Dokumentasi Final Project Perpustakaan

Grafika Komputer C

Oleh :

Rendra Dwi Lingga P	5111100008
Vessa Rizky Oktavia	5112100052
Irfan Naufal	5112100066
Wimba Agra W	5112100102

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember Surabaya

;

I. Pendahuluan

Dokumen ini dibuat untuk mendokumentasikan Final Project matakuliah Grafika Komputer kelas C. Di dalam dokumen ini terdapat implementasi ruang perpustakaan ke dalam bentuk 3 dimensi disertai penjelasan tiap kodenya. Diharapkan dokumen ini dapat memudahkan user untuk menggunakan program kami nantinya.

II. Deskripsi Project

Kami membuat perpustakaan dengan penampakan sebuah ruangan berisi lemari, kursi, dan meja. Kursi dan meja dapat digerakkan secara translasi maupun rotasi. Lantai di dalam ruangan perpustakaan dapat berganti tekstur. Pencahayaan dalam ruangan juga dapat berganti-ganti menjadi lebih terang atau lebih gelap. Ruangan juga kami buat dapat dirotasi, zoom in, dan zoom out.

III. Implementasi

Library yang kami gunakan

```
#include <Windows.h>
#include <GL/GL.h>          //untuk openGL
#include <GL/GLU.h>
#include <GL/glut.h>
#include <iostream>         //untuk io c++
#include <string>
#include <stdlib.h>
#include "GLM/glm.h"        //untuk load object untuk viewing di openGL
#include "SOIL/soil.h"      //untuk load tekstur ke object di openGL
using namespace std;
```

Mendefinisikan model, tekstur, camera, viewmode, posisi, dan lighting

```
GLMmodel *mejakecil, *mejabesar, *kursi, *lemari, *pot, *rakbuku;
GLuint tex[5];
float camX=10.0, camY=2.0, camZ=10.0, angel=0, sc=1.0;
bool inc=0, viewmode=1, first=1;
int lighting=1, mode=1, pos=1, pindah=0, texture=1;
```

Membuat struct untuk titik pada object, kamera, dan menggerakkan object.

```
struct coor{
  float x, y, z;
  float sudut=0;
};
coor object[10];
coor camera;
coor moveobject[15];
```

Fungsi Init

```
void Init() {
    glEnable(GL_TEXTURE_2D);
    glEnable(GL_POINT_SMOOTH);
    glHint(GL_POINT_SMOOTH_HINT, GL_DONT_CARE);
    glEnable(GL_DEPTH_TEST);
    glEnable(GL_LIGHTING); // enable the light source
    glEnable(GL_LIGHT0);
    glShadeModel(GL_SMOOTH);
    glEnable(GL_NORMALIZE);
    glDepthFunc(GL_LESS);
    glEnable(GL_COLOR_MATERIAL);
    glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);

    //Load Object
    if (!mejakecil) {
        mejakecil = glmReadOBJ("object/Table/wooden_office_alder_table.obj");
        if (!mejakecil) exit(0);
        else cout<<"Load object meja success"<<endl; //gagal = exit
        glmUnitize(mejakecil);
    }
    if (!mejabesar) {
        mejabesar = glmReadOBJ("object/Table/wooden_table_office.obj");
        if (!mejabesar) exit(0);
        else cout<<"Load object meja success"<<endl; //gagal = exit
        glmUnitize(mejabesar);
    }
    if (!lemari) {
        lemari = glmReadOBJ("object/Lemari1/Table de nuit_Final.obj");
        if (!lemari) exit(0); //gagal = exit
        else cout<<"Load object lemari success"<<endl;
        glmUnitize(lemari);
    }
    if (!kursi) {
        kursi = glmReadOBJ("object/Chair/chair.obj");
        if (!kursi) exit(0); //gagal = exit
        else cout<<"Load object kursi success"<<endl;
        glmUnitize(kursi);
    }
```

```
  //Load Texture
  tex[1] = SOIL_load_OGL_texture
  (
    "C:\\Users\\user\\Desktop\\FinalProject\\Texture\\texture1.jpg",
    SOIL_LOAD_AUTO,
    SOIL_CREATE_NEW_ID,
    SOIL_FLAG_MIPMAPS | SOIL_FLAG_INVERT_Y | SOIL_FLAG_NTSC_SAFE_RGB
| SOIL_FLAG_COMPRESS_TO_DXT
  );
  if (tex[1]==0)
    cout<<"Load texture "<<1<<" FAILED"<<endl;
  else
    cout<<"Load texture "<<1<<" SUCCESS"<<endl;

  tex[2] = SOIL_load_OGL_texture
  (
    "C:\\Users\\user\\Desktop\\FinalProject\\Texture\\texture2.jpg",
    SOIL_LOAD_AUTO,
    SOIL_CREATE_NEW_ID,
    SOIL_FLAG_MIPMAPS | SOIL_FLAG_INVERT_Y | SOIL_FLAG_NTSC_SAFE_RGB
| SOIL_FLAG_COMPRESS_TO_DXT
  );
  if (tex[2]==0)
    cout<<"Load texture "<<2<<" FAILED"<<endl;
  else
    cout<<"Load texture "<<2<<" SUCCESS"<<endl;

  tex[3] = SOIL_load_OGL_texture
  (
    "C:\\Users\\user\\Desktop\\FinalProject\\Texture\\texture3.jpg",
    SOIL_LOAD_AUTO,
    SOIL_CREATE_NEW_ID,
    SOIL_FLAG_MIPMAPS | SOIL_FLAG_INVERT_Y | SOIL_FLAG_NTSC_SAFE_RGB
| SOIL_FLAG_COMPRESS_TO_DXT
  );
  if (tex[3]==0)
    cout<<"Load texture "<<3<<" FAILED"<<endl;
  else
    cout<<"Load texture "<<3<<" SUCCESS"<<endl;
```

```
   tex[4] = SOIL_load_OGL_texture
   (
      "C:\\Users\\user\\Desktop\\FinalProject\\Texture\\texture4.jpg",
      SOIL_LOAD_AUTO,
      SOIL_CREATE_NEW_ID,
      SOIL_FLAG_MIPMAPS | SOIL_FLAG_INVERT_Y | SOIL_FLAG_NTSC_SAFE_RGB
| SOIL_FLAG_COMPRESS_TO_DXT
   );
   if (tex[4]==0)
      cout<<"Load texture "<<4<<" FAILED"<<endl;
   else
      cout<<"Load texture "<<4<<" SUCCESS"<<endl;

   //Set material
   GLfloat mat_ambient[]={0.1f, 0.1f, 0.1f, 1.0f}; // gray
   GLfloat mat_diffuse[]={0.2f, 0.2f, 0.2f, 1.0f};
   GLfloat mat_specular[]={1.0f, 1.0f, 1.0f, 1.0f};
   GLfloat mat_shininess[]={10.0f};
   glMaterialfv(GL_FRONT, GL_AMBIENT,  mat_ambient);
   glMaterialfv(GL_FRONT, GL_DIFFUSE,  mat_diffuse);
   glMaterialfv(GL_FRONT, GL_SPECULAR,  mat_specular);
   glMaterialfv(GL_FRONT, GL_SHININESS,  mat_shininess);

}
//Membuat menu untuk memilih fitur
void mymenu(int option){
   switch (option){
      case 5:
         exit(0);

      case 11:
         viewmode=1;
         inc=!inc;
         break;
      case 12:
         if (sc<5) sc+=0.1;
         break;
      case 13:
         if (sc>0.5) sc-=0.1;
         break;
```

```
case 21:
   lighting=1;
   break;
case 22:
   lighting=2;
   break;
case 23:
   lighting=3;
   break;
case 24:
   lighting=4;
   break;

case 31:
   viewmode=0;
   pindah=1;
   break;
case 32:
   viewmode=0;
   pindah=2;
   break;
case 33:
   viewmode=0;
   pindah=3;
   break;
case 34:
   viewmode=0;
   pindah=4;
   break;
case 35:
   viewmode=0;
   pindah=10;
   break;
case 36:
   viewmode=0;
   pindah=11;
   break;

case 41:
   texture=1;
   break;
```

```
        case 42:
            texture=2;
            break;
        case 43:
            texture=3;
            break;
        case 44:
            texture=4;
            break;
    }
}

//Membuat fungsi keyboard untuk input melalui keyboard
void keyboard(unsigned char key, int x, int y){
    //On/Off rotation
    switch(key){
        case 32:
            inc=!inc;
            break;
        //Zoom - +
        case '-':
            if (sc>0.5) sc-=0.1;
            break;
        case '=':
            if (sc<5) sc+=0.1;
            break;
        //change lighting
        case '1':
            cout << "Lighting mode 1 active" << endl;
            lighting=1;
            break;
        case '2':
            cout << "Lighting mode 2 active" << endl;
            lighting=2;
            break;
        case '3':
            cout << "Lighting mode 3 active" << endl;
            lighting=3;
            break;
        case '4':
            cout << "Lighting mode 4 active" << endl;
        lighting=4;
    break;
    //Change Texture
    case '5':
        cout << "Texture type 1" << endl;
        texture=1;
        break;
    case '6':
        cout << "Texture type 2" << endl;
        texture=2;
        break;
    case '7':
        cout << "Texture type 3" << endl;
        texture=3;
        break;
    case '8':
        cout << "Texture type 4" << endl;
        texture=4;
    break;
}
    if (!viewmode){//Moveobject
        switch (key){
            case 13:
                pindah=0;
                break;
            case 'w':   //w
                moveobject[pindah].z-=0.1;
                break;
            case 'a':    //a
                moveobject[pindah].x-=0.1;
                break;
            case 's':   //s
                moveobject[pindah].z+=0.1;
                break;
            case 'd':   //d
                moveobject[pindah].x+=0.1;
                break;
            case '[':
                moveobject[pindah].sudut+=90;
                break;
            case ']':
                moveobject[pindah].sudut-=90;
                break;
          }
       }

}
//Fungsi untuk set lighting
void setlight(int pos){
   if (pos==1){
      GLfloat lightIntensity[]={0.8f, 0.8f, 0.8f, 1.0f};
      GLfloat light_position[]={-20.0f, 20.0f, 10.0f, 0.0f};
      glLightfv(GL_LIGHT0, GL_POSITION,  light_position);
      glLightfv(GL_LIGHT0, GL_DIFFUSE,  lightIntensity);
   }
   if (pos==2){
      GLfloat lightIntensity[]={0.6f, 0.6f, 0.6f, 1.0f};
      GLfloat light_position[]={20.0f, 20.0f, 10.0f, 0.0f};
      glLightfv(GL_LIGHT0, GL_POSITION,  light_position);
      glLightfv(GL_LIGHT0, GL_DIFFUSE,  lightIntensity);
   }
   if (pos==3){
      GLfloat lightIntensity[]={0.4f, 0.4f, 0.4f, 1.0f};
      GLfloat light_position[]={-20.0f, 20.0f, 10.0f, 0.0f};
      glLightfv(GL_LIGHT0, GL_POSITION,  light_position);
      glLightfv(GL_LIGHT0, GL_DIFFUSE,  lightIntensity);
   }
   if (pos==4){
      GLfloat lightIntensity[]={0.2f, 0.2f, 0.2f, 1.0f};
      GLfloat light_position[]={20.0f, 20.0f, 10.0f, 0.0f};
      glLightfv(GL_LIGHT0, GL_POSITION,  light_position);
      glLightfv(GL_LIGHT0, GL_DIFFUSE,  lightIntensity);
   }
}
//Membuat fungsi untuk menggambar object dinding
void wall(double thickness, double l, double w){
   glPushMatrix();
   glTranslated(0.5*l, 0 ,0.5*w);
   glScaled(l, thickness, w);
   glutSolidCube(1.0);
   glPopMatrix();
}
```

```
//Membuat fungsi untuk membuat suatu ruangan dengan fungsi wall
void buildarea(float l, float w, float h, float thickness){
  glPushMatrix();
  wall(thickness,l,w); // wall #1: in xz-plane

  glRotated(90.0, 0,0,1);
  wall(thickness,h,w); // wall #2: in yz-plane
  glPopMatrix();

  glPushMatrix();
  glRotated(-90, 1, 0, 0);
  wall(thickness,l,h);
  glPopMatrix();

  glPushMatrix();
  glTranslatef(l,0,0);
  glRotated(90.0, 0,0,1);
  wall(thickness,h,w); // wall #2: in yz-plane
  glPopMatrix();

  glPushMatrix();
  glTranslatef(0.4,0,w);
  glRotated(-90, 1, 0, 0);
  wall(thickness,l-0.4,h);
  glPopMatrix();
}
//Membuat fungsi untuk menggambar lantai
void drawfloor(int index){
  glBindTexture (GL_TEXTURE_2D, tex[index]);
  glBegin(GL_QUADS);
   glTexCoord2f(0,0); glVertex3f(-1.5f, 0.0101, 1.5f);
   glTexCoord2f(1,0); glVertex3f(1.5f, 0.0101, 1.5f);
   glTexCoord2f(1,1); glVertex3f(1.5f, 0.0101, -1.5f);
   glTexCoord2f(0,1); glVertex3f(-1.5f, 0.0101, -1.5f);
  glEnd();
}
//Membuat fungsi untuk menggambar meja
void drawtable(){
  //Gambar Meja
  glPushMatrix();
    glPushMatrix();
```

```
         glTranslatef(object[1].x, object[1].y, object[1].z);
         glScalef(0.35,0.13,0.3);
         glmDraw(mejabesar, GLM_SMOOTH | GLM_TEXTURE);
      glPopMatrix();
      //Gambar kursi
      glPushMatrix();
         glTranslatef(object[2].x, object[2].y, object[2].z);
         glScalef(0.1,0.1,0.1);
         glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
      glPopMatrix();
      glPushMatrix();
         glTranslatef(-0.2,0,0);
         glTranslatef(object[2].x, object[2].y, object[2].z);
         glScalef(0.1,0.1,0.1);
         glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
      glPopMatrix();
      glPushMatrix();
         glTranslatef(0,0,0.6);
         glRotatef(180.0,0,1,0);
         glPushMatrix();
            glTranslatef(object[2].x, object[2].y, object[2].z);
            glScalef(0.1,0.1,0.1);
            glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
         glPopMatrix();
         glPushMatrix();
            glTranslatef(-0.2,0,0);
            glTranslatef(object[2].x, object[2].y, object[2].z);
            glScalef(0.1,0.1,0.1);
            glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
         glPopMatrix();
      glPopMatrix();
   glPopMatrix();
}
//Membuat fungsi untuk menggambar lemari
void drawlemari(){
   //Gambar Lemari
   glPushMatrix();
      glTranslatef(moveobject[5].x, moveobject[5].y, moveobject[5].z);
      glScalef(0.5,0.1,0.1);
      glRotatef(moveobject[5].sudut,0,1,0);
      glmDraw(lemari, GLM_SMOOTH | GLM_TEXTURE);
```

```
  glPopMatrix();
  glPushMatrix();
    glTranslatef(moveobject[6].x, moveobject[6].y, moveobject[6].z);
    glScalef(0.5,0.1,0.1);
    glRotatef(moveobject[6].sudut,0,1,0);
    glmDraw(lemari, GLM_SMOOTH | GLM_TEXTURE);
  glPopMatrix();
  glPushMatrix();
    glTranslatef(moveobject[7].x, moveobject[7].y, moveobject[7].z);
    glScalef(0.5,0.1,0.1);
    glRotatef(moveobject[7].sudut,0,1,0);
    glmDraw(lemari, GLM_SMOOTH | GLM_TEXTURE);
  glPopMatrix();
  glPushMatrix();
    glTranslatef(moveobject[8].x, moveobject[8].y, moveobject[8].z);
    glScalef(0.5,0.1,0.1);
    glRotatef(180+moveobject[8].sudut,0,1,0);
    glmDraw(lemari, GLM_SMOOTH | GLM_TEXTURE);
  glPopMatrix();
  glPushMatrix();
    glTranslatef(moveobject[9].x, moveobject[9].y, moveobject[9].z);
    glScalef(0.5,0.1,0.1);
    glRotatef(180+moveobject[9].sudut,0,1,0);
    glmDraw(lemari, GLM_SMOOTH | GLM_TEXTURE);
  glPopMatrix();
}
//Membuat fungsi untuk menggambar semua object yang telah dibuat
void drawobject(){
  glPushMatrix();
    glTranslatef(moveobject[1].x, moveobject[1].y, moveobject[1].z);
    glRotatef(moveobject[1].sudut,0,1,0);
    drawtable();
  glPopMatrix();

  glPushMatrix();
    glTranslatef(moveobject[2].x, moveobject[2].y, moveobject[2].z);
    glRotatef(moveobject[2].sudut,0,1,0);
    drawtable();
  glPopMatrix();

  glPushMatrix();
```

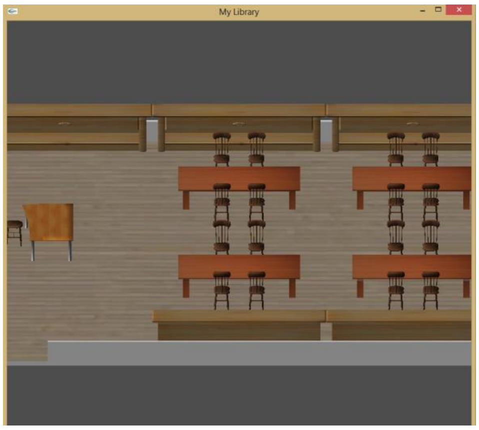
```
    glTranslatef(moveobject[3].x, moveobject[3].y, moveobject[3].z);
    glRotatef(moveobject[3].sudut,0,1,0);
    drawtable();
  glPopMatrix();

  glPushMatrix();
    glTranslatef(moveobject[4].x, moveobject[4].y, moveobject[4].z);
    glRotatef(moveobject[4].sudut,0,1,0);
    drawtable();
  glPopMatrix();

  drawlemari();

  //Kursi
  glPushMatrix();
    glTranslatef(moveobject[11].x, moveobject[11].y, moveobject[11].z);
    glScalef(0.1,0.1,0.1);
    glRotatef(90+moveobject[11].sudut,0,1,0);
    glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
  glPopMatrix();

  //Meja
  glPushMatrix();
    glTranslatef(moveobject[10].x, moveobject[10].y, moveobject[10].z);
    glScalef(0.4,0.2,0.4);
    glRotatef(-90+moveobject[10].sudut,0,1,0);
    glmDraw(mejakecil, GLM_SMOOTH | GLM_TEXTURE);
  glPopMatrix();

}
//Fungsi display untuk menampilkan di openGL
void display(){
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
  glClearColor(0.3,0.3,0.3,0);

  //Set light sources
  setlight(lighting);
  glLoadIdentity();

  //Set camera
  glMatrixMode(GL_PROJECTION);
```

```
   glLoadIdentity();
   double winHt=1.0; // half-height of the window
   glOrtho(-winHt*64/48.0,  winHt*64/48.0, -winHt,  winHt, 0.1, 1000.0);
   //glOrtho(100,  100, 100,  100, 0.1, 1000.0);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();

   gluLookAt(0, 5, 15, 0, 0, 0, 0.0, 1.0, 0.0);

   if (!viewmode){
      angel=0;
      //inc=0;
      //sc=1;
   }

   glPushMatrix();
      glRotatef(angel, 0, 1.0, 0);
      glScaled(sc,sc,sc);
      angel+=0.8*inc;
      glPushMatrix();
         glTranslatef(-1.5, 0 ,-1.5);
         buildarea(3.0, 3.0, 0.1, 0.02);
      glPopMatrix();
      drawfloor(texture);
      glTexEnvf(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_MODULATE);
      drawobject();
   glPopMatrix();

   glFlush();
   glutSwapBuffers();
   glutPostRedisplay();
}
//Fungsi main
int main(int argc, char** argv){
   if (first){             //fungsi untuk menentukan titik awal object
      moveobject[1].x=1, moveobject[1].y=0, moveobject[1].z=0.2;
      moveobject[2].x=1, moveobject[2].y=0, moveobject[2].z=-1;
      moveobject[3].x=0, moveobject[3].y=0, moveobject[3].z=0.2;
      moveobject[4].x=0, moveobject[4].y=0, moveobject[4].z=-1;
      moveobject[5].x=1, moveobject[5].y=0.1, moveobject[5].z=-1.4;
      moveobject[6].x=0, moveobject[6].y=0.1, moveobject[6].z=-1.4;
```

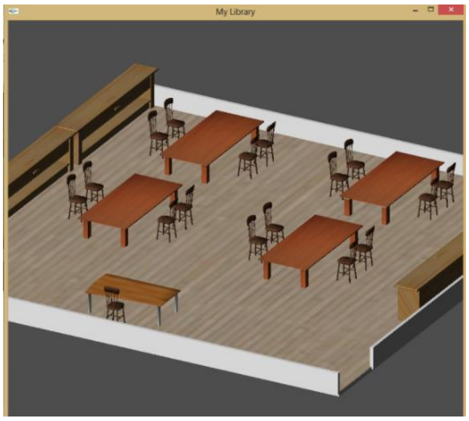
```
    moveobject[7].x=-1, moveobject[7].y=0.1, moveobject[7].z=-1.4;
    moveobject[8].x=1, moveobject[8].y=0.1, moveobject[8].z=1.4;
    moveobject[9].x=0, moveobject[9].y=0.1, moveobject[9].z=1.4;
    moveobject[10].x=-1.1, moveobject[10].y=0.05, moveobject[10].z=-0.1;
    moveobject[11].x=-1.3, moveobject[11].y=0.1, moveobject[11].z=-0.1;

    object[1].x=0.0, object[1].y=0.04, object[1].z=0.3;     //meja
    object[2].x=0.1, object[2].y=0.1, object[2].z=0.0;    //kursi

    first=0;
    system("cls");
  }

  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_DEPTH | GLUT_SINGLE | GLUT_RGBA);
  glutInitWindowPosition(300,0);
  glutInitWindowSize(800,800);
  glutCreateWindow("My Library");
  Init();
 //Membuat menu dan submenu untuk pilihan fitur
  int menu, submenu1, submenu2, submenu3, submenu4;
  submenu1 = glutCreateMenu(mymenu);
  glutAddMenuEntry("Rotate       (space)", 11);
  glutAddMenuEntry("Zoom in      (+)", 12);
  glutAddMenuEntry("Zoom out     (-)", 13);

  submenu2 = glutCreateMenu(mymenu);
  glutAddMenuEntry("Mode 1       (1)", 21);
  glutAddMenuEntry("Mode 2       (2)", 22);
  glutAddMenuEntry("Mode 3       (3)", 23);
  glutAddMenuEntry("Mode 4       (4)", 24);

  submenu3 = glutCreateMenu(mymenu);
  glutAddMenuEntry("Meja 1", 31);
  glutAddMenuEntry("Meja 2", 32);
  glutAddMenuEntry("Meja 3", 33);
  glutAddMenuEntry("Meja 4", 34);
  glutAddMenuEntry("Meja Depan", 35);
  glutAddMenuEntry("Kursi Depan", 36);

  submenu4 = glutCreateMenu(mymenu);
```

```
    glutAddMenuEntry("Type 1       (5)", 41);
    glutAddMenuEntry("Type 2       (6)", 42);
    glutAddMenuEntry("Type 3       (7)", 43);
    glutAddMenuEntry("Type 4       (8)", 44);

    menu = glutCreateMenu(mymenu);
    glutAddSubMenu("View Mode", submenu1);
    glutAddSubMenu("Lighting", submenu2);
    glutAddSubMenu("Move Object", submenu3);
    glutAddSubMenu("Change Texture", submenu4);
    glutAddMenuEntry("Exit", 5);
    glutAttachMenu(GLUT_RIGHT_BUTTON);

    glutKeyboardFunc(keyboard);
    glutDisplayFunc(display);
    glutMainLoop();
}
```

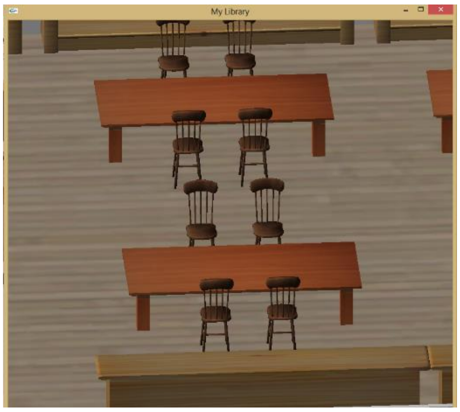
IV. Screenshot Program



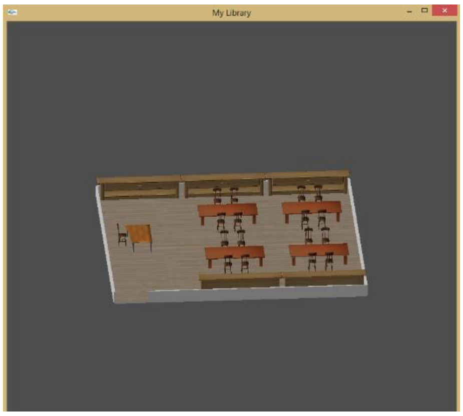
1. Pertama kali dijalankan



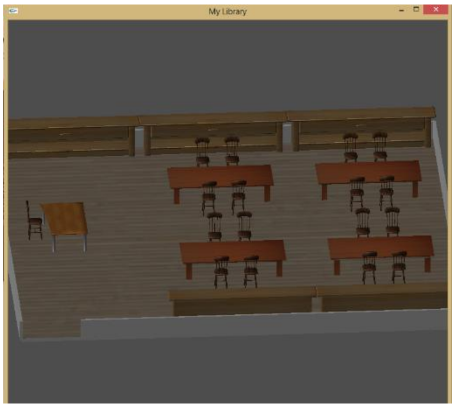
2. Ruangan dirotasi



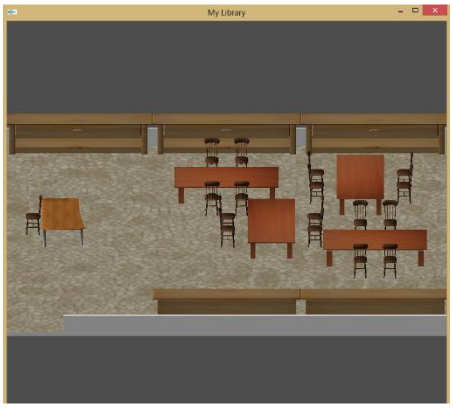
3. Zoom in



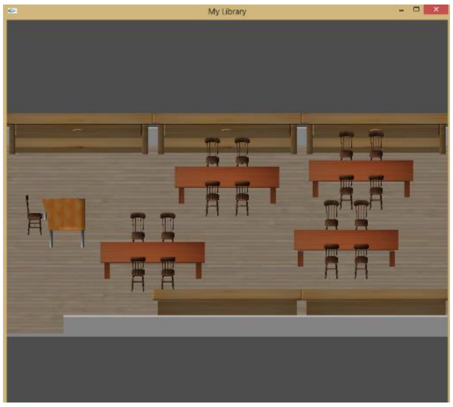
4. Zoom out



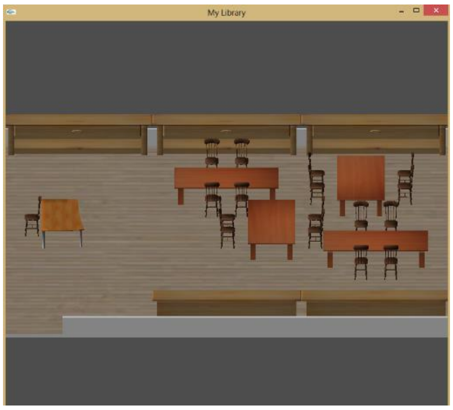
5. Perubahan lighting



6. Perubahan tekstur lantai



7. Translasi meja & kursi



8.Rotasi meja & kursi