**LAPORAN TUGAS RANCANG GRAFIKA KOMPUTER**

**Perancangan dan Pembuatan Bangunan “GPIB Tamansari salatiga”**

**Berbasis Vektor dengan OpenGL**



Oleh:

1. Aldyth M. V Nahak (672017081
2. Stefanus Y. M Nange (672012109)
3. Irvons Anugrah (672017200)

Program Studi Teknik Informatika

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

2019

# PENDAHULUAN

Pada kota salatiga, Jawa Tengah, Memiliki beberapa gereja Kristen yang mempunyai bangunan – bangunan yang indah. kami tertarik dengan salah satu bangunan gereja di kota tersebut yaitu GPIB Tamansari Salatiga, yang mana kita tahu bahwa tempat tersebut sering digunakan oleh masyarakat untuk beribadah, bagi pemeluk Agama Kristen Protesten. Kabarnya gereja tersebut merupakan gereja tertua di daerah kota Salatiga, ini semakin membuat kami tertarik memilihnya sebagai bahan object Tugas Rancang kami. Gereja tesebut berlokasi di Jl. Jend. Sudirman No.1, Salatiga, Kec. Sidorejo, Kota Salatiga, Jawa Tengah 50711. Jika kita melihat gereja tersebut terlihat seperti bangunan pada zaman Belanda, karena memang bangunan tersebut merupakan salah satu peninggalan dari zaman Belanda, yang juga digunakan untuk beribadah.Ini semakin membuat kami tertantang untuk membuat bangunan tersebut sebagai object kami karena dengan berbagai cerita masa lalu dari gereja tersebut.

Tugas Rancang yang kami buat adalah sebuah miniatur atau duplikat bangunan tesebut dalam bentuk aplikasi program untuk menampilkan vektor – vektor dengan OpenGL dari object yang kami pilih, jadi disini kami membuat suatu gambar 3Dimensi dengan menggunakan bahasa pemrograman C++. Laporan ini kami susun sebagai pemenuhan Tugas Akhir ( Tugas Rancang ) mata kuliah Grafika Komputer ( Grafkom ) di semester 1 tahun ajaran 2019 / 2020 yang diampu oleh dosen kami Bpk. Evangs Mailoa. Yang telah mengajarkan kami cara pembuatan bangunan 3D atau bidang 2D melalui bahasa pemrograman C++ berbasis Vektor dengan Open GL. Dalam perancangan dan pembuatan ini bisa menggunakan banyak aplikasi seperti Dev C++, Visual Studio, code blocks dan lain-lain. Disini kami menggunakan aplikasi code blocks.

# TINJAUAN PUSTAKA

Pada artikel lain seperti pada laporan kelompok lain mungkin juga membahas pembuatan suatu bangunan tetapi pada dasarnya cara membuatnya sama-sama memakai bahasa pemrograman C++ dan OpenGL. Dan perlu diketahui OpenGL merupakan aplikasi software berisi package ( paket ) tentang kode programan yang digunakan unutk mendesain dan menentukan titik-titik vektor. Seperti untuk menentukan bidang segi empat itu deperlukan empat titik, nah titik-titik tersebut mempunyai syntax program yang sama tetapi letak tiap titik tersebut berbeda, contoh kode program:

void balok(float x, float y, float z, float panjang, float tinggi, float lebar){

glBegin(GL\_QUADS);  memulai vektor dengan QUADS yaitu sebuah bentuk yang memiliki 4 sisi

glColor3f(0.942f, 0.9f, 0.55f); inisialisasi warna pada bentuk yaitu warna kuning Khaki

glVertex3f(x,y,z);  titik pertama dengan koordinat( x,y,z ).

glVertex3f(x,y+tinggi,z);  titik kedua dengan koordinat (x,y+tinggi,z ) dimana titik awal y ditambahkan dengan jarak atau tinggi dari titik pertama.

glVertex3f(x+panjang,y+tinggi,z);  titik ketiga dengan koordinat (x+panjang,y+tinggi,z ) dimana titik x ditambahkan dengan jarak Panjang dari titik kedua.

glVertex3f(x+panjang,y,z);  titik keempat dengan koordinat (x+panjang,y,z )

glEnd()};  akhir vektor.

Logika mengenai penempatan titik-titik vector sehingga membentuk suatu bentuk atau bidang dalam pemrograman dapat berbeda pemahaman tergantung dari kemampuan kita untuk membayangkan bentuk 3 dimensi. Dalam hal ini kami membuat suatu fungsi yang dapat membentuk titik-titik vector sesuai dengan kordinat titik awal dan titik akhir (titik awal +jarak) sehingga menggambar suatu bentuk dapat lebih mudah.

# PEMBAHASAN

Hasil project :



**Penjelasan :**

#include <windows.h> //inisialisasi library yang digunakan dalam proses pembuatan vector mengunakan glut

#include <stdlib.h>

#include <GL/glut.h>

#include <math.h> // library matematika yang dapat mengerjakan berbagai operasi perhitungan dalam matematika

#include "imageloader.h" //library yang digunakan untuk menampilkan texture atau gambar

Selanjutnya, dibawah ini merupakan nama fungsi method – method yang kami gunakan :

1. gereja(void)

Contoh coding :

//Fungsi display kami menamakannya void gereja

void gereja(void)

{//mengatur depth sehingga gambar bisa terlihat transparan

if (is\_depth)

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

else

glClear(GL\_COLOR\_BUFFER\_BIT);

glEnable(GL\_BLEND);

glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);

glLightModeli(GL\_LIGHT\_MODEL\_LOCAL\_VIEWER,GL\_TRUE);

//Fungsi untuk mengaktifkan dan mengatur lighting dalam tampilan sehingga dapat menampilkan kesan bahwa objek disinari oleh matahari

glEnable(GL\_LIGHTING);

glEnable(GL\_LIGHT1);

//set cahaya

GLfloat qaAmbientLight[] = {0.2,0.2,0.2,1.0};

GLfloat qaDiffuseLight[] = {0.2,0.2,0.2,1.0};

GLfloat qaSpecularLight[] = {0.2,0.2,0.2,1.0};

glLightfv(GL\_LIGHT1, GL\_AMBIENT,qaAmbientLight);

glLightfv(GL\_LIGHT1,GL\_DIFFUSE,qaDiffuseLight);

glLightfv(GL\_LIGHT1,GL\_SPECULAR,qaSpecularLight);

//posisi cahaya

//GLfloat cahaya[] = {25,50,100,1};

GLfloat cahaya[] = {-25,50,100,1};

glLightfv(GL\_LIGHT1,GL\_POSITION,cahaya);

glEnable(GL\_COLOR\_MATERIAL);

glShadeModel(GL\_SMOOTH);

glMatrixMode(GL\_MODELVIEW);

//Pemanggilan fungsi-fungsi yang digunakan dalam coding

///////////////////////balok segala sisi/////////////////////////////////////

glColor3f(0.5f, 0.5f, 0.5f);

balok(-11.0, -10.0, 16.0, 2.0, 40.5, 2.0);//dpn kiri

balok(9.0, -10.0, 16.0, 2.0, 40.5, 2.0);//dpn kanan

balok(-11.0, -10.0, 0.5, 0.5, 35.0, 0.5);//dpn2 kiri

balok(10.0, -10.0, 1.0, 0.5, 35.0, 0.5);//dpn2 kanan

les(-40.0, 33.0, 0.1, 2.0, 2.0, 0.1);

glColor3f(0.6f, 0.6f, 0.6f);

balok(-41.0, 33.0, 1.0, 82.0, 2.0, 5.0);//atas

balok(-41.0, 33.0, -36.0, 82.0, 2.0, 5.0);//atas

////////////////////////Bangunan Gereja///////////////////////////////////////

glColor3f(1,1,1);

atapkk(-39.9,35.0,0.9,79.8,18.0,42.0);

glColor3f(0.942f, 0.9f, 0.55f);

atapkk(-40.0,35.0,0.9,80.0,18.0,42.0);

gedungkecil(-10.0, -10.0, 15.0, 20.0, 35.0, 15.0,0);//tembok luar

gedungutama(-40.0, -10.0, 0.0, 80.0, 45.0, 40.0,0);//tembok luar

glColor3f(0.5f, 0.6f, 0.7f);

kap(33.2,-20.0,18.0,360.0,-39.8,79.0);//kap setengah lingkaran

glColor4f(0.0f, 0.0f, 1.0f,0.7);

jendelaL(37,-20.0,4.0,180.0,39.8,0.1,3.14);//jendela setengah lingkaran

jendelaL(37,-20.0,4.0,180.0,40,0.1,3.14);

jendelaL(37,-20.0,4.0,180.0,-40.1,0.1,3.14);

jendelaL(37,-20.0,4.0,180.0,-39.8,0.1,3.14);

////////jendela/////////////

jendela(-17.0,2.0,0.1,5.0,18.0,1.0);//depan

jendela(-23.0,2.0,0.1,5.0,18.0,1.0);

jendela(11.5,2.0,0.1,5.0,18.0,1.0);

jendela(17.5,2.0,0.1,5.0,18.0,1.0);

jendela(-17.0,2.0,-40.2,5.0,18.0,1.0);//belakang

jendela(-23.0,2.0,-40.2,5.0,18.0,1.0);

jendela(11.5,2.0,-40.2,5.0,18.0,1.0);

jendela(17.5,2.0,-40.2,5.0,18.0,1.0);

jendela(-3,0.0,-80.2,6.0,14.0,1.0);//jendela gedung belakang

// jendela(-3,0.0,-79.9,6.0,14.0,0.0);

jendelabelakang(-3.0,0.0,-79.9,6.0,14.0);

glColor3f(0.9f, 0.9f, 0.9f);

jendelasamping(-10.1,2.0,10.0,3.0,15.0,0);//kiri luar

jendelasamping(-10.1,2.0,6.0,3.0,15.0,0);

jendelasamping(10.1,2.0,10.0,3.0,15.0,0);//kanan luar

jendelasamping(10.1,2.0,6.0,3.0,15.0,0);

jendelasamping(-6.8,2.0,10.0,3.0,15.0,0);//kiri dalam

jendelasamping(-6.8,2.0,6.0,3.0,15.0,0);

jendelasamping(6.8,2.0,10.0,3.0,15.0,0);//kanan dalam

jendelasamping(6.8,2.0,6.0,3.0,15.0,0);

jendelasamping(-3.1,37.0,14.5,2.0,10.0,1);//kiri lonceng

jendelasamping(-3.1,37.0,11.5,2.0,10.0,1);

jendelasamping(3.1,37.0,14.5,2.0,10.0,1);//kanan lonceng

jendelasamping(3.1,37.0,11.5,2.0,10.0,1);

jendela(-2.5,37.0,15.1,2.0,7.0,1.0);//lonceng//depan

jendela(0.5,37.0,15.1,2.0,7.0,1.0);

jendela(-2.5,37.0,8.9,2.0,7.0,1.0);//belakang

jendela(0.5,37.0,8.9,2.0,7.0,1.0);

//////////////////Dalam Gereja///////////////////////////////

glColor3f(0.5,0.53,0.6);

tangga(30.0,15.0,-0.2,9.8,7.0,39.7);

kursi(2.0,-4.0,-2.0,2.0,5.0,7.0,39.0);//kursi jemaat

kursi(2.0,-4.0,-14.0,2.0,5.0,7.0,39.0);

kursi(2.0,-4.0,-26.0,2.0,5.0,7.0,30.0);

kursi2(-31.0,-4.0,-0.5,10.0,8.0,3.0,7.8);//kursi majelis kiri

kursi2(-19.0,-4.0,-0.5,10.0,8.0,3.0,7.8);

kursi2(-31.0,-4.0,-5.5,10.0,5.0,3.0,7.8);

kursi2(-19.0,-4.0,-5.5,10.0,5.0,3.0,7.8);

kursi2(-31.0,-4.0,-39.8,10.0,5.0,3.0,7.8);//kursi majelis kanan

kursi2(-19.0,-4.0,-39.8,10.0,5.0,3.0,7.8);

kursi2(-31.0,-4.0,-34.0,10.0,5.0,3.0,7.8);//kursi majelis kanan

kursi2(-19.0,-4.0,-34.0,10.0,5.0,3.0,7.8);

//mimbar//

glColor3f(0.1,0.05,0.02);

balok(-31.0,-9.8,-16.0,5.0,1.0,8.0);

balok(-29.0,-7.8,-17.2,3.0,1.0,5.5);

balok(-28.5,-7.8,-16.0,2.5,12.0,0.5);//

balok(-28.5,-7.8,-23.5,2.5,12.0,0.5);

balok(-26.5,-7.8,-16.0,0.5,12.0,8.0);

balok(-29.0,-8.8,-16.0,3.0,1.0,8.0);

//Salib

glColor3f(1,1,1);//lonceng

balok(-2.0,45.0,16.5,4.0,1.8,1.0);

balok(-0.7,37.0,16.5,1.4,12.0,1.0);

glColor4f(1.0,0.0,0.0,0.5);

balok(-1.5,45.2,16.6,3.0,1.1,0.5);

balok(-0.5,37.5,16.6,1.0,11.0,0.5);

glColor3f(0,0,0);//dalam gereja

balok(-39.8,12.5,-19.99,0.1,3.0,0.1);

balok(-39.8,15.0,-19.5,0.1,0.1,1.0);

//////////////daerah pintu,menara,dll////////////////

lonceng2(-3.0,32.0,9.0,6.0,22.0,-5.9);

glColor3f(0.5f, 0.5f, 0.5f);

glColor3f(0.647f, 0.165f, 0.165f);

kapdpn(0.0,18.0,7.0,100.0,15.1,6.9,3.14);//kap pintu

glColor3f(0.8,0.8,0.8);

kapdpn(0.0,17.8,7.0,100.0,15.1,6.9,3.14);

glColor3f(0.5f, 0.5f, 0.5f);

balok(9.0, 25.0, 16.0, 3.0, 3.0, 16.0);

balok(-12.0, 25.0, 16.0, 3.0, 3.0, 16.0);

dpn();//zigzag

balokL(6.0,16.0,17.0,4.0);

glColor3f(0.55f, 0.55f, 0.55f);

balok(6.0, 16.0, 22.0, 3.0, 3.0, 7.0);//dibwh stgh lingkaran

balok(-9.0, 16.0, 22.0, 3.0, 3.0, 7.0);

kapdpn(0.0,17.9,7.0,100.0,22.0,0.2,3.14);

kapdpn(0.0,18.0,5.5,100.0,15.5,0.4,3.14);//sekeliling jendela

kapdpn(0.0,18.0,6.0,100.0,15.5,0.4,3.14);//sekeliling jendela

glBegin(GL\_LINES);//garis samping pintu

glVertex3f(5.5,-9.8,15.5);

glVertex3f(5.5,17.5,15.5);

glVertex3f(-5.5,-9.8,15.5);

glVertex3f(-5.5,17.5,15.5);

glEnd();

balok(6.0, 16.0, 22.0, 3.0, 3.0, 7.0);//dibwh stgh lingkaran

balok(-9.0, 16.0, 22.0, 3.0, 3.0, 7.0);

glColor4f(0.7,0.7,0.7,0.5);

lingkaran(0.0,18.0,5.0,100.0,15.1,0.0,3.14);//jendela

lingkaran(0.0,18.0,5.0,100.0,14.8,0.0,3.14);//jendela

//pintu

glColor3f(0.5,0.27,0.08);

balok(-5.0,-10.0,15.5,0.5,27.0,1.0);

balok(4.5,-10.0,15.5,0.5,27.0,1.0);

balok(-5.0,16.0,15.5,10.0,1.0,1.0);

//glColor3f(1.0,1.0,1.0);

glColor3f(0.5,0.27,0.08);

//pemanggilan texture atau gambar pada objek

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdpkanan);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

pintu(4.0,-10.0,15.0,3.0,26.0,5.0);glColor3f(0.5,0.27,0.08);

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdpkiri);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

pintu(-4.0,-10.0,15.0,-3.0,26.0,5.0);

glColor3f(0.6,0.6,0.6);

glBegin(GL\_QUADS);

glVertex3f(-12.0, -9.9, -45.5);//belakang

glVertex3f(-12.0, 17.8, -45.5);

glVertex3f(-12.0, 17.8, -54.5);

glVertex3f(-12.0, -9.9, -54.5);

glEnd();

//////////////////gedung gereja + texture///////////////////////////////

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdkeramik2);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

glColor3f(1.0f, 1.0f, 1.0f);

gedungkecil(-9.9, -9.9, 14.9, 19.8, 35.1, 14.9,1);//tembok dalam

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdkeramik);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

gedungutama(-39.9, -9.9, -0.1, 79.8, 45.1, 39.8,1);//tembok dalam

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdkeramik3);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

balok(-10.0,-10.1,24.0,20.0,0.1,9.0);//keramik luar

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIddasar);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

balok(-50.0,-10.5,70.0,100.0,0.1,200.0);//dasar

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdgenteng);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

atap2(-40.0,35.0,0.9,80.0,18.0,42.0);//atap gedung gereja

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdatap);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

lonceng(-3.0,32.0,9.0,6.0,22.0,-5.9);//atap menara

//JENDELA //texture start

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdLF1);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(-17.0,2.0,-0.2,5.0,18.0,0.0);//dalam depan//jendelaLF1

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdLF2);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(-23.0,2.0,-0.2,5.0,18.0,0.0);//jendelaLF2

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdLB2);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(11.5,2.0,-0.2,5.0,18.0,0.0);//jendelaLB2

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdLB1);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(17.5,2.0,-0.2,5.0,18.0,0.0);//jendelaLB1

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdRF1);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(-17.0,2.0,-39.8,5.0,18.0,0.0);//dalam belakang//jendelaRF1

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdRF2);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(-23.0,2.0,-39.8,5.0,18.0,0.0);//jendelaRF2

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdRB2);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(11.5,2.0,-39.8,5.0,18.0,0.0);//jendelaRB2

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdRB1);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

jendela(17.5,2.0,-39.8,5.0,18.0,0.0);//jendelaRB1

//textureEND

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdlingkaran);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

glBegin(GL\_QUADS);

glTexCoord2f(0.0f, 0.0f);

glVertex3f(-3, 18, 15.3);

glTexCoord2f(1.0f, 0.0f);

glVertex3f(-3, 24.8, 15.3);

glTexCoord2f(1.0f, 1.0f);

glVertex3f(3, 24.8, 15.3);

glTexCoord2f(0.0f, 1.0f);

glVertex3f(3 , 18, 15.3);

glTexCoord2f(0.0f, 0.0f);/////////

glVertex3f(-3, 18, 14.7);

glTexCoord2f(1.0f, 0.0f);

glVertex3f(-3, 24.8, 14.7);

glTexCoord2f(1.0f, 1.0f);

glVertex3f(3, 24.8, 14.7);

glTexCoord2f(0.0f, 1.0f);

glVertex3f(3 , 18, 14.7);

glEnd();

glEnable(GL\_TEXTURE\_2D);

glBindTexture(GL\_TEXTURE\_2D, \_textureIdkacadalam);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MIN\_FILTER, GL\_LINEAR);

glTexParameteri(GL\_TEXTURE\_2D, GL\_TEXTURE\_MAG\_FILTER, GL\_LINEAR);

glBegin(GL\_QUADS);

glColor4f(1,1,1,0.7);

glTexCoord2f(0.0f, 0.0f);

glVertex3f(-7, 15, 0.0);

glTexCoord2f(1.0f, 0.0f);

glVertex3f(-7, 25.0, 0.0);

glTexCoord2f(1.0f, 1.0f);

glVertex3f(7, 25.0, 0);

glTexCoord2f(0.0f, 1.0f);

glVertex3f(7 , 15, 0.0);

glEnd();

glDisable(GL\_TEXTURE\_2D);

//fungsi untuk menampilkan gambar

glutSwapBuffers();

}

Contoh coding yang tidak menggunakan texture:

void kap(float yp, float zp, float r, float n,float x,float panjang){

//’yp’ adalah titik y letak kap setengah lingkaran di tempatkan, ‘zp’ merupakan titik tengah dari lingkaran, ‘r’ merupakan jari-jari lingkaran, ‘n’ merupakan jumlah sudut pada lingkaran, ‘x’ merupakan letak titik pertama lingkaran terhadap sumbu x dan Panjang merupakan jarak ata ukuran dari objek.

float a,z,y;

glLineWidth(50.0);//menentukan ukuran garis

glBegin(GL\_LINES);// menggunakan lines atau garis supaya bentuknya menjadi setengah lingkaran

a=3.14/n; //3.14 merupakan nilai pi untuk setengah lingkaran

for (int i=0; i<n; i++){

z = zp+r \* cos(i\*a);

y = yp+(r/1.75) \* sin(i\*a);

glVertex3f(x,y,z);

glVertex3f(x+panjang,y,z);

}

glEnd();

}

Contoh coding yang menggunakan texture:

//contoh coding untuk membuat kaca jendela dalam gereja

glBegin(GL\_QUADS);

glColor3f(1,1,1);

glTexCoord2f(0.0,0.0);//texture

glVertex3f(x+0.5, y+0.5, z+i);

glTexCoord2f(1.0,0.0);

glVertex3f(x+0.5, y + tinggi-0.5, z+i);

glTexCoord2f(1.0,1.0);

glVertex3f(x + panjang-0.5, y + tinggi-0.5, z+i);

glTexCoord2f(0.0,1.0);

glVertex3f(x + panjang-0.5, y+0.5, z+i);

glEnd();

contoh texture yang dimunculkan:



2. gambar ()

Berisi fungsi untuk memanggil gambar atau texture.

Contoh coding :

void gambar(void)

{//fungsi untuk mengaktifkan depth

glEnable(GL\_DEPTH\_TEST);

is\_depth = 1;

glEnable(GL\_COLOR\_MATERIAL);

glMatrixMode(GL\_MODELVIEW);

glEnable(GL\_NORMALIZE);

glEnable(GL\_LINE\_SMOOTH);

glShadeModel(GL\_SMOOTH);

//inisialisasi ukuran titik dan garis

//pemanggilan gambar dengan format file BMP

Image\* image = loadBMP("keramik.bmp");

\_textureIdkeramik = loadTexture(image);

Image\* image2 = loadBMP("keramik2.bmp");

\_textureIdkeramik2 = loadTexture(image2);

Image\* image3 = loadBMP("keramik3.bmp");

\_textureIdkeramik3 = loadTexture(image3);

Image\* image4 = loadBMP("dasar.bmp");

\_textureIddasar = loadTexture(image4);

Image\* image5 = loadBMP("genteng.bmp");

\_textureIdgenteng= loadTexture(image5);

Image\* image6 = loadBMP("atap.bmp");

\_textureIdatap = loadTexture(image6);

Image\* image7 = loadBMP("pintukanan.bmp");

\_textureIdpkanan = loadTexture(image7);

Image\* image8 = loadBMP("pintukiri.bmp");

\_textureIdpkiri = loadTexture(image8);

Image\* image9 = loadBMP("jendelaLB1.bmp");

\_textureIdLB1 = loadTexture(image9);

Image\* image10 = loadBMP("jendelaLB2.bmp");

\_textureIdLB2 = loadTexture(image10);

Image\* image11 = loadBMP("jendelaLF1.bmp");

\_textureIdLF1 = loadTexture(image11);

Image\* image12 = loadBMP("jendelaLF2.bmp");

\_textureIdLF2 = loadTexture(image12);

Image\* image13 = loadBMP("jendelaRB1.bmp");

\_textureIdRB1 = loadTexture(image13);

Image\* image14 = loadBMP("jendelaRB2.bmp");

\_textureIdRB2 = loadTexture(image14);

Image\* image15 = loadBMP("jendelaRF1.bmp");

\_textureIdRF1 = loadTexture(image15);

Image\* image16 = loadBMP("jendelaRF2.bmp");

\_textureIdRF2 = loadTexture(image16);

Image\* image17 = loadBMP("lingkaran.bmp");

\_textureIdlingkaran = loadTexture(image17);

Image\* image18 = loadBMP("kacadalam.bmp");

\_textureIdkacadalam = loadTexture(image18);

delete image;delete image2;delete image3;delete image4;delete image5;delete image6;delete image7;delete image8;delete image9;delete image10;delete image11;delete image12;delete image13;delete image14;delete image15;delete image16;delete image17;delete image18;

}

3. Keyboard()

Membuat fungsi kontrol pada keyboard agar bisa melihat object dari berbagai sudut.

Contoh coding :

// ini nama method keyboard, di sini memberikan fungsi keyboar special yang telah di sediakan oleh aplikasi yang kita gunakan.

void keyboard(int key, int x, int y)

{ switch(key)

{ // ini untuk tombol keyboard Up, berfungsi untuk memindahkan object berdasarkan kordinat z.

case GLUT\_KEY\_UP:

glTranslatef(0.0, 0.0, 3.0);

;break;

case GLUT\_KEY\_RIGHT:

glTranslatef(3.0, 0.0, 0.0); break; case GLUT\_KEY\_DOWN:

glTranslatef(0.0, 0.0, -3.0); break; case GLUT\_KEY\_LEFT: glTranslatef(-3.0, 0.0, 0.0); break;

}

gereja ();

}

void keyboard2(unsigned char key,int x,int y)

{

switch (key)

{

//fungsi untuk memindahkan sudut pandang kita ke titik yang telah ditentukan dalam hal ini jika dikaitkan dengan hasil project maka akan membuat kita seolah-olah berada dalam Gedung gereja.

case 'i':case 'I':

glMatrixMode(GL\_PROJECTION);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluPerspective(50.0, 1280/720, 5.0, 500.0);

glTranslatef(0.0, -10.0, -10.0);

break;

//fungsi untuk mengembalikan sudut pandang ke semula

case '0':

glMatrixMode(GL\_PROJECTION);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

gluPerspective(50.0, 1280/720, 5.0, 500.0);

glTranslatef(0.0, -10.0, -150.0);

// ini untuk tombol keyboard angka 7, berfungsi untuk memindahkan object berdasarkan kordinat y.

case '7': glTranslatef(0.0,3.0,0.0); break;

// ini untuk tombol keyboard angka 9, berfungsi untuk memindahkan object berdasarkan kordinat -y.

case '9': glTranslatef(0.0,-3.0, 0.0); break;

// ini untuk tombol keyboard angka 2, berfungsi untuk memutar object berdasarkan kordinat x dan y.

case '2': glRotatef(2.0,1.0,0.0,0.0); break; case '8': glRotatef(-2.0,1.0,0.0,0.0); break; case '6': glRotatef(2.0,0.0,1.0,0.0); break; case '4': glRotatef(-2.0,0.0,1.0,0.0); break; case '1': glRotatef(2.0,0.0,0.0,1.0); break;

// ini untuk tombol keyboard angka 3, berfungsi untuk memutar object berdasarkan sumbu tengahnya z dan memutar ke kanan.

case '3': glRotatef(-2.0,0.0,0.0,1.0); break; case '5':

if (is\_depth)

{ is\_depth = 0;

glDisable(GL\_DEPTH\_TEST);

} else { is\_depth = 1;

glEnable(GL\_DEPTH\_TEST);

}

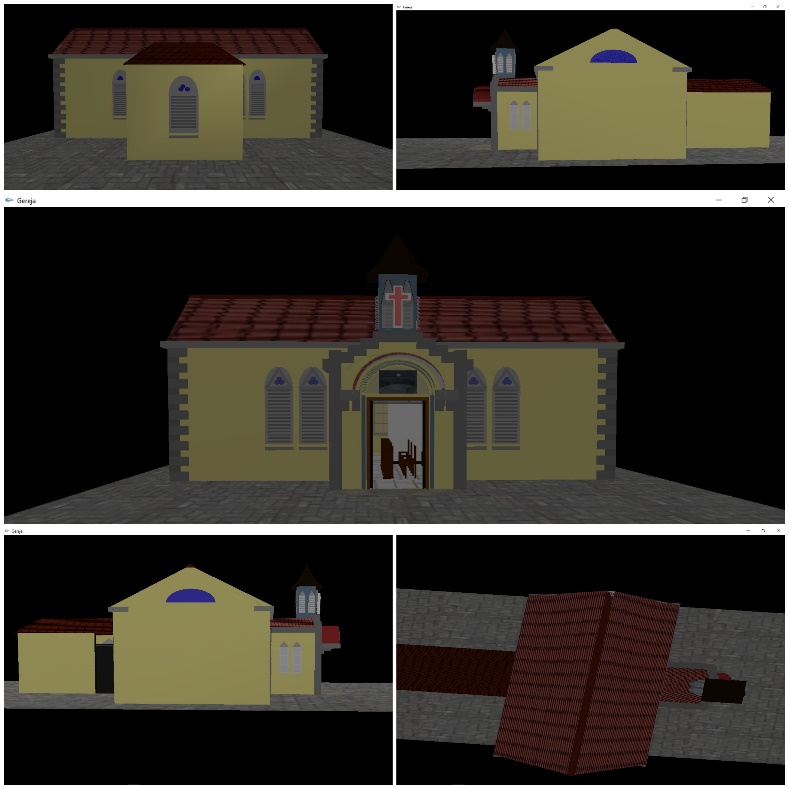
}

gereja();

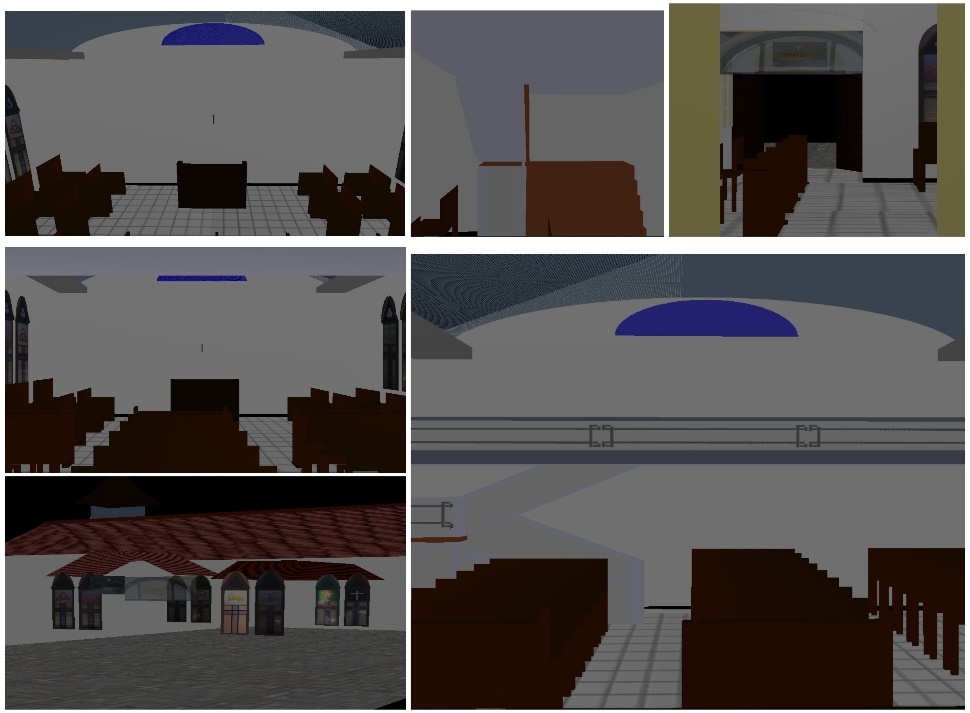
}

Contoh gambar object dilihat beberapa sisi, dan transparent :

Tampak Luar:



Tampak Dalam:



4. Scale()

Fungsi untuk menentukan ukuran objek yang ditampilkan pada layar.

Contoh coding :

/ resize dengan nama method ukuran dan mempunyai paramater integer lebar ( lebar window ) dan integer tinggi ( tinggi widow ).

void scale(int w, int h){

// ini merupkan suatu kondisi dimana, jika tinggi bernilai 0 maka tinggi bernilai 1. if (h == 0) h= 1;

// Ini untuk memposisikan object yang kita buat tepat ditengah, dalam artian koordinat x,y,z : (0,0,0);

glMatrixMode(GL\_PROJECTION);

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity();

// Ini untuk mengatur letak object gluPerspective(50.0, w / h, 5.0, 500.0); glTranslatef(0.0, -10.0, -150.0);

}

5. Main()

Fungsi utama untuk menampilkan semua fungsi dan objek glut.

Contoh coding :

int main (int argc, char \*\*argv)

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(1366,768);//inisialisasi ukuran windows

glutInitWindowPosition(40,0);

glutCreateWindow("Gereja");//memberi nama pada open windows

gambar();//pemanggilan fungsi untuk mengambil texture

glutReshapeFunc(scale);//pemanggilan fungsi ukuran atau scala objek

glutDisplayFunc(gereja);//pemanggilan objek

glutSpecialFunc(keyboard);//pemanggilan funsi kontrol

glutKeyboardFunc(keyboard2);

glutMainLoop();

return 0;

}

Berbagai kesulitan yang dialami saat mengerjakan project ini:

1. Kedetailan object yang rumit.

Seperti yang kita tahu bahwa object yang kita teliti memiliki tingkat kerumitan yang menurut kami cukup sulit karena kedetailan dari gedung tersebut, seperti ukirukurannya, detail teksturnya, dan juga kita tidak bisa melihat detailnya dari atas. Untuk itu kami hanya bisa membuat sebisanya agar terlihat mirip seperti aslinya.

1. Tekstur object.

Untuk membuat objek semirip mungkin dengan aslinya maka kami harus mencari jadwal yang tepat supaya bisa memotret teksture yang ada pada gereja secara langsung akan tetapi gereja hanya terbuka pada hari sabtu dan minggu.

# KESIMPULAN

Dengan menggunakan Glut dan openGL maka kita dapat membuat project berbasis grafik menggunakan Bahasa pemrograman sederhana yaitu C++, asalkan kita tahu mengaplikasikan coding dan logika mengenai bentuk 3 dimensi harus bisa dibayangkan dan dipahami dengan baik sehingga tidak terjadi kebingungan dalam menentukan titik.

# SUMBER PUSTAKA

Grafika Komputer, Evangs Mailoa (n.d.), diambil 26 November 2013, dari

<http://evangsmailoa.wordpress.com/materi-kuliah/grafika/>

Belajar grafika komputer dengan opengl, Badcoding (2011), diambil 1 Desember 2013, dari <http://badcoding.wordpress.com/2011/10/19/belajar-grafika-komputer-dengan-opengl/>

Laporan Perancangan dan pembuatan bangunan”Gereja Santa Paulus Miki Salatiga” berbasis vector dengan openGL(2013), David Karismata Tion & Rino Tri Aji Pamungkas ,diambil 29 Juli 2019.

***Tuhan Yesus Memberkati***