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();
        //TITIK 8
        temp.x = centerPoint.get(0) - radiusX / 2.0f;
        temp.y = centerPoint.get(1) - radiusY / 2.0f;
        temp.z = centerPoint.get(2) + radiusZ / 2.0f;
        tempVertices.add(temp);
        temp = new Vector3f();

        vertices.clear();
        //kotak yg sisi belakang
        vertices.add(tempVertices.get(0));
```

```
        vertices.add(tempVertices.get(1));
        vertices.add(tempVertices.get(2));
        vertices.add(tempVertices.get(3));
        //kotak yg sisi depan
        //vertices.add(tempVertices.get(4));
        //vertices.add(tempVertices.get(5));
        vertices.add(tempVertices.get(6));
        vertices.add(tempVertices.get(7));
        //kotak yg sisi kiri
        //vertices.add(tempVertices.get(0));
        //vertices.add(tempVertices.get(4));
        //vertices.add(tempVertices.get(7));
        //vertices.add(tempVertices.get(3));
        //kotak yg sisi kanan
        vertices.add(tempVertices.get(1));
        //vertices.add(tempVertices.get(5));
        vertices.add(tempVertices.get(6));
        vertices.add(tempVertices.get(2));
        //kotak yg sisi atas
        //vertices.add(tempVertices.get(0));
        //vertices.add(tempVertices.get(1));
        //vertices.add(tempVertices.get(5));
        //vertices.add(tempVertices.get(4));
        //kotak yg sisi bawah
        vertices.add(tempVertices.get(3));
        //vertices.add(tempVertices.get(2));
        vertices.add(tempVertices.get(7));
        //vertices.add(tempVertices.get(6));
    }

public Cylinder(List<ShaderModuleData> shaderModuleDataList, List<Vector3f> vertices,
Vector4f color, List<Float> centerPoint, Float radiusX, Float radiusY, float radiusZ, int
stackCount, int sectorCount, float height) {
        super(shaderModuleDataList, vertices, color, centerPoint, radiusX, radiusY);
        this.radiusZ = radiusZ;
        this.stackCount = stackCount;
        this.sectorCount = sectorCount;
        this.height = height;

        createCylinder(height);
        setupVAOVBO();
    }

    public void createCylinder(float height) {
        float pi = (float) Math.PI;

        float sectorStep = 2 * pi / sectorCount;
        float stackStep = height / stackCount;
        float sectorAngle, x, y, z;

//        for (int i = 0; i <= stackCount; ++i) {
//            z = i * stackStep - height / 2;
//            for (int j = 0; j <= sectorCount; ++j) {
//                sectorAngle = j * sectorStep;
//                Vector3f temp_vector = new Vector3f();
//                temp_vector.x = centerPoint.get(0) + radiusX * (float) Math.cos(sectorAngle);
//                temp_vector.y = centerPoint.get(1) + radiusY * (float) Math.sin(sectorAngle);
//                temp_vector.z = centerPoint.get(2) + z;
//                vertices.add(temp_vector);
//            }
//        }


        for (int i = 0; i <= stackCount; ++i) {
            y = i * stackStep - height / 2;
            for (int j = 0; j <= sectorCount; ++j) {
                sectorAngle = j * sectorStep;
                Vector3f temp_vector = new Vector3f();
                temp_vector.x = centerPoint.get(0) + radiusX * (float) Math.cos(sectorAngle);
                temp_vector.y = centerPoint.get(1) + y;
                temp_vector.z = centerPoint.get(2) + radiusZ * (float) Math.sin(sectorAngle);
                vertices.add(temp_vector);
            }
        }
    }
```