DOKUMENTASI FINAL PROJECT
GRAFKOM C 2014/2015

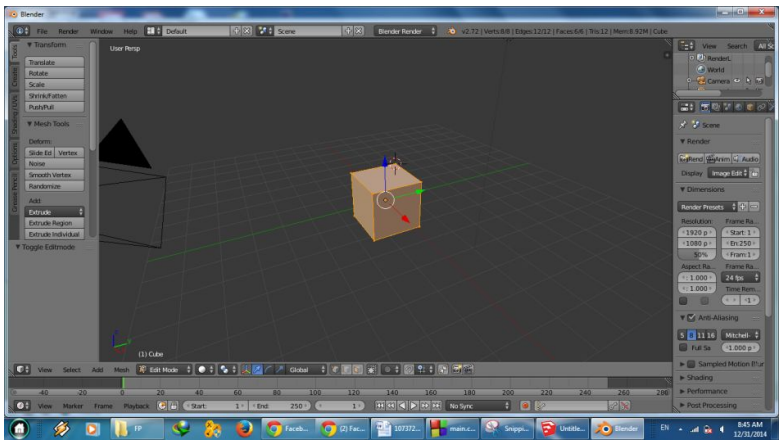
Membahas tentang bagaimana OpenGL dapat menghasilkan aplikasi sederhana yang semirip dengan Permainan Mix & Match. Di sini objek berupa 3D dapat melakukan translasi dan rotasi melalui cara yang ergonomis. Lalu, objek dapat memilih tekstur yang telah disediakan oleh aplikasi.

Berikut langkah-langkah pengerjaan Final Project:

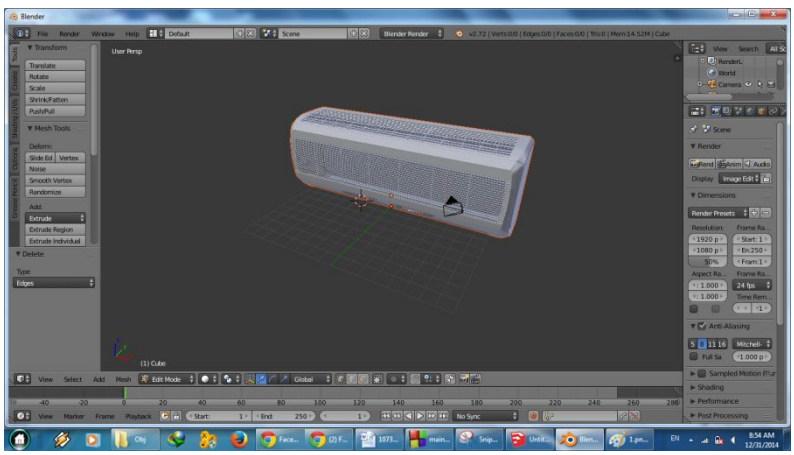
I. Penggantian Tekstur

1.Membuat Objek di Blender (Contoh Pembuatan Objek AC)

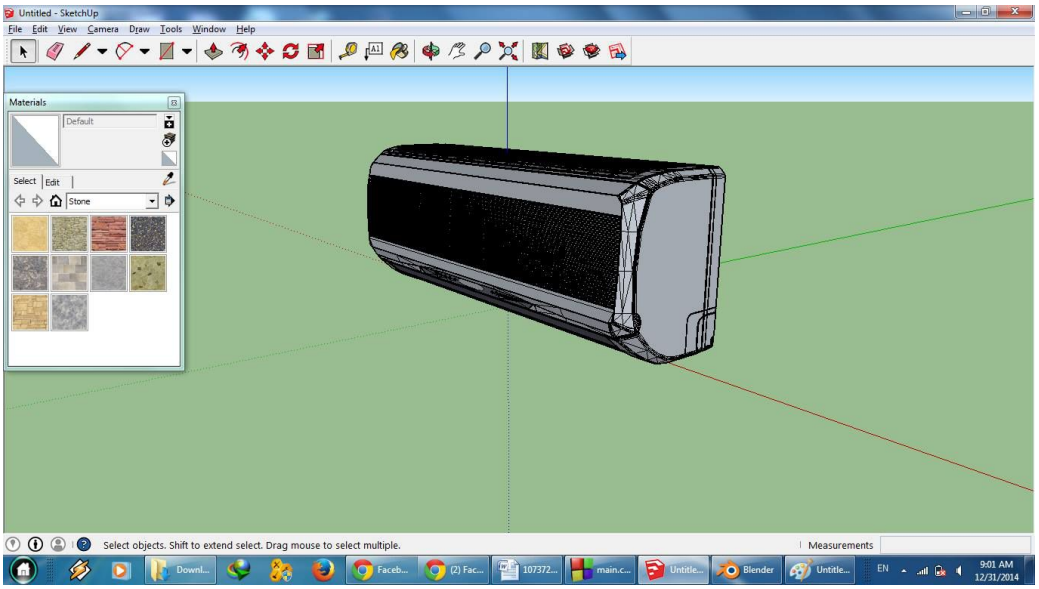
a. Pertama Download dan Install aplikasi Blender. (Free Version). Ini merupakan tampilan awal saat membuka blender



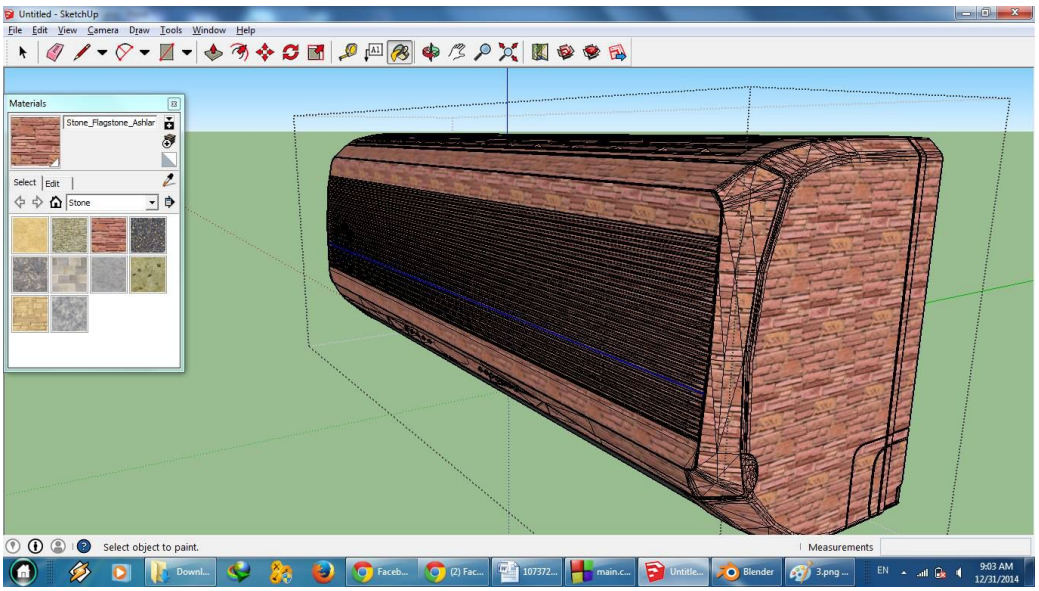
b.Setelah membuat Objek dibuat diblender, kita akan mengekspor Objek yang sudah kita buat untuk selanjutnya akan diberikan tekstur. Buka File-Export-Collada (Default).dae



c. Setelah diekspor ke .dae, di aplikasi SketchUp yang sudah kita install, maka selanjutnya kita akan mengimpor objek yang sudah kita buat di Blender tadi dengan cara File- Import.. Lalu arahkan ke .dae yang sudah kita buat.



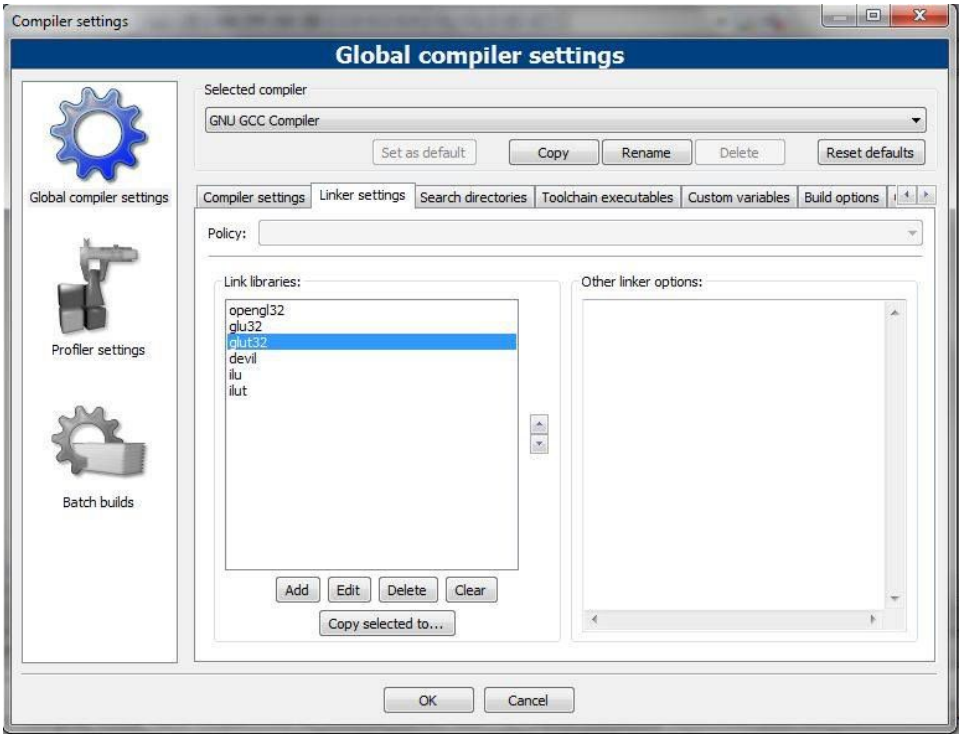
d. Setelah itu pada menubar, select Tools-Paint Bucket. Kemudian kita pilih texture yang kita inginkan kemudian Obj kita select kemudian kita berikan texture-nya.



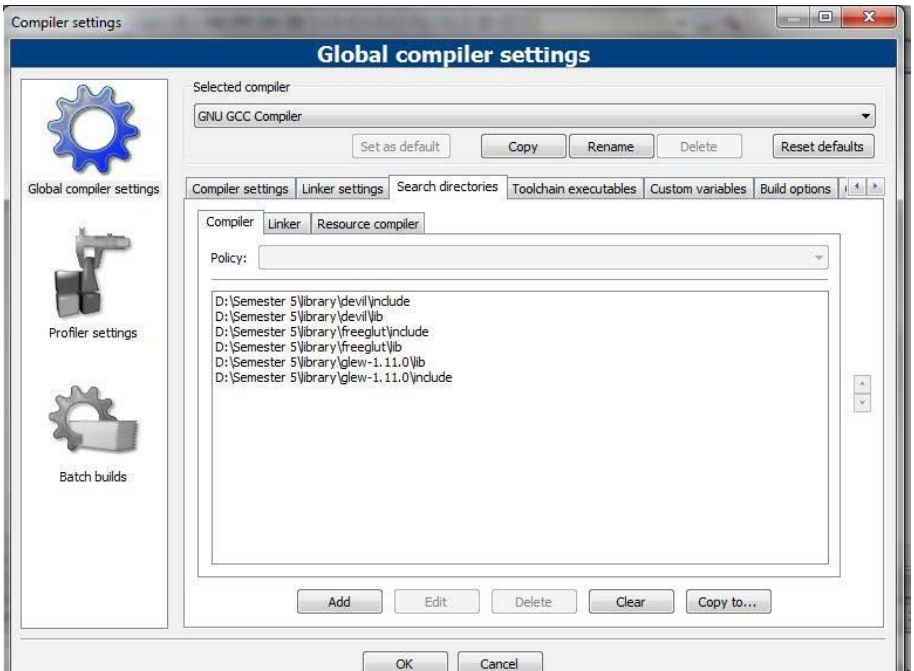
II. INSTALASI LIBRARY DEVIL (glm.h) Pada CodeBlock

Library ini berperan penting dalam load object 3D yang kami lakukan, seperti *method* GLMmodel, GLMReadOBJ, dan GLMUnitize.

1. Pada Menu Bar di Code Block, buka Settings-Compiler-Pilih Tab Linker Settings. Kemudian pada link libraries-nya ditambahkan seperti dibawah.



2. Kemudian pada Tab Search Directories kita tambahkan dan arahkan ke glut yang sudah didownload.



III. Fungsi

1. Menginisiasi variabel-variabel (global) yang dibutuhkan.

```
#include <Windows.h>
#include <gl\GL.h>
#include <gl\GLU.h>
#include <gl\glut.h>
#include "glm.h"
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <iostream>
using namespace std;

GLMmodel *sample,*sample2,*sample3,*sample4,*sample5, *sample6, *sample7, *sample8, *sample9;
float ratio;
int jml=10;
int obj[10];
float my=1.0;
int x[10], y[10], z[10];

int sudut[10];
int a[10],b[10];

char tekstur1[10][100];
char tekstur2[10][100];

// angle of rotation for the camera direction
float angle=0.0;
// actual vector representing the camera's direction
float lx=0.0f,lz=-1.0f;
// XZ position of the camera
float mx=0.0f,mz=3.0f;

```

2. Menginisiasi fungsi init() yang mengatur dasar-dasar *viewing* objek 3D.

```
void Init() {
//inisialisasi mode smoot dan texture dari gambar
    glEnable(GL_TEXTURE_2D);
    glEnable(GL_POINT_SMOOTH);
    glHint(GL_POINT_SMOOTH_HINT, GL_DONT_CARE);
    //Perspektif View
    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LESS);
    //blend warna untuk texture dan warna
    glEnable(GL_COLOR_MATERIAL);
    glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);

    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-100,500,-100,400,-100,100);
}
```

3. Membuat fungsi reshape() yang berfungsi mengatur tampilan layar.

```
115  void reshape(int w, int h){
116       if(h==0)   h=1;
117       ratio = 1.0 * w / h;
118       glMatrixMode(GL_PROJECTION);
119       glLoadIdentity();
120       glViewport(0,0,w,h);
121       gluPerspective(80,ratio,1,300);
122       glMatrixMode(GL_MODELVIEW);
123       glLoadIdentity();
124  }
```

4. Melakukan *load* objek pada void display().

```
53  void display() {
54      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
55      glClearColor(0,1,1,1);
56      glLoadIdentity();
57
58      gluLookAt(  mx, 1.0f, mz,
59                  mx+lx, my,  mz+lz,
60                  0.0f, 1.0f,  0.0f);
61
62      glPushMatrix();
63      if (!sample) {
64          sample = glmReadOBJ("objek/kamar.obj");
65          if (!sample) exit(0);
66          glmUnitize(sample);
67      }
68      glTranslatef(1,9,2);
69      glScalef(10,10,10);
70      glRotatef(0,0,1,0);
71      glTexEnvf(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_MODULATE);
72      glColor3ub(255,255,255);
73      glmDraw(sample, GLM_SMOOTH | GLM_TEXTURE);
74      glPopMatrix();
75
76      glPushMatrix();
77      if (!sample2) {
78          sample2 = glmReadOBJ("objek/ac2.obj");
79          if (!sample2) exit(0);
80          glmUnitize(sample2);
81      }
82      glTranslatef(x[2],y[2],z[2]);
83      glScalef(2,2,2);
84      glRotatef(sudut[2],0,1,0);
85      glRotatef(a[2],0,1,0);
86      glRotatef(b[2],1,0,0);
87      glTexEnvf(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_MODULATE);
88      glColor3ub(255,255,255);
89      glmDraw(sample2, GLM_SMOOTH | GLM_TEXTURE);
90      glPopMatrix();
```

Method yang digunakan untuk *load* objek adalah glmReadOBJ(*namafileobj*).

Sebelumnya di langkah 2, kami membuat variabel sample sebagai *dummy* untuk selanjutnya mengecek apakah objek sudah di-*load* atau belum (belum di-*load* adalah !sample).

Selanjutnya, setelah *load* objek, objek tersebut menyimpan informasi yang penting, di antaranya adalah posisi rotasi, translasi, dan skala.

Fungsi Untuk Mengganti Tekstur

```
void keyPressed(unsigned char key, int p, int q){
    char j[1];
    j[0] = key;
    if(key>='0' && key <='9')
    {
        int no;
        no = atoi(j);
        for(int i=0;i<jml;i++)
        {

            if(i==no){
                if(obj[i]==1)
                {
                    if(i==1)
                    {
                        if(strcmp(tekstur1[1],"objek/kursi2.obj")==0)
                        {
                            memset(tekstur1[1],'\0',sizeof(tekstur1[1]));
                            strcpy(tekstur1[1],"objek/kursi.obj");
                        }
                        else
                        {
                            memset(tekstur1[1],'\0',sizeof(tekstur1[1]));
                            strcpy(tekstur1[1],"objek/kursi2.obj");
```

Fungsi Untuk Menggerakkan Kamera

```
void display() {


    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glClearColor(0,0,0,0);
    glLoadIdentity();

    gluLookAt(  mx, 1.0f, mz,
                mx+lx, my,  mz+lz,
                0.0f, 1.0f,  0.0f);
```

Membuat fungsi processSpecialKeys() untuk melakukan suatu aksi yang ditangani oleh

keyboard. Di sini kami membuat fungsi untuk melakukan rotasi.

```
198  void processSpecialKeys(int key, int xx, int yy) {
199      obj[0]=1;
200      float fraction = 0.5f;
201
202          switch (key) {
203              case GLUT_KEY_LEFT :
204                  angle -= 0.1f;
205                  lx = sin(angle);
206                  lz = -cos(angle);
207                  break;
208              case GLUT_KEY_RIGHT :
209                  angle += 0.1f;
210                  lx = sin(angle);
211                  lz = -cos(angle);
212                  break;
213              case GLUT_KEY_UP :
214                  mx += lx * fraction;
215                  mz += lz * fraction;
216                  break;
217              case GLUT_KEY_DOWN :
218                  mx -= lx * fraction;
219                  mz -= lz * fraction;
220                  break;
221          }
```

Memanggil fungsi-fungsi pada main().

```
224  int main(int argc, char** argv){
225      glutInit(&argc, argv);
226      glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGBA);
227
228      strcpy(tex_kursi[1],"objek/kursi.obj");
229      strcpy(tex_papan[1],"objek/papan_tulis.obj");
230      strcpy(tex_medos[1],"objek/mejadosen.obj");
231      strcpy(tex_kudos[1],"objek/kursidosen.obj");
232      strcpy(tex_cpupc[1],"objek/CPU.obj");
233      strcpy(tex_airco[1],"objek/ac.obj");
234
235      glutInitWindowPosition(0, 0);
236      glutInitWindowSize(1200, 600);
237      glutCreateWindow("Classroom");
238      glutDisplayFunc(display);
239      glutIdleFunc(display);
240      glutReshapeFunc(reshape);
241      glutKeyboardFunc(keyPressed);
242      glutSpecialFunc(processSpecialKeys);
243      Init();
244      glutMainLoop();
245  }
```

Fungsi Untuk Me-Load salah satu Object

```
/* Menampilkan objek kursi mahasiswa */
    for (int i = 1; i <= kolom; i++) {
        for (int j = 1; j <= baris; j++) {
            glPushMatrix();
            kursi = glmReadOBJ(tekstur1[1]);
            glmUnitize(kursi);
            glTranslatef(x_pos*i-2, 0, -z_pos*j);
            glScalef(skala, skala, skala);
            glTexEnvf(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_MODULATE);
            glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
            glPopMatrix();

            glPushMatrix();
            kursi = glmReadOBJ(tekstur1[1]);
            glmUnitize(kursi);
            glTranslatef(-x_pos*i-2, 0, -z_pos*j);
            glScalef(skala, skala, skala);
            glTexEnvf(GL_TEXTURE_ENV,GL_TEXTURE_ENV_MODE,GL_MODULATE);
            glmDraw(kursi, GLM_SMOOTH | GLM_TEXTURE);
            glPopMatrix();
```