Dokumentasi Final Project “Summer Beach Paradise”

Grafika Komputer C

Dibuat oleh:

Fajar Setiawan (5112100010)

Yohanes Aditya Sutanto (5112100135)

Ikrom Aulia Fahdi (5112100157)

Otniel Yehezkiel (5112100212)

Jurusan Teknik Informatika

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh November

Surabaya

## I. Pendahuluan

Dokumen ini dibuat untuk mendokumentasikan Final Project matakuliah Grafika Komputer kelas C. Di dalam dokumen ini terdapat implementasi Tata ruang sebuah pantai 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 ruang pantai berisi pepohonan, langit, pantai, gedung dan rumah pantai, serta object lainnya. Gedung dan rumah pantai dapat digerakkan secara translasi maupun rotasi. Beberapa object dalam ruang juga dapat berganti tekstur. Pencahayaan dalam ruang berada diatas yang menjadi sinar matahari. Posisi kamera dan titik lihat pada program ini kami buat dapat diubah sesuai keinginan.

## III. Implementasi

```
//List Library
#include <stdio.h>
#include <Windows.h>
#include <gl\GL.h>
#include <gl\GLU.h>
#include <gl\glut.h>
#include <cstdio>
#include <algorithm>
#include <cmath>
#include <iostream>
#include "glm.h"

using namespace std;
#define PI 3.14159265

GLMmodel *sample0, *sample1, *sample2, *sample3, *sample4, *sample5, *sample6,
*sample7, *sample8, *objTerrain, *objOcean;
float ratio;
float sumbux[101],sumbuz[101],  angle=0;
float radians     = 0.0f;                          // camera angle in radians
float cameraX, cameraY, cameraZ;
float testcmX = 0, testcmY = 3, testcmZ = 5, testlookX = 0, testlookY = 0, testlookZ = -5;
int ganti[101];
float nilaiarray[101][7],cx,cz, selisihx, selisihz, nilaiselisih[101][7], fixarray[101][7];
int b=1;
bool cameraActive = false;
int putar[101], putarb[101];
int kanan, bawah, atas, kiri, kanankanan[3], bawahbawah[3], atasatas[3], kirikiri[3];

typedef struct Points{
    float x,y,z;
```

```
}Points;
Points pts[100];

int num = 0, a = 0;

void Init()
{
    glEnable(GL_TEXTURE_2D);
    glEnable(GL_POINT_SMOOTH);
    glHint(GL_POINT_SMOOTH_HINT, GL_DONT_CARE);

    glEnable(GL_DEPTH_TEST);
    glDepthFunc(GL_LESS);

    glEnable(GL_LIGHTING);
      glEnable(GL_LIGHT0);

    glEnable(GL_COLOR_MATERIAL);
    glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
}

//Collision masih belum bisa
void cekcollision(){
    //kanankanan={};
    //bawahbawah={};
    //kirikiri={};
    //atasatas={};
    for(int i=1;i<2;i++){
        for(int j=2; j<3;j++){
            kanan=kiri=bawah=atas=0;
            if(((fixarray[j][0] >= fixarray[i][0]) || (fixarray[j][0] <= fixarray[i][i])) &&
((fixarray[j][3] >= fixarray[i][2]) || (fixarray[j][3] <= fixarray[i][3]))){
                kiri++;
                bawah++;
            }
            if(((fixarray[j][1] >= fixarray[i][0]) || (fixarray[j][1] <= fixarray[i][i])) &&
((fixarray[j][3] >= fixarray[i][2]) || (fixarray[j][3] <= fixarray[i][3]))){
                kanan++;
                bawah++;
            }
            if(((fixarray[j][1] >= fixarray[i][0]) || (fixarray[j][1] <= fixarray[i][i])) &&
((fixarray[j][2] >= fixarray[i][2]) || (fixarray[j][2] <= fixarray[i][3]))){
                kanan++;
                atas++;
            }
            if(((fixarray[j][0] >= fixarray[i][0]) || (fixarray[j][0] <= fixarray[i][i])) &&
((fixarray[j][2] >= fixarray[i][2]) || (fixarray[j][2] <= fixarray[i][3]))){
                kiri++;
                atas++;
            }
            if(bawah==2) sumbuz[j] += 0.1;
            else if(kanan==2) sumbux[j] -= 0.1;
            else if(kiri==2) sumbux[j] += 0.1;
            else if(atas==2) sumbuz[j] -= 0.1;
            else if(bawah==1) sumbuz[j] += 0.1;
            else if(kanan==1) sumbux[j] -= 0.1;
            else if(kiri==1) sumbux[j] += 0.1;
            else if(atas==1) sumbuz[j] -= 0.1;
        }
    }
}

//Membuat langit
void skybox()
{
    glPushMatrix();
    if(!sample0)
    {
        sample0 = glmReadOBJ("skybox_pasture.obj",num);
        if(!sample0) exit(0);
        glmUnitize(sample0);
    }
    glTranslated(0,-4,-3);
    glRotated(270, 1, 0, 0);
    glScalef(50, 50, 50);

    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample0, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat laut
void ocean()
{
    glPushMatrix();
    if(!objOcean){                              //buat load obj
        objOcean = glmReadOBJ("ocean/1.obj",num);
        if(!objOcean) exit(0);
        glmUnitize(objOcean);
    }
    glTranslatef(0, -3, 0);

    glScalef(55, 55, 55);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(objOcean, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
```

```
}

//Membuat terrain pantai
void terrain()
{
   glPushMatrix();
   if(!objTerrain){                                    //buat load obj
      objTerrain = glmReadOBJ("island/1.obj",num);
      if(!objTerrain) exit(0);
      glmUnitize(objTerrain);
   }
   glTranslatef(0, -3, 3);

   glScalef(43, 43, 43);
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
   glColor3ub(255, 255, 255);
   glmDraw(objTerrain, GLM_SMOOTH | GLM_TEXTURE);
   glPopMatrix();
   glFlush();
}

//Membuat rumah pantai yang terbuat dari kayu
void display2()
{
   glPushMatrix();
   if(!sample2){                                       //buat load obj
      sample2 = glmReadOBJ("beachhouse/1.obj",num);
      if(!sample2) exit(0);
      glmUnitize(sample2);
      //fixarray[2][0]=nilaiarray[2][0]=xmin();
      //fixarray[2][1]=nilaiarray[2][1]=xmax();
      //fixarray[2][2]=nilaiarray[2][2]=zmin();
      //fixarray[2][3]=nilaiarray[2][3]=zmax();
      //nilaiarray[2][4]=ymin();
   }

   if(num == 2){
      glTranslatef(sumbux[2]-10, 2.6, sumbuz[2]-2);
      glRotated(putar[2], 0, 1, 0);
   }
   else{
      glTranslatef(sumbux[2]-10, 2.6, sumbuz[2]-2);
      glRotated(putar[2], 0, 1, 0);
   }

   glScalef(10, 10, 10);
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
   glColor3ub(255, 255, 255);
   glmDraw(sample2, GLM_SMOOTH | GLM_TEXTURE);
   glPopMatrix();
```

```
    glFlush();
}

//Membuat object tong sampah, texture dapat diubah
void display3()
{
    glPushMatrix();
    if(!sample3){                                   //buat load obj
        sample3 = glmReadOBJ("tongsampah_1/tongsampah.obj",num);
        if(!sample3) exit(0);
        glmUnitize(sample3);
        fixarray[3][0]=nilaiarray[3][0]=xmin();
        fixarray[3][1]=nilaiarray[3][1]=xmax();
        fixarray[3][2]=nilaiarray[3][2]=zmin();
        fixarray[3][3]=nilaiarray[3][3]=zmax();
        nilaiarray[3][4]=ymin();
    }
    if(ganti[3]>0){                                 //buat ganti texture
        sample3 = glmReadOBJ("tongsampah_1/tongsampah.obj",ganti[3]);
        ganti[3]==0;
        if(!sample3) exit(0);
        glmUnitize(sample3);
        //printf("%d ganti2 ",ganti[2]);
    }
    if(num == 3){
        glTranslatef(sumbux[3]+3, -1.1, sumbuz[3]);
        glRotated(putar[3], 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[3]+3, -1.1, sumbuz[3]);
        glRotated(putar[3], 0, 1, 0);
    }

    glScalef(0.5, 0.5, 0.5);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample3, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat cottage
void display4()
{
    glPushMatrix();
    if(!sample4)
    {
        sample4 = glmReadOBJ("cottage/1.obj",num);
        if(!sample4) exit(0);
        glmUnitize(sample4);
    }

    if(num == 4){
        glTranslatef(sumbux[4]+9, 0, sumbuz[4]-3);
        glRotated(putar[4]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[4]+9, 0, sumbuz[4]-3);
        glRotated(putar[4]+90, 0, 1, 0);
    }

    glScalef(3, 3, 3);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample4, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object payung, texture dapat diubah
void display5()
{
    glPushMatrix();
    if(!sample5){                                   //buat load obj
        sample5 = glmReadOBJ("Payung_1/payung.obj",num);
        if(!sample5) exit(0);
        glmUnitize(sample5);
    }
    if(ganti[5]>0){                                 //buat ganti texture
        sample5 = glmReadOBJ("Payung_1/payung.obj",ganti[5]);
        ganti[5]==0;
        if(!sample5) exit(0);
        glmUnitize(sample5);
        //printf("%d ganti2 ",ganti[2]);
    }
    if(num == 5){
        glTranslatef(sumbux[5]+15, -0.5, sumbuz[5]-2);
        glRotated(putar[5], 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[5]+15, -0.5, sumbuz[5]-2);
        glRotated(putar[5], 0, 1, 0);
    }

    glScalef(10, 10, 10);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample5, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
```

```
}

//Membuat object pohon
void display6()
{
   glPushMatrix();
   if(!sample6)
   {
      sample6 = glmReadOBJ("pohon/pohon.obj",num);
      if(!sample6) exit(0);
      glmUnitize(sample6);
   }

   if(num == 6){
      glTranslatef(sumbux[6]-15, 0, sumbuz[6]+25);
      glRotated(putar[6]+90, 0, 1, 0);
   }
   else{
      glTranslatef(sumbux[6]-15, 0, sumbuz[6]+25);
      glRotated(putar[6]+90, 0, 1, 0);
   }

   glScalef(15, 15, 15);
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
   glColor3ub(255, 255, 255);
   glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
   glPopMatrix();
   glFlush();
}

//Membuat object rumah pantai modern
void display7()
{
   glPushMatrix();
   if(!sample7){                              //buat load obj
      sample7 = glmReadOBJ("Motel/motel.obj",num);
      if(!sample7) exit(0);
      glmUnitize(sample7);
   }
   if(num == 7){
      glTranslatef(sumbux[7]-15, 1.8, sumbuz[7]+15);
      glRotated(putar[7], 0, 1, 0);
   }
   else{
      glTranslatef(sumbux[7]-15, 1.8, sumbuz[7]+15);
      glRotated(putar[7], 0, 1, 0);
   }

   glScalef(18, 18, 18);
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
```

```
    glColor3ub(255, 255, 255);
    glmDraw(sample7, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object apartemen
void display8()
{
    glPushMatrix();
    if(!sample8)
    {
        sample8 = glmReadOBJ("Apartment/apartment.obj",num);
        if(!sample8) exit(0);
        glmUnitize(sample8);
    }

    if(num == 8){
        glTranslatef(sumbux[8]+9, 14, sumbuz[8]+15);
        glRotated(putar[8]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[8]+9, 14, sumbuz[8]+15);
        glRotated(putar[8]+90, 0, 1, 0);
    }

    glScalef(16, 16, 16);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample8, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6a()
{
    glPushMatrix();
    if(!sample6)
    {
        sample6 = glmReadOBJ("pohon/pohon.obj",num);
        if(!sample6) exit(0);
        glmUnitize(sample6);
    }

    if(num == 6){
        glTranslatef(sumbux[6]-13, 0, sumbuz[6]+27);
        glRotated(putar[6]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[6]-13, 0, sumbuz[6]+27);
        glRotated(putar[6]+90, 0, 1, 0);
    }

    glScalef(15, 15, 15);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6b()
{
    glPushMatrix();
    if(!sample6)
    {
        sample6 = glmReadOBJ("pohon/pohon.obj",num);
        if(!sample6) exit(0);
        glmUnitize(sample6);
    }

    if(num == 6){
        glTranslatef(sumbux[6]-15, 0, sumbuz[6]+30);
        glRotated(putar[6]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[6]-15, 0, sumbuz[6]+30);
        glRotated(putar[6]+90, 0, 1, 0);
    }

    glScalef(15, 15, 15);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6c()
{
    glPushMatrix();
    if(!sample6)
    {
        sample6 = glmReadOBJ("pohon/pohon.obj",num);
        if(!sample6) exit(0);
        glmUnitize(sample6);
    }
    if(num == 6){
        glTranslatef(sumbux[6]-10, 0, sumbuz[6]+27);
        glRotated(putar[6]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[6]-10, 0, sumbuz[6]+27);
        glRotated(putar[6]+90, 0, 1, 0);
    }

    glScalef(15, 15, 15);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6d()
{
    glPushMatrix();
    if(!sample6)
    {
        sample6 = glmReadOBJ("pohon/pohon.obj",num);
        if(!sample6) exit(0);
        glmUnitize(sample6);
    }

    if(num == 6){
        glTranslatef(sumbux[6]-11, 0, sumbuz[6]+26);
        glRotated(putar[6]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[6]-11, 0, sumbuz[6]+26);
        glRotated(putar[6]+90, 0, 1, 0);
    }

    glScalef(15, 15, 15);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6e()
{
    glPushMatrix();
    if(!sample6)
    {
        sample6 = glmReadOBJ("pohon/pohon.obj",num);
        if(!sample6) exit(0);
        glmUnitize(sample6);
    }

    if(num == 6){
        glTranslatef(sumbux[6]-10, 0, sumbuz[6]+29);
        glRotated(putar[6]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[6]-10, 0, sumbuz[6]+29);
        glRotated(putar[6]+90, 0, 1, 0);
    }

    glScalef(15, 15, 15);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6f()
{
    glPushMatrix();
    if(!sample6)
    {
        sample6 = glmReadOBJ("pohon/pohon.obj",num);
        if(!sample6) exit(0);
        glmUnitize(sample6);
    }

    if(num == 6){
        glTranslatef(sumbux[6]-8, 0, sumbuz[6]+25);
        glRotated(putar[6]+90, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[6]-8, 0, sumbuz[6]+25);
        glRotated(putar[6]+90, 0, 1, 0);
    }

    glScalef(15, 15, 15);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
```

```
}

//Membuat object pohon dengan posisi yang berbeda dari yang sudah ada
void display6g()
{
   glPushMatrix();
   if(!sample6)
   {
      sample6 = glmReadOBJ("pohon/pohon.obj",num);
      if(!sample6) exit(0);
      glmUnitize(sample6);
   }

   if(num == 6){
      glTranslatef(sumbux[6]-17, 0, sumbuz[6]+25);
      glRotated(putar[6]+90, 0, 1, 0);
   }
   else{
      glTranslatef(sumbux[6]-17, 0, sumbuz[6]+25);
      glRotated(putar[6]+90, 0, 1, 0);
   }

   glScalef(15, 15, 15);
   glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
   glColor3ub(255, 255, 255);
   glmDraw(sample6, GLM_SMOOTH | GLM_TEXTURE);
   glPopMatrix();
   glFlush();
}

//Fungsi display utama
void display()
{
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glClearColor(0,1,1,1);
   glLoadIdentity();
   glPushMatrix();
   //glRotated(rotasi[num], 0, 1, 0);
   if(!sample1)
   {
      sample1 = glmReadOBJ("farmhouse_1/FarmhouseOBJ.obj",num);

      if(!sample1) exit(0);
      glmUnitize(sample1);
      fixarray[1][0]=nilaiarray[1][0]=xmin();
      fixarray[1][1]=nilaiarray[1][1]=xmax();
      fixarray[1][2]=nilaiarray[1][2]=zmin();
      fixarray[1][3]=nilaiarray[1][3]=zmax();
      nilaiarray[1][4]=ymin();
   }
```

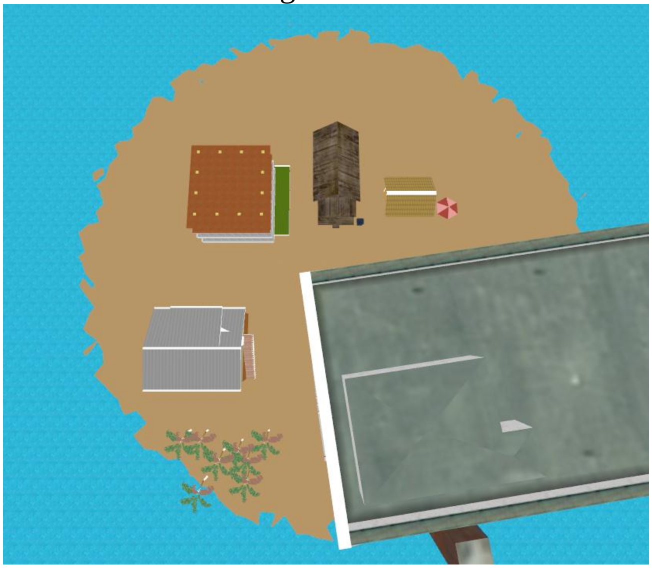
```
if(ganti[1]>0){                                    //buat ganti texture
    sample1 = glmReadOBJ("farmhouse_1/FarmhouseOBJ.obj",ganti[1]);
    ganti[1]==0;
    if(!sample1) exit(0);
    glmUnitize(sample1);
    //printf("%d ganti2 ",ganti[2]);
}
radians = float(PI*(angle-90.0f)/180.0f);

  cameraX = 0 + sin(radians);
  cameraZ = 3 + cos(radians);

  //printf("%f lookx\n",cameraX);
//printf("%f lookz\n",cameraZ);

//gluLookAt(0, 3, 5, 0, 0, -5, 0.0, 1.0, 0.0);
gluLookAt(testcmX,testcmY,testcmZ,testlookX,testlookY,testlookZ,0,1,0);

GLfloat ambientLight[] = {1.0f, 1.0f, 1.0f, 1.0f};
  glLightModelfv(GL_LIGHT_MODEL_AMBIENT, ambientLight);

  GLfloat lightColor[] = { 2.0f, 2.0f, 3.0f, 0.0f };
  GLfloat lightPos[] = { -2.0f, -2.0f, -3.0f };
  glLightfv(GL_LIGHT0, GL_DIFFUSE, lightColor);
  glLightfv(GL_LIGHT0, GL_POSITION, lightPos);
//if(num == 100) glRotatef(sumbux[100],0,1,0);
skybox();
ocean();
terrain();
//display0();
display2();
display3();
display4();
display5();
display6();
display6a();
display6b();
display6c();
display6d();
display6e();
display6f();
display6g();
display7();
display8();

for(int i=0; i<2; i++){
    glColor3f(0,1,0);
    if(num == i+1){
        putarb[num]=putar[num];
        putarb[num]=putarb[num]%180;
        //printf("%d ea ", putarb[num]);
        if(putarb[num]==90 || putarb[num]==-90){
           cx = (nilaiarray[i+1][0] + nilaiarray[i+1][1]) / 2.0;
           cz = (nilaiarray[i+1][2] + nilaiarray[i+1][3]) / 2.0;
           selisihx = nilaiarray[i+1][1] - cx;
           selisihz = nilaiarray[i+1][3] - cz;
           nilaiselisih[i+1][0]=cx-selisihz;
           nilaiselisih[i+1][1]=cx+selisihz;
           nilaiselisih[i+1][2]=cz-selisihx;
           nilaiselisih[i+1][3]=cz+selisihx;

           glBegin(GL_POLYGON);

glVertex3f(nilaiselisih[i+1][0]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiseli
sih[i+1][3]+pts[i].z+sumbuz[num]);

glVertex3f(nilaiselisih[i+1][1]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiseli
sih[i+1][3]+pts[i].z+sumbuz[num]);

glVertex3f(nilaiselisih[i+1][1]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiseli
sih[i+1][2]+pts[i].z+sumbuz[num]);

glVertex3f(nilaiselisih[i+1][0]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiseli
sih[i+1][2]+pts[i].z+sumbuz[num]);
           glEnd();
           //printf("jajal");
           fixarray[i+1][0]=nilaiselisih[i+1][0];
           fixarray[i+1][1]=nilaiselisih[i+1][1];
           fixarray[i+1][2]=nilaiselisih[i+1][2];
           fixarray[i+1][3]=nilaiselisih[i+1][3];
        }
        else{
           glBegin(GL_POLYGON);

glVertex3f(nilaiarray[i+1][0]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiarray
[i+1][3]+pts[i].z+sumbuz[num]);

glVertex3f(nilaiarray[i+1][1]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiarray
[i+1][3]+pts[i].z+sumbuz[num]);

glVertex3f(nilaiarray[i+1][1]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiarray
[i+1][2]+pts[i].z+sumbuz[num]);

glVertex3f(nilaiarray[i+1][0]+pts[i].x+sumbux[num],nilaiarray[i+1][4]+pts[i].y,nilaiarray
[i+1][2]+pts[i].z+sumbuz[num]);
           glEnd();
           fixarray[i+1][0]=nilaiarray[i+1][0];
           fixarray[i+1][1]=nilaiarray[i+1][1];
           fixarray[i+1][2]=nilaiarray[i+1][2];
```

```
            fixarray[i+1][3]=nilaiarray[i+1][3];
          }
          //nilaiarray[i+1][0]=nilaiselisih[i+1][0];
          //nilaiarray[i+1][1]=nilaiselisih[i+1][1];
          //nilaiarray[i+1][2]=nilaiselisih[i+1][2];
          //nilaiarray[i+1][3]=nilaiselisih[i+1][3];
          //glRotated(putar[num], 0, 1, 0);
        }
        else{
          glBegin(GL_POLYGON);

glVertex3f(fixarray[i+1][0]+pts[i].x+sumbux[i+1],nilaiarray[i+1][4]+pts[i].y,fixarray[i+1
][3]+pts[i].z+sumbuz[i+1]);

glVertex3f(fixarray[i+1][1]+pts[i].x+sumbux[i+1],nilaiarray[i+1][4]+pts[i].y,fixarray[i+1
][3]+pts[i].z+sumbuz[i+1]);

glVertex3f(fixarray[i+1][1]+pts[i].x+sumbux[i+1],nilaiarray[i+1][4]+pts[i].y,fixarray[i+1
][2]+pts[i].z+sumbuz[i+1]);

glVertex3f(fixarray[i+1][0]+pts[i].x+sumbux[i+1],nilaiarray[i+1][4]+pts[i].y,fixarray[i+1
][2]+pts[i].z+sumbuz[i+1]);
          glEnd();
          //glRotated(putar[i+1], 0, 1, 0);
        }

        //if(b==1) printf("a");
    }
    b=0;
    glRotated(0, 0, 1, 0);
    if(num == 1){
        glTranslatef(sumbux[1], 0.8, -5+sumbuz[1]);
        glRotated(putar[1]+180, 0, 1, 0);
    }
    else{
        glTranslatef(sumbux[1], 0.8, -5+sumbuz[1]);
        glRotated(putar[1]+180, 0, 1, 0);
    }
    //}
    //cekcollision();

    glScalef(6, 6, 6);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    glColor3ub(255, 255, 255);
    glmDraw(sample1, GLM_SMOOTH | GLM_TEXTURE);
    glPopMatrix();
    glFlush();
    glutSwapBuffers();
}
```

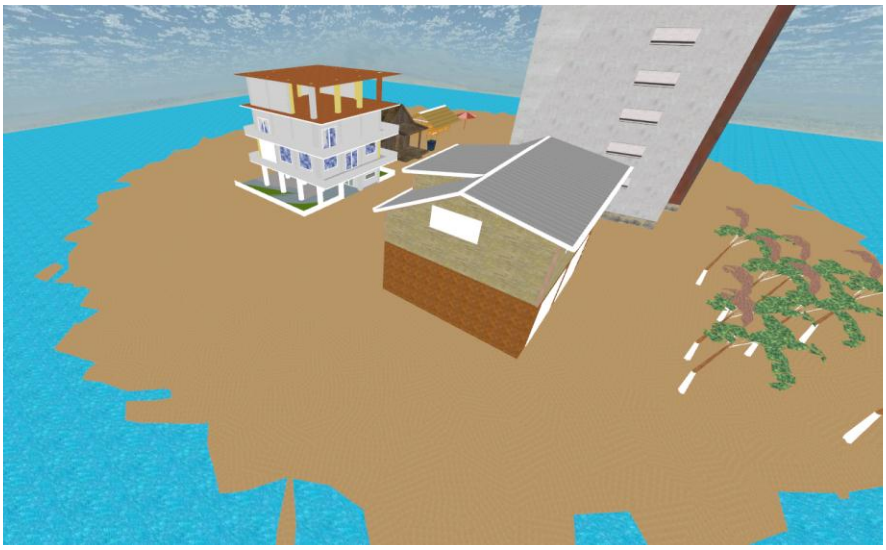
```
void reshape(int w, int h)
{
   if (h == 0) h = 1;
   ratio = 1.0 * w / h;
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   glViewport(0, 0, w, h);
   gluPerspective(80, ratio, 1, 300);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
}

//int onMouse = 1;
/*void mouseClicks(int button, int state, int x, int y)
{
   if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
   {
      onMouse = 1;
   }
   if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
   {
      onMouse = 2;
   }
}*/

void KeyboardFuncs(int key, int x, int y)
{
   if(key == GLUT_KEY_UP)
   {
      sumbuz[num] -= 0.1;
      if(cameraActive){
         testcmZ-=0.5;
         testlookZ-=0.5;
      }
   }
   if(key == GLUT_KEY_DOWN)
   {
      sumbuz[num] += 0.1;
      if(cameraActive){
         testcmZ+=0.5;
         testlookZ+=0.5;
      }
   }
   if(key == GLUT_KEY_RIGHT)
   {
      sumbux[num] += 0.1;
      if(cameraActive){
         testcmX+=0.5;
         testlookX+=0.5;
      }
```

```
        if(num == 100) angle += 10;
        //if(num == 4 || num == 2)
    }
    if(key == GLUT_KEY_LEFT)
    {
        if(cameraActive){
            testcmX-=0.5;
            testlookX-=0.5;
        }
        sumbux[num] -= 0.1;
        if(num == 100) angle -= 10;
        //if(num == 4 || num == 2)
    }
    if(key == GLUT_KEY_PAGE_DOWN && cameraActive)
    {
        testcmY-=0.5;
    }
    if(key == GLUT_KEY_PAGE_UP && cameraActive)
    {
        testcmY+=0.5;
    }
    if(key == GLUT_KEY_HOME && cameraActive)
    {
        testlookY+=0.5;
    }
    if(key == GLUT_KEY_END && cameraActive)
    {
        testlookY-=0.5;
    }
    if(key == GLUT_KEY_F1 && cameraActive)
    {
        testlookX-=0.5;
    }
    if(key == GLUT_KEY_F2 && cameraActive)
    {
        testlookX+=0.5;
    }
    //if(key == GLUT_KEY)
    //if(key == GLUT_KEY_PAGE_UP)
    //if(key == GLUT_KEY_PAGE_DOWN)
}

void keyboard(unsigned char key, int x, int y ){
    if(key=='1') num = 1;
    if(key=='2') num = 2;
    if(key=='3') num = 3;
    if(key=='4') num = 4;
    if(key=='5') num = 5;
    if(key=='6') num = 6;
    if(key=='7') num = 7;
    if(key=='8') num = 8;
    if(key=='c') {
        cameraActive = true;
        num = 0;
    }
    if(key=='v') cameraActive = false;
    if(key=='t'){
        if(ganti[num]==1) ganti[num] = 2;
        else if(ganti[num] == 0) ganti[num] = 1;
        else if(ganti[num] == 2) ganti[num] = 1;
    }
    if(key==',') putar[num] += 90;
    if(key=='.') putar[num] -= 90;
}

int main(int argc, char** argv)
{
    pts[0].x = 0;
    pts[0].y = -0.1;
    pts[0].z = -5;
    pts[1].x = -1;
    pts[1].y = -0.1;
    pts[1].z = 0;
    for(int i = 0; i < 101; i++)
    {
        sumbux[i] = 0.0;
        sumbuz[i] = 0.0;
        putar[i] = 0.0;
        ganti[i] = 0;
    }
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowPosition(300, 1);
    glutInitWindowSize(1000, 700);
    glutCreateWindow("Final Project");

    //int palm = glutCreateMenu(mymenu);
    //glutAddMenuEntry("Translasi", 1);
    //glutAddMenuEntry("Rotasi", 2);

    //int rumah = glutCreateMenu(mymenu);
    //glutAddMenuEntry("Translasi", 3);
    //glutAddMenuEntry("Rotasi", 4);

    //glutCreateMenu(mymenu);
    //    glutAddSubMenu("Palm", palm);
        //glutAddSubMenu("Rumah", rumah);
    //glutAttachMenu(GLUT_RIGHT_BUTTON);

    glutDisplayFunc(display);
    glutIdleFunc(display);
    //glutMouseFunc(mouse);
    glutSpecialFunc(KeyboardFuncs);
    glutKeyboardFunc(keyboard);
    glutReshapeFunc(reshape);
    Init();
    glutMainLoop();
}
```

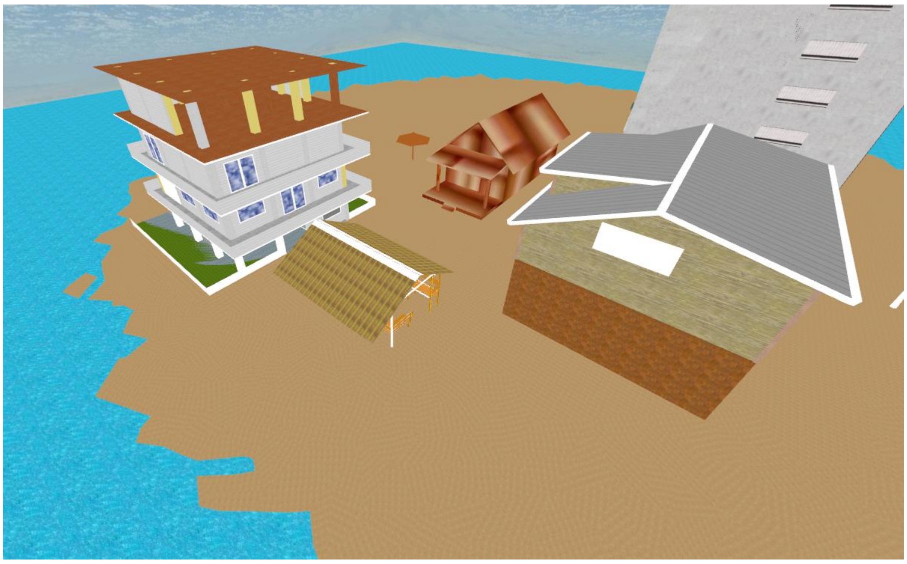
## IV. Screenhot Program



Sky view pada pantai



Tampak samping pantai



Translasi, Rotasi dan mengganti texture object