

# Laporan Proyek Akhir Grafkom

## Kelompok 13

Kelompok & Pembagian tugas:

- **Patricia Jennifer Leoni Soetikno (C14210092)** - Model, Animasi, Editor, Direktor 2
- **Clarissa Soetikno (C14210134)** - Skybox, Texture, Collision
- **Nicholas Soetikno (C14210110)** - Lighting, Direktor, Performance (Back Face Culling), Editor 2

Video:

<https://youtu.be/owwFB8hlq24>

GitHub:

<https://github.com/JenniferLeoni/proyekGrafkom>

Story:

Karakter utama akan bangun di suatu ruangan yang tidak familiar. Karakter utama pun bingung dengan apa yang terjadi dan mulai mengeksplorasi ruangan. Lama kelamaan user menyadari bahwa keadaan ini sangatlah tidak normal. Ia pun berhadapan dengan hantu dan harus lari untuk menyelamatkan nyawanya. Karakter utama berhasil keluar dari ruangan gelap itu dan terbangun dari mimpinya hanya untuk mengetahui bahwa ia bangun di ruangan yang baru saja ia eksplor. Apakah ia masih terjebak dalam mimpi?

*Is this the real life?  
Is this just fantasy?  
Caught in a landslide,  
No escape from reality  
- Bohemian Rhapsody -*

### Pengaturan Collision:

Collision yang dihardcode secara sederhana terdapat pada camera untuk membatasinya keluar dari model ruangan yang ada.

Berikut cuplikan dari kode untuk pengaturan collision yang dimaksud

```
}  
if (window.isKeyPressed(GLFW_KEY_UP)) {  
    if (camera.getPosition().y <= 15.0f) {  
        camera.moveUp(move);  
    }  
} else if (window.isKeyPressed(GLFW_KEY_DOWN)) {  
    if (camera.getPosition().y >= 8.0f) {  
        camera.moveDown(move);  
    }  
}  
}
```

### Pengaturan lighting:

Rata rata pengaturan lighting di berbagai tempat redup untuk membuat suasana horor dan tua serta penaruhan lighting seminim mungkin agar banyak ruangan yang memiliki lighting minim agar sesuai dengan tema story yaitu horor. Beberapa tempat diberi lighting yang sangat terang dengan tujuan untuk memindah fokus penonton/user agar terarah ke objek/area tersebut.

Berikut cuplikan dari kode lighting yang digunakan untuk menambah lighting pointlight

```
//add langsung
//Room 1
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: -1.5f, y: 20.0f, z: -1.1f),
    intensity: 5.0f);
//Chair+Doll Room 1
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 0.0f, z: 0.0f), new Vector3f( x: 0.0f, y: 7.0f, z: -1.0f),
    intensity: 20.0f);
//DoorFrame Room 1
sceneLights.addPointLights(new Vector3f(1.0f,0.0f,0.0f), new Vector3f(14.0f, 25.0f, -25.0f), 100.0f);

//Hallway 1
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: 14.0f, y: 16.0f, z: -60.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: 14.0f, y: 16.0f, z: -100.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: 14.0f, y: 16.0f, z: -140.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: 14.0f, y: 16.0f, z: -180.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: 14.0f, y: 16.0f, z: -220.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: 14.0f, y: 16.0f, z: -260.0f),
    intensity: 10.0f);

//Hallway 2
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: -32.0f, y: 16.0f, z: -180.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: -72.0f, y: 16.0f, z: -180.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: -112.0f, y: 16.0f, z: -180.0f),
    intensity: 10.0f);
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 1.0f, z: 1.0f), new Vector3f( x: -152.5f, y: 16.0f, z: -180.0f),
    intensity: 10.0f);

//Doll Room Door frame
sceneLights.addPointLights(new Vector3f( x: 1.0f, y: 0.0f, z: 0.0f), new Vector3f( x: -303.0f, y: 16.0f, z: -180.0f),
    intensity: 100.0f);
```

Berikut cuplikan dari kode lighting yang digunakan untuk menambah lighting spotlight

```
//Spotlight 1 (End of Hallway 1)
PointLight temp = new PointLight(new Vector3f( x: 1.0f, y: 0.0f, z: 0.0f), new Vector3f( x: 12.5f, y: 0.0f, z: -338.0f),
    intensity: 250.0f);
sceneLights.addSpotLights(temp, new Vector3f( x: 0.0f, y: 1.0f, z: 0.0f), cutOffAngle: 0.0f);
```

Sedangkan dir light dan dan ambient light hanya terdapat 1 pada “world” tersebut, dimana ambience nya di set warna coklat agar memberi suasana “tua”

### Pengaturan Kamera:

Kamera bisa diatur menjadi first person view dimana user bisa mengeksplor ruangan ruangan yang ada.

Untuk Cinematography, kamera diatur tiap detiknya sesuai scene.

- Pada saat jalan, kamera diatur untuk maju dan naik turun menyerupai motion manusia ketika berjalan.

- Pada saat menolehkan kepala, kamera diatur untuk berputar
- Kedua nya bisa dilakukan disaat yang bersamaan.

```

if (window.isKeyPressed(GLFW_KEY_SPACE) || animasiStart){
    if (countAnimasi == 0){
        camera.setPosition( (x) -17.5f, camera.getPosition().y, (x) -20.25f);
        camera.setRotation((float) Math.toRadians(45), (float) Math.toRadians(180));
    }

    countAnimasi++;
    System.out.println(countAnimasi);

    if (countAnimasi < 40 && countLoop == 0){
        camera.addRotation( (x) -0.023f, (y) -0.023f);
    }
    if (countAnimasi > 55 && countAnimasi < 90){
        camera.addRotation( (x) 0.0f, (y) -0.023f);
    }
    if (countAnimasi > 100 && countAnimasi < 180){
        camera.addRotation( (x) 0.0f, (y) 0.023f);
    }
    if (countAnimasi > 180 && countAnimasi < 240){
        camera.addRotation( (x) 0.023f, (y) 0.023f);
    }
    if (countAnimasi > 300 && countAnimasi < 360){
        camera.addRotation( (x) -0.021f, (y) -0.023f);
    }
    if (countAnimasi > 370 && countAnimasi < 450 && countLoop == 0){
        camera.moveForward(jalan);
        headBopsJalan();
    }
    if (countAnimasi > 450 && countAnimasi < 490 && countLoop == 0){
        camera.addRotation( (x) 0.023f, (y) 0.0f);
    }
    if (countAnimasi > 500 && countAnimasi < 540 && countLoop == 0){
        camera.addRotation( (x) -0.023f, (y) 0.021f);
    }
    if (countAnimasi > 540 && countAnimasi < 660 && countLoop == 0){
        camera.addRotation( (x) 0.0f, (y) -0.023f);
    }
    if (countAnimasi > 660 && countAnimasi < 730 && countLoop == 0){
        camera.moveForward(jalan);
        headBopsJalan();
    }
    if (countAnimasi > 730 && countAnimasi < 770 && countLoop == 0){
        camera.addRotation( (x) 0.0f, (y) 0.02f);
    }
    if (countAnimasi > 780 && countAnimasi < 900 && countLoop == 0){
        camera.addRotation( (x) 0.0f, (y) -0.02f);
    }
    if (countAnimasi > 910 && countAnimasi < 960 && countLoop == 0){
        camera.moveForward(jalan);
        headBopsJalan();
    }
}

```

contoh kode untuk pengaturan cinematography

## Skybox:

Skybox dibuat dengan inisialisasi model dan entity skybox berbentuk cube (diimpor dalam bentuk .obj) yang nantinya akan bergerak mengikuti kamera sebagai center point untuk memberikan kesan langit dan terrain yang tidak terbatas. Skybox sendiri dibuat dengan komponen class / file yang terdiri dari SkyBox.java, SkyBoxRender.java, skybox.frag, skybox.vert, dan lain sebagainya.

Berikut cuplikan dari kode untuk pengaturan skybox dan terrainnya

```
public void init(Window window, Scene scene, Render render) {
    // Terrain
    String quadModelId = "quad-model";
    Model quadModel = ModelLoader.loadModel(modelId: "quad-model", modelPath: "resources/models/quad/quad.obj",
        scene.getTextureCache());
    scene.addModel(quadModel);

    int numRows = NUM_CHUNKS * 2 + 1;
    int numCols = numRows;
    terrainEntities = new Entity[numRows][numCols];
    for (int j = 0; j < numRows; j++) {
        for (int i = 0; i < numCols; i++) {
            Entity entity = new Entity(id: "TERRAIN_" + j + "_" + i, quadModelId);
            terrainEntities[j][i] = entity;
            scene.addEntity(entity);
        }
    }
}
```

```
SkyBox skyBox = new SkyBox(skyBoxModelPath: "resources/models/skybox/skybox.obj", scene.getTextureCache());
skyBox.getSkyBoxEntity().setScale(50);
scene.setSkyBox(skyBox);
```

2 usages

```
public void updateTerrain(Scene scene) {
    int cellSize = 10;
    Camera camera = scene.getCamera();
    Vector3f cameraPos = camera.getPosition();
    int cellCol = (int) (cameraPos.x / cellSize);
    int cellRow = (int) (cameraPos.z / cellSize);

    int numRows = NUM_CHUNKS * 2 + 1;
    int numCols = numRows;
    int zOffset = -NUM_CHUNKS;
    float scale = cellSize / 2.0f;
    for (int j = 0; j < numRows; j++) {
        int xOffset = -NUM_CHUNKS;
        for (int i = 0; i < numCols; i++) {
            Entity entity = terrainEntities[j][i];
            entity.setScale(scale);
            entity.setPosition(x: (cellCol + xOffset) * 2.0f, y: -0.01f, z: (cellRow + zOffset) * 2.0f);
            entity.getModelMatrix().identity().scale(scale).translate(entity.getPosition());
            xOffset++;
        }
        zOffset++;
    }
}
```

## Texture / Material:

Texture disample pada model dengan cara menyimpan koordinat texture yang nantinya akan dihubungkan dengan vertex model. Texture sendiri dibuat dengan komponen class / file yang terdiri dari Texture.java, TextureCache.java, Material.java, dan lain sebagainya

Berikut cuplikan dari kode untuk pengaturan texture / material

```

public class Texture {

    4 usages
    private int textureId;

    3 usages
    private String texturePath;

    1 usage
    public Texture(int width, int height, ByteBuffer buf) {
        this.texturePath = "";
        generateTexture(width, height, buf);
    }

    2 usages
    public Texture(String texturePath) {
        try (MemoryStack stack = MemoryStack.stackPush()) {
            this.texturePath = texturePath;
            IntBuffer w = stack.mallocInt( size: 1);
            IntBuffer h = stack.mallocInt( size: 1);
            IntBuffer channels = stack.mallocInt( size: 1);

            ByteBuffer buf = stbi_load(texturePath, w, h, channels, desired_channels: 4);
            if (buf == null) {
                throw new RuntimeException("Image file [" + texturePath + "] not loaded: " + stbi_failure_reason());
            }

            int width = w.get();
            int height = h.get();

            generateTexture(width, height, buf);

            stbi_image_free(buf);
        }
    }
}

```

## Performance:

Pada proyek kita untuk mengurangi beban pada hardware agar dapat meningkatkan performa kami menggunakan method “Back Face Culling” untuk rendering dimana objek objek yang akan di render hanyalah objek yang ada di dalam sudut pandang kamera/karakter utama.

Berikut cuplikan code untuk render dengan menggunakan “Back Face Culling” method:



```
public void render(Scene scene) {
    IGuiInstance guiInstance = null;
    if (guiInstance == null) {
        return;
    }
    guiInstance.drawGui();

    shaderProgram.bind();

    glEnable(GL_BLEND);
    glBlendEquation(GL_FUNC_ADD);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    glDisable(GL_DEPTH_TEST);
    glDisable(GL_CULL_FACE);

    glBindVertexArray(guiMesh.getVaoId());

    glBindBuffer(GL_ARRAY_BUFFER, guiMesh.getVerticesVB0());
    glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, guiMesh.getIndicesVB0());

    ImGuiIO io = ImGui.getIO();
    scale.x = 2.0f / io.getDisplaySizeX();
    scale.y = -2.0f / io.getDisplaySizeY();
    uniformsMap.setUniform( uniformName: "scale", scale);
}
```

```

ImDrawData drawData = ImGui.getDrawData();
int numLists = drawData.getCmdListsCount();
for (int i = 0; i < numLists; i++) {
    glBindBuffer(GL_ARRAY_BUFFER, drawData.getCmdListVtxBufferData(i), GL_STREAM_DRAW);
    glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, drawData.getCmdListIdxBufferData(i), GL_STREAM_DRAW);

    int numCmds = drawData.getCmdListCmdBufferSize(i);
    for (int j = 0; j < numCmds; j++) {
        final int elemCount = drawData.getCmdListCmdBufferElemCount(i, j);
        final int idxBufferOffset = drawData.getCmdListCmdBufferIdxOffset(i, j);
        final int indices = idxBufferOffset * ImDrawData.SIZEOF_IM_DRAW_IDX;

        texture.bind();
        glDrawElements(GL_TRIANGLES, elemCount, GL_UNSIGNED_SHORT, indices);
    }
}

glEnable(GL_DEPTH_TEST);
glEnable(GL_CULL_FACE);
glDisable(GL_BLEND);
}

```

Link GitHub: <https://github.com/JenniferLeoni/proyekGrafkom>

Link Youtube: <https://youtu.be/owwFB8hlq24>