

## MODUL 10

### Load Objek 3D

#### A. KOMPETENSI DASAR

- Memahami Inisialisasi Objek 3D
- Memahami Cara Meload file .obj

#### B. ALOKASI WAKTU

4 js (4x50 menit)

#### C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

#### D. DASAR TEORI

1. Pada konsep load file .obj, perlu ditambahkan kode program tambahan, diantaranya adalah file glm.h, Texture.h yang merupakan file header.
2. Kode untuk meload obj juga membutuhkan fungsi-fungsi khusus. Fungsi-fungsi tersebut terdapat pada file **glm.cpp**, **glmimg.cpp** dan **Texture.cpp**
3. Fungsi yang digunakan untuk meload file .obj pada kode sumber utama adalah:

```
GLMmodel* pmodel1 = NULL;

//fungsi untuk pemanggilan objek
void drawmodel_box(void){
    if (!pmodel1){
        //inisialisasi untuk pemanggilan objek
        pmodel1 = glmReadOBJ("objek.obj");
        if (!pmodel1) exit(0);
        glmUnitize(pmodel1);
        glmFacetNormals(pmodel1);
        glmVertexNormals(pmodel1, 90.0);
    }
    glmDraw(pmodel1, GLM_SMOOTH| GLM_TEXTURE);
}
```

**E. AKTIFITAS KELAS PRAKTIKUM**

1. Buatlah project dan beri nama LoadObj.
2. Letakkan file header glm.h, stdafx.h, dan Texture.h ke dalam folder yang telah dibuat, kemudian tambahkan kode untuk memanggil file header

```
#include "stdafx.h"
#include <stdlib.h>
#include <windows.h>
#include <glut.h>
#include "glm.h"
```

3. Letakkan kode glm.cpp dan glmimg.cpp dan Texture.cpp di folder project yang telah dibuat.
4. Letakkan file 3D “mouse.obj” ke dalam folder project yang telah dibuat.
5. Lengkapi kode program seperti syntax berikut ini:

```
#include "stdafx.h"
#include <stdlib.h>
#include <windows.h>
#include <glut.h>
#include "glm.h"

float tx=0, ty=0, tz=-3;

//var untuk parameter lighting
float x_light, y_light, z_light = 0.1;

//deklarasi pointer untuk load model
GLMmodel* pmodel1 = NULL;

//fungsi untuk pemanggilan objek
void drawmodel_box(void){
    if (!pmodel1){
        //inisialisasi untuk pemanggilan objek
        pmodel1 = glmReadOBJ("mouse.obj");
        if (!pmodel1) exit(0);
        glmUnitize(pmodel1);
        glmFacetNormals(pmodel1);
        glmVertexNormals(pmodel1, 90.0);
    }
    glmDraw(pmodel1, GLM_SMOOTH| GLM_TEXTURE);
}

//fungsi callback untuk menampilkan objek
void display(void){
    glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    /*--lighting--*/
    //deklarasi dan inisialisasi array untuk posisi lighting
    GLfloat LightPosition[] = {x_light, y_light, z_light,6};
    //LightPosition[0] = cx;
    //LightPosition[1] = cy;
    //LightPosition[2] = cz;

    //deklarasi dan inisialisasi array untuk parameter lighting specular
```

```
    GLfloat LightSpecular[] = {0.5, 0.5, 0.5, 1.0};
    //deklarasi dan inisialisasi array untuk parameter shine
    GLfloat Shine[] = {100};
    glClearDepth(1);
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
    glMaterialfv(GL_FRONT, GL_SPECULAR, LightSpecular);
    glMaterialfv(GL_FRONT, GL_SHININESS, Shine);
    glLightfv(GL_LIGHT0, GL_POSITION, LightPosition);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
    glTranslatef(tx, ty, tz);
    drawmodel_box();

    glutSwapBuffers();
}

void init(void){
    /*Lighting*/
    GLfloat LightPosition[] = {x_light, y_light, z_light, 20, 0};
    GLfloat LightAmbient[] = {0.0f, 0.0f, 0.0f, 0.0f};
    GLfloat LightDiffuse[] = {0.7f, 0.7f, 0.7f, 0.1f};
    GLfloat LightSpecular[] = {0.5f, 0.5f, 0.5f, 0.1f};
    GLfloat Shine[] = {80};
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glEnable(GL_DEPTH_TEST);
    glShadeModel (GL_SMOOTH);
}

void reshape (int w, int h){
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 20.0);
    glMatrixMode (GL_MODELVIEW);
}

void update (int value){
    glutPostRedisplay();
    glutTimerFunc(30, update, 0);
}

void main(int argc, char** argv){
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_RGB | GLUT_DEPTH | GLUT_DOUBLE);
    glutInitWindowSize (640, 480);
    glutInitWindowPosition (80, 0);
    glutCreateWindow ("Latihan Object Loader");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutTimerFunc(30, update, 0);
    glutMainLoop();
}
```