**TECHNICAL REPORT**

**GRAFIKA KOMPUTER**

**MODUL 3**



**Disusun Oleh :**

TGL. PRAKTIKUM : Jum’at, 30 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 dan memahami cara melakukan transformasi pada object dengan openGL.

1. **DASAR TEORI**

Dalam matematika, transformasi adalah fungsi yang memetakan suatu set X ke set yang lain ataupun ke set X sendiri. Dalam dunia Grafika Komputer, set X (yang mengalami proses transformasi) biasanya berupa strukur geometri, sehingga disebut transformasi geometri. Terdapat banyak jenis operasi transformasi geometri: translasi, refleksi, rotasi, scaling, shearing.

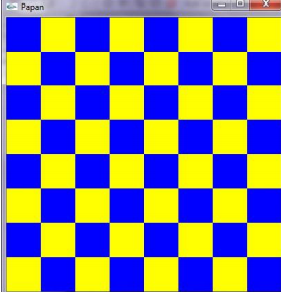
OpenGL memiliki 3 perintah transformasi:

1. glTranslated(a, b, c): melakukan operasi translasi/pergeseran sejauh a pada sumbu x, sejauh b pada sumbu y, dan sejauh c pada sumbu z. Contoh: jika ingin menggeser obyek sejauh 4 pada sumbu x dan -3 pada sumbu y, maka perintahnya adalah: glTranslated(4.0, -3.0, 0.0).
2. glScaled(d, e, f): melakukan penskalaan sebesar d pada sumbu x, sebesar e pada sumbu y, sebesar f pada sumbu z. Contoh: jika ingin memperbesar obyek pada sumbu x sebesar 2 kali dan memperkecil obyek menjadi seperempatnya, maka perintahnya adalah: glScaled(2.0, 0.25, 0.0).
3. glRotated(alpha, i, j, k): melakukan rotasi sebesar alpha. Alpha ada dalam satuan derajat, bukan radian. i, j, dan k mewakili sumbu rotasi x, y, dan z. Set nilainya menjadi 1.0 pada sumbu yang diingikan. Contoh: jika ingin merotasi obyek sebesar 90 derajat pada sumbu x, maka perintahnya adalah: glRotated(90.0, 1, 0, 0).

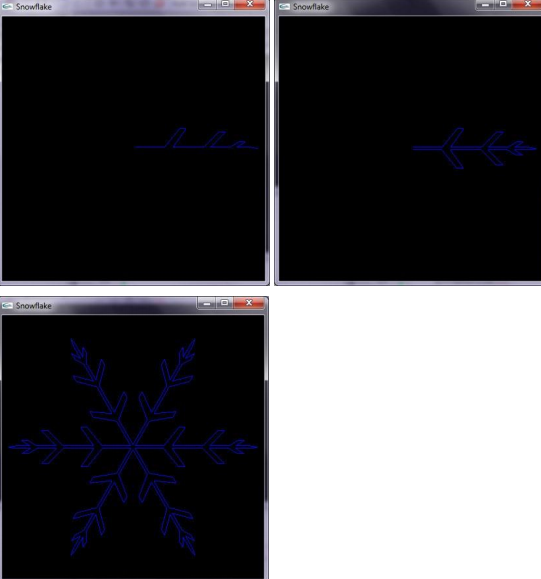
**BAB II**

**PEMBAHASAN**

1. **SOAL**
2. Buat checker board 8 x 8 kotak dengan menggunakan glTranslate.

****

1. Buat snow flake (bunga salju) berikut menggunakan transformasi.

****

Cukup buat 1 bagian, lalu duplikasi menggunakan transformasi 11 kali untuk membuat keseluruhan gambar. Gambar tidak perlu persis, asal cukup mirip.

1. Buat program dengan tampilan sebagai berikut:

****

Gambar tidak harus persis, asal cukup mirip.

1. **JAWABAN**
2. **Code Program**
3. 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<iostream>

#include <stdlib.h>

void biru()

{

glColor3f(0,0,1);

glRecti(-20,15,-15,20);

}

void kuning()

{

glColor3f(1,1,0);

glRecti(-15,15,-10,20);

}

void gabunganObject()

{

biru();

kuning();

glTranslated(10,0,0);

biru();

glTranslated(0,0,0);

kuning();

glTranslated(10,0,0);

biru();

glTranslated(0,0,0);

kuning();

glTranslated(10,0,0);

biru();

glTranslated(0,0,0);

kuning();

}

void finalObject()

{

gabunganObject();

glRotated(180,0,0,1);

glTranslated(30,-30,0);

gabunganObject();

}

void display(void)

{

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

glClear (GL\_COLOR\_BUFFER\_BIT);

finalObject();

glTranslated(-30,10,0);

finalObject();

glTranslated(-30,-10,0);

finalObject();

glTranslated(-30,10,0);

finalObject();

glFlush ();

}

void myinit()

{

glClearColor(1,1,1,0); //untuk memberi warna pada background

glColor3f(1.0, 0.0, 0.0); // color pada object

glEnable(GL\_POINT\_SMOOTH);

glEnable(GL\_BLEND);

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

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-20.0, 20.0, -20.0, 20.0); // kordinat pada area menggambar

glMatrixMode(GL\_MODELVIEW);

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("Nomer 1");

glutDisplayFunc(display);

myinit();

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<iostream>

#include <stdlib.h>

void object()

{

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_LINE\_STRIP);

glVertex3f(0.0, 0.1, 0.0);

glVertex3f(1.75, 0.1, 0.0);

glVertex3f(2.9, 0.8, 0.0);

glVertex3f(3.2, 0.8, 0.0);

glVertex3f(2.3, 0.1, 0.0);

glVertex3f(3.75, 0.1, 0.0);

glVertex3f(4.9, 0.6, 0.0);

glVertex3f(5.2, 0.6, 0.0);

glVertex3f(4.3, 0.1, 0.0);

glVertex3f(5.75, 0.1, 0.0);

glVertex3f(6.6, 0.4, 0.0);

glVertex3f(6.9, 0.4, 0.0);

glVertex3f(6.3, 0.1, 0.0);

glVertex3f(7.5, 0.1, 0.0);

glVertex3f(7.9, 0.0, 0.0);

glEnd();

}

void editObject()

{

glTranslated(0, 0, 0);

object();

glRotatef(180, 1, 0, 0);

glTranslated(0, 0, 0);

object();

glRotatef(180, 0, 1, 0);

glTranslated(0, 0, 0);

object();

glRotatef(180, 1, 0, 0);

glTranslated(0, 0, 0);

object();

}

void display(void)

{

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

glClear (GL\_COLOR\_BUFFER\_BIT);

glScaled(2, 2, 0);

editObject();

glRotatef(55, 0, 0, 0.5);

glTranslated(0, 0, 0);

editObject();

glRotatef(110, 0, 0, 0.5);

glTranslated(0,0,0);

editObject();

glFlush ();

}

void myinit()

{

glClearColor(1,1,1,0); //untuk memberi warna pada background

glColor3f(1.0, 0.0, 0.0); // color pada object

glEnable(GL\_POINT\_SMOOTH);

glEnable(GL\_BLEND);

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

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-20.0, 20.0, -20.0, 20.0); // kordinat pada area menggambar

glMatrixMode(GL\_MODELVIEW);

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("Nomer 2");

glutDisplayFunc(display);

myinit();

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<iostream>

#include <stdlib.h>

void object()

{

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_LINE\_STRIP);

glVertex3f(-0.34, 1.7, 0.0);

glVertex3f(0.13, 1.7, 0.0);

glVertex3f(-0.56, 3.8, 0.0);

glVertex3f(-1.5, 0.81, 0.0);

glVertex3f(-1.16, 0.6, 0.0);

glVertex3f(-0.55, 2.45, 0.0);

glVertex3f(-0.34, 1.7, 0.0);

glEnd();

}

void editObject()

{

glTranslated(0, 0, 0);

object();

glRotatef(-72.5, 0.0, 0.0, 1.0);

glTranslated(-0.355, -1.075, 0.0);

object();

}

void finalObject()

{

glTranslated(0, 0, 0);

editObject();

glRotatef(-72.25, 0.0, 0.0, 1.0);

glTranslated(-0.355, -1.075, 0.0);

editObject();

glRotatef(-71, 0.0, 0.0, 1.0);

glTranslated(-0.355, -1.05, 0.0);

object();

}

void display(void)

{

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

glClear (GL\_COLOR\_BUFFER\_BIT);

glScaled(5, 5, 0);

glTranslated(0.5, -0.5, 0);

finalObject();

glFlush ();

}

void myinit()

{

glClearColor(1,1,1,0); //untuk memberi warna pada background

glColor3f(1.0, 0.0, 0.0); // color pada object

glEnable(GL\_POINT\_SMOOTH);

glEnable(GL\_BLEND);

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

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-20.0, 20.0, -20.0, 20.0); // kordinat pada area menggambar

glMatrixMode(GL\_MODELVIEW);

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(400,400);

glutInitWindowPosition(100,100);

glutInitDisplayMode(GLUT\_RGB | GLUT\_SINGLE);

glutCreateWindow("Nomer 3");

glutDisplayFunc(display);

myinit();

glutMainLoop();

return 0;

}

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

void biru()

{

glColor3f(0,0,1);

glRecti(-20,15,-15,20);

}

void kuning()

{

glColor3f(1,1,0);

glRecti(-15,15,-10,20);

}

Membuat fungsi kuning dan biru yang didalamnya terdapat syntax glcolor untuk menetukan masing – masing warna sesuai dengan nama fungsi, kemudian membuat sebuah object kotak sekaligus dengan titik koordinatnya dengan fungsi glRecti.

void gabunganObject()

{

biru();

kuning();

glTranslated(10,0,0);

biru();

glTranslated(0,0,0);

kuning();

glTranslated(10,0,0);

biru();

glTranslated(0,0,0);

kuning();

glTranslated(10,0,0);

biru();

glTranslated(0,0,0);

kuning();

}

Membuat fungsi gabunganObject yang berisi pemanggilan object biru dan kuning yang sebelumnya sudah dibuat, kemudia melakukan transformasi glTranslated atau pergeseran agar bentuk object yang dibuat sesuai dengan tugas yaitu membentuk papan catur.

void finalObject()

{

gabunganObject();

glRotated(180,0,0,1);

glTranslated(30,-30,0);

gabunganObject();

}

Membuat fungsi finalObject untuk melakukan transformasi glRotated / rotasi sebesar 180 derajat pada fungsi gabunganObject, kemudian melakukan transformasi dengan glTranslated agar object dari fungsi gabunganObject tersusun dengan rapi.

void display(void)

{

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

glClear (GL\_COLOR\_BUFFER\_BIT);

finalObject();

glTranslated(-30,10,0);

finalObject();

glTranslated(-30,-10,0);

finalObject();

glTranslated(-30,10,0);

finalObject();

glFlush ();

}

Kemudian pada fungsi display dipanggil fungsi finalObject agar tampil pada output, dan kemudian melakukan transformasi dengan glTranslated agar fungsi finalObject yang akan dipanggil selanutnya bisa tersusun ke bawah dengan rapi.

1. Nomor 2

void object()

{

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_LINE\_STRIP);

glVertex3f(0.0, 0.1, 0.0);

glVertex3f(1.75, 0.1, 0.0);

glVertex3f(2.9, 0.8, 0.0);

glVertex3f(3.2, 0.8, 0.0);

glVertex3f(2.3, 0.1, 0.0);

glVertex3f(3.75, 0.1, 0.0);

glVertex3f(4.9, 0.6, 0.0);

glVertex3f(5.2, 0.6, 0.0);

glVertex3f(4.3, 0.1, 0.0);

glVertex3f(5.75, 0.1, 0.0);

glVertex3f(6.6, 0.4, 0.0);

glVertex3f(6.9, 0.4, 0.0);

glVertex3f(6.3, 0.1, 0.0);

glVertex3f(7.5, 0.1, 0.0);

glVertex3f(7.9, 0.0, 0.0);

glEnd();

}

Mmebuat fungsi object untuk membentuk object 1 batang snow dengan menghubungkan masing – masing vertex pada fungsi tersebut. Dan memberikan warna dengan glColor.

void editObject()

{

glTranslated(0, 0, 0);

object();

glRotatef(180, 1, 0, 0);

glTranslated(0, 0, 0);

object();

glRotatef(180, 0, 1, 0);

glTranslated(0, 0, 0);

object();

glRotatef(180, 1, 0, 0);

glTranslated(0, 0, 0);

object();

}

Kemudian membuat sebuah fungsi editObject untuk melekukan transformasi pada object yang ada di fungsi object agar bentuknya sesuai, setelah ditransforamasi menggunakan glRotate dan glTranslate, dipanggil kembali fungsi obect akan membentuk sebuah object gabungan dari beberapa object sebelumnya.

void display(void)

{

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

glClear (GL\_COLOR\_BUFFER\_BIT);

glScaled(2, 2, 0);

editObject();

glRotatef(55, 0, 0, 0.5);

glTranslated(0, 0, 0);

editObject();

glRotatef(110, 0, 0, 0.5);

glTranslated(0,0,0);

editObject();

glFlush ();

}

Pada fungsi display digunakan glScaled agar object yang terbentuk bertambah besar sebesar 2 pada sumbu x dan y. dan memanggi fungsi editObject agar ditampilkan kemudian dilakukan transformasi menggunakan glRotate dan glTranslate agar bentuk object bergeser dan memiliki kemiringan yang sesuai pada soal.

1. Nomor 3

void object()

{

glColor3f(0.0, 0.0, 1.0);

glBegin(GL\_LINE\_STRIP);

glVertex3f(-0.34, 1.7, 0.0);

glVertex3f(0.13, 1.7, 0.0);

glVertex3f(-0.56, 3.8, 0.0);

glVertex3f(-1.5, 0.81, 0.0);

glVertex3f(-1.16, 0.6, 0.0);

glVertex3f(-0.55, 2.45, 0.0);

glVertex3f(-0.34, 1.7, 0.0);

glEnd();

}

Membuat sebuah fungsi object untuk membuat object awal, yang nantinya akan digabung agar membentuk object baru yang diinginkan dengan menggunakan gabungan dari glVertex agar membentuk object tersebut, dan untuk memberi warna pada object tersebut menggunakan glColor dan object tersebut dibentuk menggunakan fungsi GL\_LINE\_STRIP

void editObject()

{

glTranslated(0, 0, 0);

object();

glRotatef(-72.5, 0.0, 0.0, 1.0);

glTranslated(-0.355, -1.075, 0.0);

object();

}

Membuat sebuah fungsi editObject untuk menggabungkan dua object dari fungsi object yang sudah ditransformasi menggunakan glRotate dan glTranslate agar object tersebut tidak menumpuk.

void finalObject()

{

glTranslated(0, 0, 0);

editObject();

glRotatef(-72.25, 0.0, 0.0, 1.0);

glTranslated(-0.355, -1.075, 0.0);

editObject();

glRotatef(-71, 0.0, 0.0, 1.0);

glTranslated(-0.355, -1.05, 0.0);

object();

}

Selanjutnya adalah menggabungkan beberapa object yang dihasilkan oleh fungsi editObject dan fungsi object untuk membuat sebauh object baru sesuai dengan soal dengan cara mentransformasi menggunakan glRotate dan glTranslate sebelum melakukan pemanggilan fungsi editObject berikutnya.

void display(void)

{

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

glClear (GL\_COLOR\_BUFFER\_BIT);

glScaled(5, 5, 0);

glTranslated(0.5, -0.5, 0);

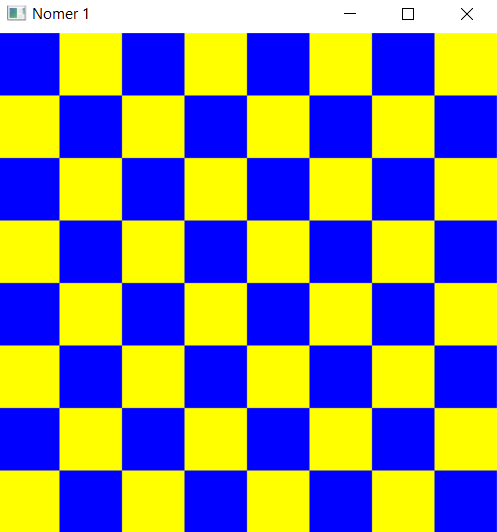
finalObject();

glFlush ();

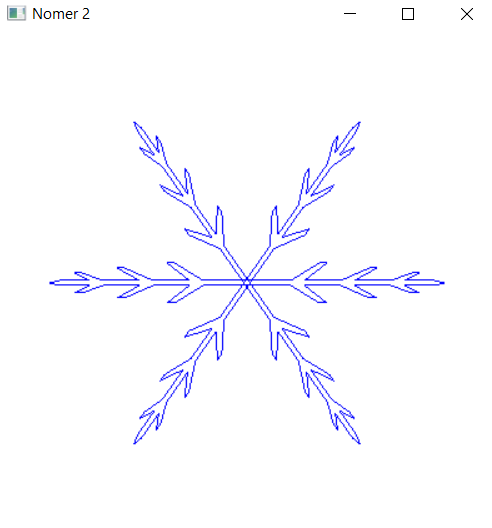
}

Kemudian pada fungsi display melakukan transformasi glScaled sebesar 5 pada sumbu x dan y, dan transformasi glTranslated agar object yang diasilkan bergeser ke tengah, selanjutnya memanggil fungsi finalObject, agar object yang digambar atau dibuat tampil pada output.

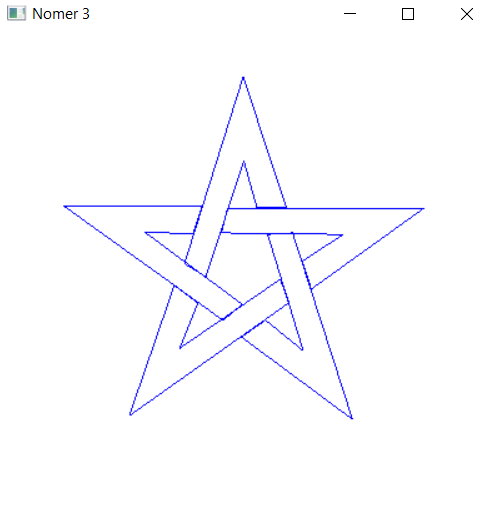
1. **Hasil Running Program** 
   1. Nomor 1



1. Nomor 2



1. Nomor 3



**BAB III**

**PENUTUP**

1. **Kesimpulan**
2. Mahasiswa mampu melakukan transformasi pada object yang dibuat.
3. Terdapat 3 fungsi transformasi yang dapat digunakan, dan masing – masing transformasi dapat digabungkan dalam pemakaiannya.
4. glTranslated dan glRotated sangat membantu untuk melakukan pergeseran pada object, sedengkan glScaled sangat berguna untuk memberpesar suatu object.
5. **Saran**

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