**TECHNICAL REPORT**

**GRAFIKA KOMPUTER**

**MODUL 1**



**Disusun Oleh :**

TGL. PRAKTIKUM : Jum’at, 2 April 2021

NAMA : Achmad Farid Alfa Waid

NRP : 190411100073

KELOMPOK : 2

DOSEN : Ach. Khozaimi, S.Kom., M.Kom

TELAH DISETUJUI TANGGAL :

...........................................

ASISTEN PRAKTIKUM

Rizky Alfriansyah

(180411100068)

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**BAB I**

**TUJUAN DAN DASAR TEORI**

1. **TUJUAN**

Mahasiswa dapat mengetahui pengertian OpenGL, alur pembuatan program grafis berbasis OpenGL menggunakan CodeBlocks dan mengetahui dasar Application Programming Interface.

1. **DASAR TEORI**

Grafika Komputer dewasa ini telah melesat sangat jauh dibandingkan pada masa awal kemunculannya. Cakupannya telah meluas ke berbagai bidang: kedokteran, sains, teknik, bisnis, industri, seni, hiburan, iklan, dan lain-lain. Salah satu tools/library pembuatan aplikasi grafik adalah OpenGL (Open Graphics Library). OpenGL adalah suatu standar grafik yang menyediakan fungsi-fungsi low-level untuk pembuatan berbagai gambar pada komputer. Sebagai API (Application Programming Interface), OpenGL bersifat platformindependent/tidak tergantung pada piranti dan platform yang digunakan. Sehingga, adalah hal yang wajar jika aplikasi OpenGL berjalan pada sistem operasi Windows, UNIX, Mac, Android, dll. OpenGL pada awalnya didesain untuk digunakan oleh bahasa pemrograman C/C++, namun dalam perkembangannya OpenGL dapat juga digunakan oleh bahasa pemrograman yang lain seperti Java, Tcl, Ada, Visual Basic, Delphi, maupun Fortran.

**BAB II**

**PEMBAHASAN**

1. **SOAL**
2. Lakukan percobaan sesuai dengan perintah di atas.
3. Buat Pelangi horizontal yang terdiri dari 7 warna berbeda menggunakan 7 rectangle (GL\_POLYGON).
4. Buat Pelangi vertical yang terdiri dari 7 warna berbeda menggunakan 7 rectangle (GL\_POLYGON).
5. Ulang soal nomor 2 dan 3 menggunakan glRect().
6. **JAWABAN**
7. **Code Program**
8. Nomor 1

/\*

\* GLUT Shapes Demo

\*

\* Written by Nigel Stewart November 2003

\*

\* This program is test harness for the sphere, cone

\* and torus shapes in GLUT.

\*

\* Spinning wireframe and smooth shaded shapes are

\* displayed until the ESC or q key is pressed. The

\* number of geometry stacks and slices can be adjusted

\* using the + and - keys.

\*/

#include <windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

void display(void)

{

/\* bersihkan layar dari titik pixel yang masih ada \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glColor3f (1.0, 1.0, 0.0);

/\* gambar 5 titik di layar \*/

//glBegin(GL\_POINTS);

//glBegin(GL\_LINE\_STRIP);

//glBegin(GL\_LINE\_LOOP);

//glBegin(GL\_LINES);

//glBegin(GL\_TRIANGLES);

//glBegin(GL\_TRIANGLE\_FAN);

//glBegin(GL\_TRIANGLE\_STRIP);

//glBegin(GL\_QUADS);

//glBegin(GL\_QUAD\_STRIP);

glBegin(GL\_POLYGON);

glVertex3f (0.0, 0.0, 0.0);

glColor3f (1.0, 0.0, 0.0);

glVertex3f (0.0, 0.8, 0.0);

glColor3f (0.0, 1.0, 0.0);

glVertex3f (0.8, 0.0, 0.0);

glColor3f (0.0, 0.0, 1.0);

glVertex3f (0.0, -0.8, 0.0);

glColor3f (1.0, 1.0, 1.0);

glVertex3f (-0.8, 0.0, 0.0);

glColor3f (0.0, 0.0, 0.0);

glEnd();

glFlush ();

}

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glPointSize(100.0);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("Primitif");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

1. Nomor 2

/\*

\* GLUT Shapes Demo

\*

\* Written by Nigel Stewart November 2003

\*

\* This program is test harness for the sphere, cone

\* and torus shapes in GLUT.

\*

\* Spinning wireframe and smooth shaded shapes are

\* displayed until the ESC or q key is pressed. The

\* number of geometry stacks and slices can be adjusted

\* using the + and - keys.

\*/

#include <windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

void display(void)

{

/\* bersihkan layar dari titik pixel yang masih ada \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

// warna merah

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3f (-1.0, 0.7, 0.0);

glVertex3f (1.0, 0.7, 0.0);

glVertex3f (1.0, 1.0, 0.0);

glVertex3f (-1.0, 1.0, 0.0);

glEnd();

// warna hijau

glBegin(GL\_POLYGON);

glColor3f (0.0, 1.0, 0.0);

glVertex3f (-1.0, 0.4, 0.0);

glVertex3f (1.0, 0.4, 0.0);

glVertex3f (1.0, 0.7, 0.0);

glVertex3f (-1.0, 0.7, 0.0);

glEnd();

// warna biru

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 1.0);

glVertex3f (-1.0, 0.1, 0.0);

glVertex3f (1.0, 0.1, 0.0);

glVertex3f (1.0, 0.4, 0.0);

glVertex3f (-1.0, 0.4, 0.0);

glEnd();

// warna hitam

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 0.0);

glVertex3f (-1.0, -0.2, 0.0);

glVertex3f (1.0, -0.2, 0.0);

glVertex3f (1.0, 0.1, 0.0);

glVertex3f (-1.0, 0.1, 0.0);

glEnd();

// warna putih

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 1.0);

glVertex3f (-1.0, -0.5, 0.0);

glVertex3f (1.0, -0.5, 0.0);

glVertex3f (1.0, -0.2, 0.0);

glVertex3f (-1.0, -0.2, 0.0);

glEnd();

// warna kuning

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 0.0);

glVertex3f (-1.0, -0.8, 0.0);

glVertex3f (1.0, -0.8, 0.0);

glVertex3f (1.0, -0.5, 0.0);

glVertex3f (-1.0, -0.5, 0.0);

glEnd();

// warna ungu

glBegin(GL\_POLYGON);

glColor3f (1.0, 0.0, 1.0);

glVertex3f (-1.0, -1.1, 0.0);

glVertex3f (1.0, -1.1, 0.0);

glVertex3f (1.0, -0.8, 0.0);

glVertex3f (-1.0, -0.8, 0.0);

glEnd();

glFlush ();

}

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 2");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

1. Nomor 3

/\*

\* GLUT Shapes Demo

\*

\* Written by Nigel Stewart November 2003

\*

\* This program is test harness for the sphere, cone

\* and torus shapes in GLUT.

\*

\* Spinning wireframe and smooth shaded shapes are

\* displayed until the ESC or q key is pressed. The

\* number of geometry stacks and slices can be adjusted

\* using the + and - keys.

\*/

#include <windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

void display(void)

{

/\* bersihkan layar dari titik pixel yang masih ada \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

// warna merah

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3f (0.7, -1.0, 0.0);

glVertex3f (0.7, 1.0, 0.0);

glVertex3f (1.0, 1.0, 0.0);

glVertex3f (1.0, -1.0, 0.0);

glEnd();

// warna hijau

glBegin(GL\_POLYGON);

glColor3f (0.0, 1.0, 0.0);

glVertex3f (0.4, -1.0, 0.0);

glVertex3f (0.4, 1.0, 0.0);

glVertex3f (0.7, 1.0, 0.0);

glVertex3f (0.7, -1.0, 0.0);

glEnd();

// warna biru

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 1.0);

glVertex3f ( 0.1, -1.0, 0.0);

glVertex3f ( 0.1, 1.0, 0.0);

glVertex3f ( 0.4, 1.0, 0.0);

glVertex3f ( 0.4, -1.0, 0.0);

glEnd();

// warna hitam

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 0.0);

glVertex3f ( -0.2, -1.0, 0.0);

glVertex3f ( -0.2, 1.0, 0.0);

glVertex3f ( 0.1, 1.0, 0.0);

glVertex3f ( 0.1, -1.0, 0.0);

glEnd();

// warna putih

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 1.0);

glVertex3f ( -0.5, -1.0, 0.0);

glVertex3f ( -0.5, 1.0, 0.0);

glVertex3f ( -0.2, 1.0, 0.0);

glVertex3f ( -0.2, -1.0, 0.0);

glEnd();

// warna kuning

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 0.0);

glVertex3f ( -0.8, -1.0, 0.0);

glVertex3f ( -0.8, 1.0, 0.0);

glVertex3f ( -0.5, 1.0, 0.0);

glVertex3f ( -0.5, -1.0, 0.0);

glEnd();

// warna ungu

glBegin(GL\_POLYGON);

glColor3f (1.0, 0.0, 1.0);

glVertex3f ( -1.1, -1.0, 0.0);

glVertex3f ( -1.1, 1.0, 0.0);

glVertex3f ( -0.8, 1.0, 0.0);

glVertex3f ( -0.8, -1.0, 0.0);

glEnd();

glFlush ();

}

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 3");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

1. Nomor 4a

/\*

\* GLUT Shapes Demo

\*

\* Written by Nigel Stewart November 2003

\*

\* This program is test harness for the sphere, cone

\* and torus shapes in GLUT.

\*

\* Spinning wireframe and smooth shaded shapes are

\* displayed until the ESC or q key is pressed. The

\* number of geometry stacks and slices can be adjusted

\* using the + and - keys.

\*/

#include <windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

void display(void)

{

/\* bersihkan layar dari titik pixel yang masih ada \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glBegin(GL\_POLYGON);

glEnd();

// warna merah

glColor3f(1.0, 0.0, 0.0);

glRectf(-1.0, 0.7, 1.0, 1.0);

// warna hijau

glColor3f (0.0, 1.0, 0.0);

glRectf(-1.0, 0.4, 1.0, 0.7);

// warna biru

glColor3f (0.0, 0.0, 1.0);

glRectf(-1.0, 0.1, 1.0, 0.4);

// warna hitam

glColor3f (0.0, 0.0, 0.0);

glRectf(-1.0, -0.2, 1.0, 0.1);

// warna putih

glColor3f (1.0, 1.0, 1.0);

glRectf(-1.0, -0.5, 1.0, -0.2);

// warna kuning

glColor3f (1.0, 1.0, 0.0);

glRectf(-1.0, -0.8, 1.0, -0.5);

// warna ungu

glColor3f (1.0, 0.0, 1.0);

glRectf(-1.0, -1.1, 1.0, -0.8);

glFlush ();

}

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 4a");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

1. Nomor 4b

/\*

\* GLUT Shapes Demo

\*

\* Written by Nigel Stewart November 2003

\*

\* This program is test harness for the sphere, cone

\* and torus shapes in GLUT.

\*

\* Spinning wireframe and smooth shaded shapes are

\* displayed until the ESC or q key is pressed. The

\* number of geometry stacks and slices can be adjusted

\* using the + and - keys.

\*/

#include <windows.h>

#ifdef \_\_APPLE\_\_

#include <GLUT/glut.h>

#else

#include <GL/glut.h>

#endif

#include <stdlib.h>

void display(void)

{

/\* bersihkan layar dari titik pixel yang masih ada \*/

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glBegin(GL\_POLYGON);

glEnd();

// warna merah

glColor3f(1.0, 0.0, 0.0);

glRectf(0.7, -1.0, 1.0, 1.0);

// warna hijau

glColor3f (0.0, 1.0, 0.0);

glRectf(0.4, -1.0, 0.7, 1.0);

// warna biru

glColor3f (0.0, 0.0, 1.0);

glRectf(0.1, -1.0, 0.4, 1.0);

// warna hitam

glColor3f (0.0, 0.0, 0.0);

glRectf(-0.2, -1.0, 0.1, 1.0);

// warna putih

glColor3f (1.0, 1.0, 1.0);

glRectf(-0.5, -1.0, -0.2, 1.0);

// warna kuning

glColor3f (1.0, 1.0, 0.0);

glRectf(-0.8, -1.0, -0.5, 1.0);

// warna ungu

glColor3f (1.0, 0.0, 1.0);

glRectf(-1.1, -1.0, -0.8, 1.0);

glFlush ();

}

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 4b");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

1. **Penjelasan Code Program**
   1. Nomor 1

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glColor3f (1.0, 1.0, 0.0);

glClear untuk membersihkan layer dari titik pixel, glPointSize untuk meberikan ukuran pada glPoint, dan glLineWidth untuk memberikan ukuran pada vertex selain dari glPoint. glColor3f untuk memberikan warna campuran antara merah dan hijau, karena nilai 1 terletak pada sumbu x dan y.

//glBegin(GL\_POINTS);

//glBegin(GL\_LINE\_STRIP);

//glBegin(GL\_LINE\_LOOP);

//glBegin(GL\_LINES);

//glBegin(GL\_TRIANGLES);

//glBegin(GL\_TRIANGLE\_FAN);

//glBegin(GL\_TRIANGLE\_STRIP);

//glBegin(GL\_QUADS);

//glBegin(GL\_QUAD\_STRIP);

glBegin(GL\_POLYGON);

Karena pada soal disuruh mencoba semua glBegin tersebut, maka semua syntax tersebut dicoba satu persatu dan diamati hasilnya.

glVertex3f (0.0, 0.0, 0.0);

glColor3f (1.0, 0.0, 0.0);

glVertex3f (0.0, 0.8, 0.0);

glColor3f (0.0, 1.0, 0.0);

glVertex3f (0.8, 0.0, 0.0);

glColor3f (0.0, 0.0, 1.0);

glVertex3f (0.0, -0.8, 0.0);

glColor3f (1.0, 1.0, 1.0);

glVertex3f (-0.8, 0.0, 0.0);

glColor3f (0.0, 0.0, 0.0);

glEnd();

glFlush ();

Kemudian menentukan titik koordinat pada vertex dengan glVertex3f dan memberi warna pada masing – masing koordinat tersebut dengan syntax glColor3f. kemudian menutupnya dengan glEnd, dan glFlush.

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

Membuat fungsi kunci gar ketika program dinjalankan dan apabila menerima inputan dari keyboar berupa huruf ‘q’ maka aplikasi akan tertutup.

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glPointSize(100.0);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("Primitif");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

Membuat fungsi main, dan memanggil glutInit untuk menginisialisasi program dan menentukan ukuran window dengan glutInitWindowSize, dan menentukan letak dimana window tersebut muncul dengan glutInitWIndowPosition. Kemudian memebri judul pada window dengan ‘primitid’ dengan glutCreateWindow, dan memanggi fungsi display dengan glutDisplayFunc, dan memanggi fungsi kuncin dengan glutKeyboardFunc.

1. Nomor 2

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glClear untuk membersihkan layer dari titik pixel, glPointSize untuk meberikan ukuran pada glPoint, dan glLineWidth untuk memberikan ukuran pada vertex selain dari glPoint.

// warna merah

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3f (-1.0, 0.7, 0.0);

glVertex3f (1.0, 0.7, 0.0);

glVertex3f (1.0, 1.0, 0.0);

glVertex3f (-1.0, 1.0, 0.0);

glEnd();

// warna hijau

glBegin(GL\_POLYGON);

glColor3f (0.0, 1.0, 0.0);

glVertex3f (-1.0, 0.4, 0.0);

glVertex3f (1.0, 0.4, 0.0);

glVertex3f (1.0, 0.7, 0.0);

glVertex3f (-1.0, 0.7, 0.0);

glEnd();

// warna biru

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 1.0);

glVertex3f (-1.0, 0.1, 0.0);

glVertex3f (1.0, 0.1, 0.0);

glVertex3f (1.0, 0.4, 0.0);

glVertex3f (-1.0, 0.4, 0.0);

glEnd();

// warna hitam

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 0.0);

glVertex3f (-1.0, -0.2, 0.0);

glVertex3f (1.0, -0.2, 0.0);

glVertex3f (1.0, 0.1, 0.0);

glVertex3f (-1.0, 0.1, 0.0);

glEnd();

// warna putih

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 1.0);

glVertex3f (-1.0, -0.5, 0.0);

glVertex3f (1.0, -0.5, 0.0);

glVertex3f (1.0, -0.2, 0.0);

glVertex3f (-1.0, -0.2, 0.0);

glEnd();

// warna kuning

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 0.0);

glVertex3f (-1.0, -0.8, 0.0);

glVertex3f (1.0, -0.8, 0.0);

glVertex3f (1.0, -0.5, 0.0);

glVertex3f (-1.0, -0.5, 0.0);

glEnd();

// warna ungu

glBegin(GL\_POLYGON);

glColor3f (1.0, 0.0, 1.0);

glVertex3f (-1.0, -1.1, 0.0);

glVertex3f (1.0, -1.1, 0.0);

glVertex3f (1.0, -0.8, 0.0);

glVertex3f (-1.0, -0.8, 0.0);

glEnd();

glFlush ();

Menentukan koordinat vertex pada titik awal, kemudian selanjutnya memberi warna dnegan menggunakan glColor3f, dan kemudian pada koordinat vertex selanjutnya masing-masing pada sumbu y dikurangi dengan 0.3 agar posisi bangun yang digambar berada dibawah bangun awal, dan tidak lupa memberi warna dengan menggunakan glColor3f. dan menutupnya dengan glEnd dan glFlush.

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

Membuat fungsi kunci gar ketika program dinjalankan dan apabila menerima inputan dari keyboar berupa huruf ‘q’ maka aplikasi akan tertutup.

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 2");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

Membuat fungsi main, dan memanggil glutInit untuk menginisialisasi program dan menentukan ukuran window dengan glutInitWindowSize, dan menentukan letak dimana window tersebut muncul dengan glutInitWIndowPosition. Kemudian memebri judul pada window dengan ‘No 2’ dengan glutCreateWindow, dan memanggi fungsi display dengan glutDisplayFunc, dan memanggi fungsi kuncin dengan glutKeyboardFunc.

1. Nomor 3

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glClear untuk membersihkan layer dari titik pixel, glPointSize untuk meberikan ukuran pada glPoint, dan glLineWidth untuk memberikan ukuran pada vertex selain dari glPoint.

// warna merah

glBegin(GL\_POLYGON);

glColor3f(1.0, 0.0, 0.0);

glVertex3f (0.7, -1.0, 0.0);

glVertex3f (0.7, 1.0, 0.0);

glVertex3f (1.0, 1.0, 0.0);

glVertex3f (1.0, -1.0, 0.0);

glEnd();

// warna hijau

glBegin(GL\_POLYGON);

glColor3f (0.0, 1.0, 0.0);

glVertex3f (0.4, -1.0, 0.0);

glVertex3f (0.4, 1.0, 0.0);

glVertex3f (0.7, 1.0, 0.0);

glVertex3f (0.7, -1.0, 0.0);

glEnd();

// warna biru

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 1.0);

glVertex3f ( 0.1, -1.0, 0.0);

glVertex3f ( 0.1, 1.0, 0.0);

glVertex3f ( 0.4, 1.0, 0.0);

glVertex3f ( 0.4, -1.0, 0.0);

glEnd();

// warna hitam

glBegin(GL\_POLYGON);

glColor3f (0.0, 0.0, 0.0);

glVertex3f ( -0.2, -1.0, 0.0);

glVertex3f ( -0.2, 1.0, 0.0);

glVertex3f ( 0.1, 1.0, 0.0);

glVertex3f ( 0.1, -1.0, 0.0);

glEnd();

// warna putih

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 1.0);

glVertex3f ( -0.5, -1.0, 0.0);

glVertex3f ( -0.5, 1.0, 0.0);

glVertex3f ( -0.2, 1.0, 0.0);

glVertex3f ( -0.2, -1.0, 0.0);

glEnd();

// warna kuning

glBegin(GL\_POLYGON);

glColor3f (1.0, 1.0, 0.0);

glVertex3f ( -0.8, -1.0, 0.0);

glVertex3f ( -0.8, 1.0, 0.0);

glVertex3f ( -0.5, 1.0, 0.0);

glVertex3f ( -0.5, -1.0, 0.0);

glEnd();

// warna ungu

glBegin(GL\_POLYGON);

glColor3f (1.0, 0.0, 1.0);

glVertex3f ( -1.1, -1.0, 0.0);

glVertex3f ( -1.1, 1.0, 0.0);

glVertex3f ( -0.8, 1.0, 0.0);

glVertex3f ( -0.8, -1.0, 0.0);

glEnd();

glFlush ();

}

Menentukan koordinat vertex pada titik awal, kemudian selanjutnya memberi warna dnegan menggunakan glColor3f, dan kemudian pada koordinat vertex selanjutnya masing-masing pada sumbu x dikurangi dengan 0.3 agar posisi bangun yang digambar berada di samping bangun awal, dan tidak lupa memberi warna dengan menggunakan glColor3f. dan menutupnya dengan glEnd dan glFlush.

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

Membuat fungsi kunci gar ketika program dinjalankan dan apabila menerima inputan dari keyboar berupa huruf ‘q’ maka aplikasi akan tertutup.

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 3");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

Membuat fungsi main, dan memanggil glutInit untuk menginisialisasi program dan menentukan ukuran window dengan glutInitWindowSize, dan menentukan letak dimana window tersebut muncul dengan glutInitWIndowPosition. Kemudian memebri judul pada window dengan ‘No 3’ dengan glutCreateWindow, dan memanggi fungsi display dengan glutDisplayFunc, dan memanggi fungsi kuncin dengan glutKeyboardFunc.

1. Nomor 4a

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glClear untuk membersihkan layer dari titik pixel, glPointSize untuk meberikan ukuran pada glPoint, dan glLineWidth untuk memberikan ukuran pada vertex selain dari glPoint.

glBegin(GL\_POLYGON);

glEnd();

// warna merah

glColor3f(1.0, 0.0, 0.0);

glRectf(-1.0, 0.7, 1.0, 1.0);

// warna hijau

glColor3f (0.0, 1.0, 0.0);

glRectf(-1.0, 0.4, 1.0, 0.7);

// warna biru

glColor3f (0.0, 0.0, 1.0);

glRectf(-1.0, 0.1, 1.0, 0.4);

// warna hitam

glColor3f (0.0, 0.0, 0.0);

glRectf(-1.0, -0.2, 1.0, 0.1);

// warna putih

glColor3f (1.0, 1.0, 1.0);

glRectf(-1.0, -0.5, 1.0, -0.2);

// warna kuning

glColor3f (1.0, 1.0, 0.0);

glRectf(-1.0, -0.8, 1.0, -0.5);

// warna ungu

glColor3f (1.0, 0.0, 1.0);

glRectf(-1.0, -1.1, 1.0, -0.8);

glFlush ();

}

Membuat glBegin untuk memulai dan glEnd untuk mengakhiri, kemudian memberikan warna dengan glColor3f dikuti dengan menetukan titik koordinat bangunsatu dengan glRectf yang memiliki value (x1,y1,x2,y2) dimana nilai y1 dan y2 memiliki selisih 0.3 agar bangun yang digambar sesuai. Kemudian langkah selanjutnya yaitu memberikan warna berbeda dengan menggunakan glColor3f agar memiliki tampilan ang berbeda kemudian dilanjut dengan glRectf yang sama seperti sebelumnya akan tetapi nilai y1 dan y2 dikurangi 0.3 agar bagun yang akan dibangun berada dibawah bangun awal yang sudah dibangun pada tahap sebeleumnya. Dilakukan terus menerus hingga pada bagian akhir diberi glFlush untuk mengakhiri program.

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

Membuat fungsi kunci gar ketika program dinjalankan dan apabila menerima inputan dari keyboar berupa huruf ‘q’ maka aplikasi akan tertutup.

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 4a");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

glutMainLoop();

return 0;

}

Membuat fungsi main, dan memanggil glutInit untuk menginisialisasi program dan menentukan ukuran window dengan glutInitWindowSize, dan menentukan letak dimana window tersebut muncul dengan glutInitWIndowPosition. Kemudian memebri judul pada window dengan ‘No 4a’ dengan glutCreateWindow, dan memanggi fungsi display dengan glutDisplayFunc, dan memanggi fungsi kuncin dengan glutKeyboardFunc.

1. Nomor 4b

glClear (GL\_COLOR\_BUFFER\_BIT);

glPointSize(10.0);

glLineWidth(10.0);

glClear untuk membersihkan layer dari titik pixel, glPointSize untuk meberikan ukuran pada glPoint, dan glLineWidth untuk memberikan ukuran pada vertex selain dari glPoint.

glBegin(GL\_POLYGON);

glEnd();

// warna merah

glColor3f(1.0, 0.0, 0.0);

glRectf(0.7, -1.0, 1.0, 1.0);

// warna hijau

glColor3f (0.0, 1.0, 0.0);

glRectf(0.4, -1.0, 0.7, 1.0);

// warna biru

glColor3f (0.0, 0.0, 1.0);

glRectf(0.1, -1.0, 0.4, 1.0);

// warna hitam

glColor3f (0.0, 0.0, 0.0);

glRectf(-0.2, -1.0, 0.1, 1.0);

// warna putih

glColor3f (1.0, 1.0, 1.0);

glRectf(-0.5, -1.0, -0.2, 1.0);

// warna kuning

glColor3f (1.0, 1.0, 0.0);

glRectf(-0.8, -1.0, -0.5, 1.0);

// warna ungu

glColor3f (1.0, 0.0, 1.0);

glRectf(-1.1, -1.0, -0.8, 1.0);

glFlush ();

}

Membuat glBegin untuk memulai dan glEnd untuk mengakhiri, kemudian memberikan warna dengan glColor3f dikuti dengan menetukan titik koordinat bangunsatu dengan glRectf yang memiliki value (x1,y1,x2,y2) dimana nilai x1 dan x2 memiliki selisih 0.3 agar bangun yang digambar sesuai. Kemudian lamngkah selanjutnya yaitu memberikan warna berbeda dengan menggunakan glColor3f agar memiliki tampilan ang berbeda kemudian dilanjut dengan glRectf yang sama seperti sebelumnya akan tetapi nilai x1 dan x2 dikurangi 0.3 agar bagun yang akan dibangun berada di samping bangun awal yang sudah dibangun pada tahap sebeleumnya. Dilakukan terus menerus hingga pada bagian akhir diberi glFlush untuk mengakhiri program.

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

{

switch (key)

{

/\* aplikasi berhenti ketika tombol q ditekan \*/

case 27 :

case 'q':

exit(0);

break;

}

glutPostRedisplay();

}

Membuat fungsi kunci gar ketika program dinjalankan dan apabila menerima inputan dari keyboar berupa huruf ‘q’ maka aplikasi akan tertutup.

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("No 4b");

glutDisplayFunc(display);

glutKeyboardFunc(kunci);

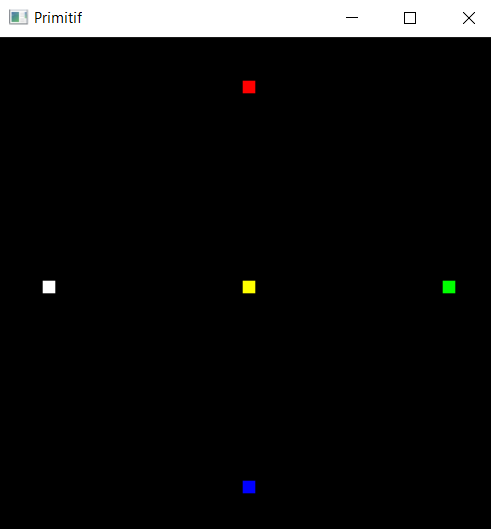
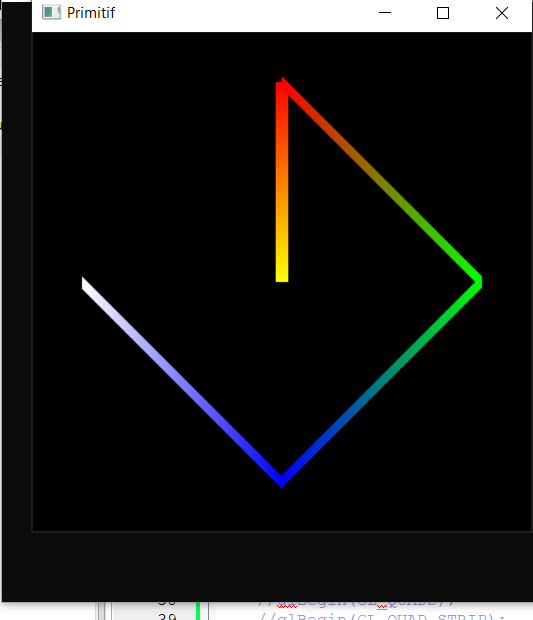
glutMainLoop();

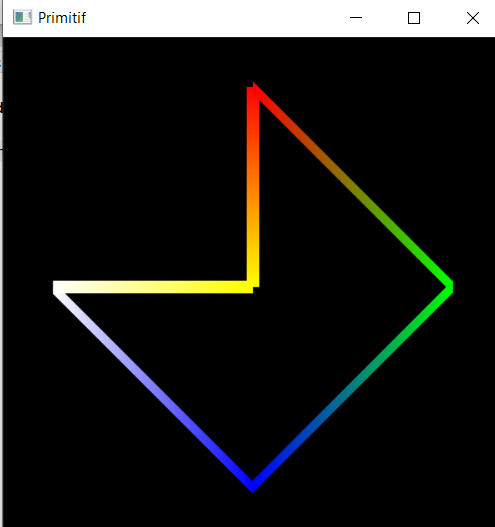
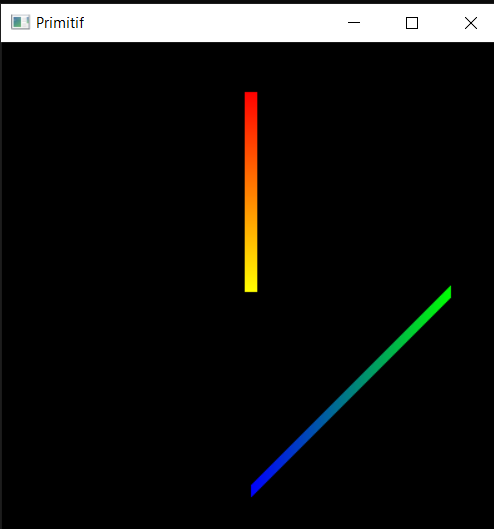
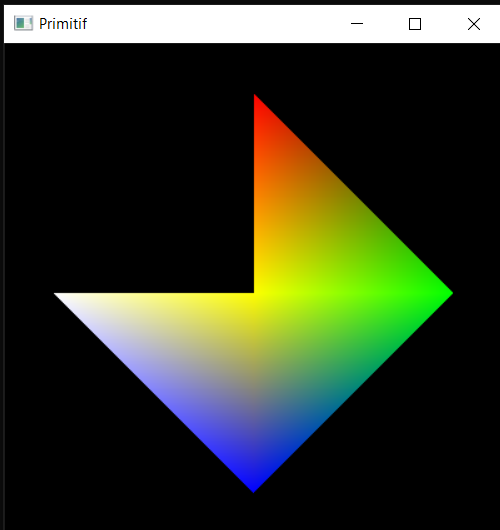
return 0;

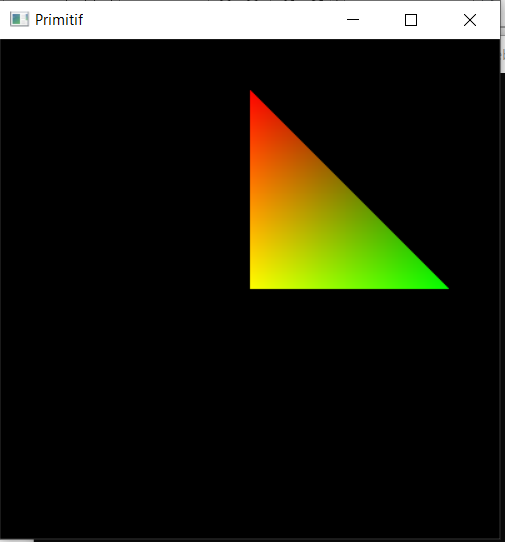
}

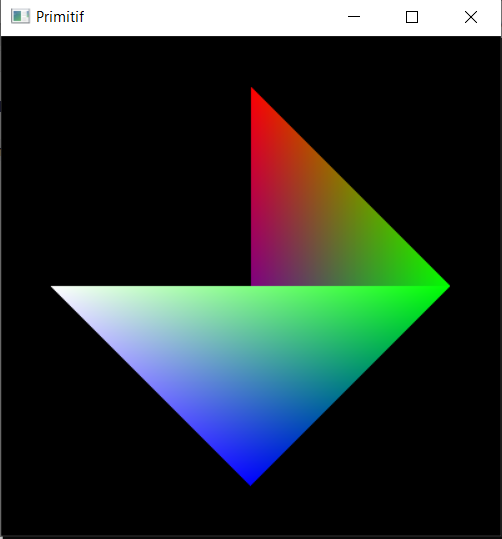
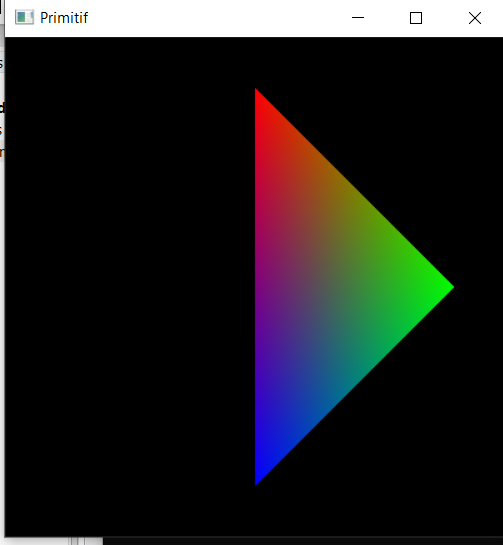
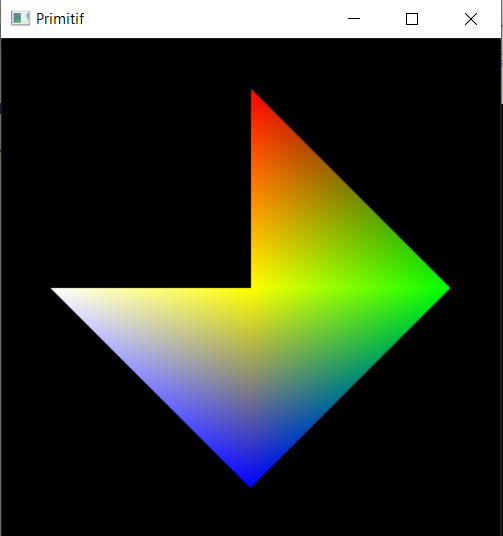
Membuat fungsi main, dan memanggil glutInit untuk menginisialisasi program dan menentukan ukuran window dengan glutInitWindowSize, dan menentukan letak dimana window tersebut muncul dengan glutInitWIndowPosition. Kemudian memebri judul pada window dengan ‘No 4b’ dengan glutCreateWindow, dan memanggi fungsi display dengan glutDisplayFunc, dan memanggi fungsi kuncin dengan glutKeyboardFunc.

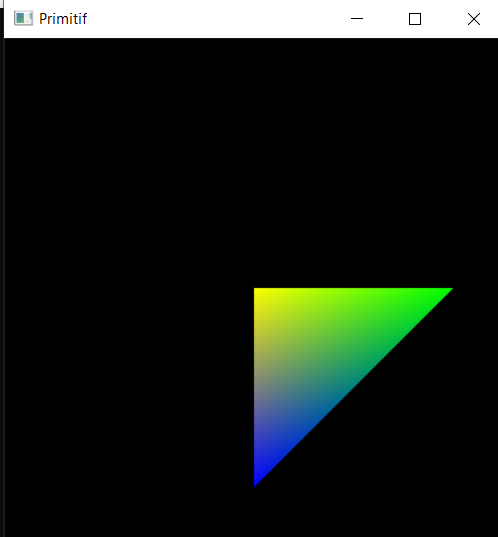
1. **Hasil Running Program** 
   1. Nomor 1



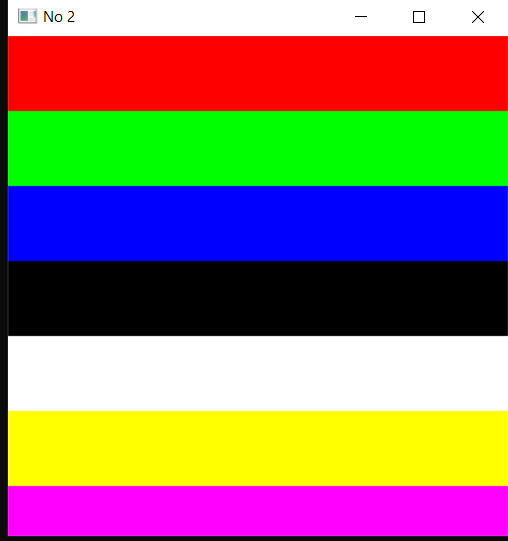




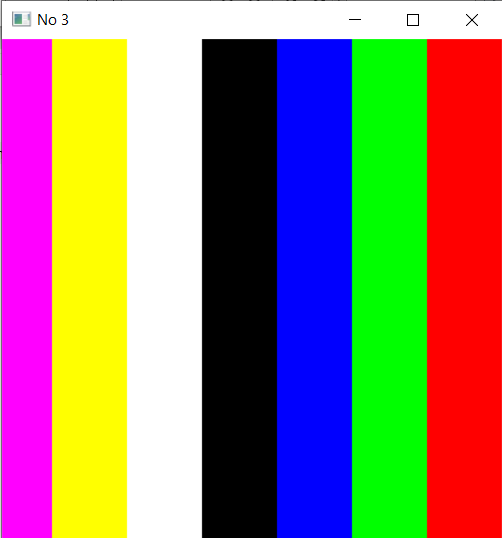


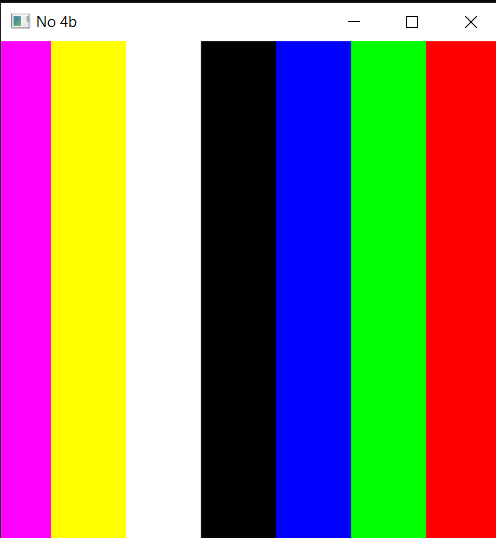


1. Nomor 2



* 1. Nomor 3



* 1. Nomor 4a dan 4b



**BAB II**

**PENUTUP**

1. **Kesimpulan**
2. Mahasiswa mampu mengaplikasikan materi pada modul 1
3. Mahsisswa mampu membuat aplikasi berbasis grafik (openGL) melalui CodeBlocks
4. Mahasiswa mengetahui serta dapat memahami dan menjalankan fungsi yang ada pada openGL project
5. **Saran**

Diperlukan banyak latihan dan mencoba megotak-atik sendiri agar mahasiswa lebih memahami dan mendalami fungsi – fungsi yang ada pada openGL.