

**TECHNICAL REPORT**  
**GRAFIKA KOMPUTER**  
**MODUL 4**



**Disusun Oleh :**

TGL. PRAKTIKUM	: Jum'at, 21 Mei 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

### A. TUJUAN

Mahasiswa dapat mengetahui dan memahami cara melakukan transformasi pada object dengan `glPushMatrix()` dan `glPopMatrix()`.

### B. DASAR TEORI

Yang perlu pula diperhatikan di topik transformasi adalah Current Transformation Matrix (CTM). Perintah OpenGL yang berkaitan dengan CTM ini adalah: `glPushMatrix()`, `glPopMatrix()`, dan `glLoadIdentity()`. Berikut keterangan terkait ketiga perintah tersebut:

1. `glPushMatrix()` menduplikasi CTM dan meletakkannya di bagian teratas stack
2. `glPopMatrix()` pop matrix teratas dari stack
3. `glLoadIdentity()` mengubah matrix teratas stack menjadi matrix identitas

Contoh:

1. Inisialisasi stack

```
glMatrixMode(GL_MODEL_VIEW);  
glLoadIdentity();
```

I

2. Scale by 2

```
glScaled(1.0, 2.0, 1.0);
```

S(2)

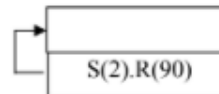
3. Rotate terhadap x by 90 derajat

```
glRotated(90, 1, 0, 0);
```

S(2).R(90)

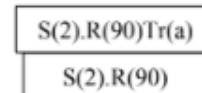
Push stack

4. `glPushMatrix();`      S(2).R(90) copy



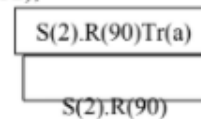
5. Translate by a

```
glTranslatef(a.x, a.y, a.z);
```



6. Menggambar point pada layar `glBegin(GL_POINTS);`

```
glVertex3f(1,1,1);  
glEnd();
```



- proses perkalian matriksnya:  $S(2) * R(90) * Tr(a) * (1,1,1)$

7. Pop off stack `glPopMatrix();`

$S(2).R(90)$

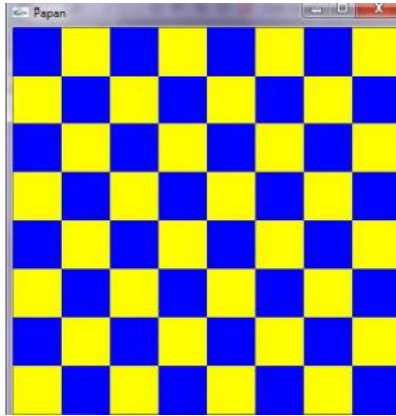
Stack transformasi ini berguna kalau kita ingin membuat hirarki transformasi. Contoh: kita hendak memodelkan tangan yang terdiri dari lengan atas, lengan bawah, telapak tangan, jari-jari. Untuk keseluruhan tangan, transformasinya kita simpan di stack terbawah. Untuk lengan bawah, telapak tangan, dan jari-jari, transformasinya kita simpan di stack nomor 2 dari bawah. Untuk telapak tangan dan jari-jari, transformasinya kita simpan di stack nomor 3 dari bawah. Sedangkan untuk jari-jari, transformasinya kita simpan di stack teratas. Dengan demikian kita jadi lebih fleksibel untuk mentransformasi tiap bagian obyek.

## BAB II

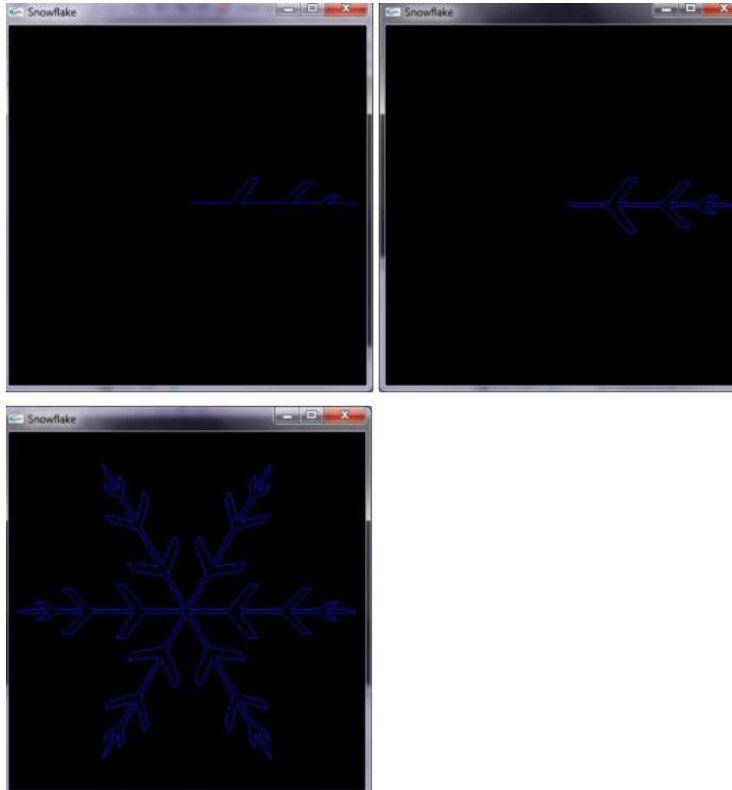
### PEMBAHASAN

#### A. SOAL

1. Buat checker board 8 x 8 kotak dengan menggunakan `glTranslate`.

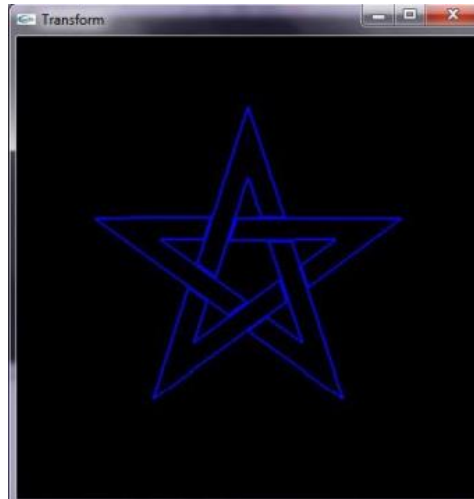


2. Buat snow flake (bunga salju) berikut menggunakan transformasi.



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

3. Buat program dengan tampilan sebagai berikut:



Gambar tidak harus persis, asal cukup mirip.

## B. JAWABAN

### 1. Code Program

#### 1) 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 display(void)
{
    /* bersihkan layar dari titik pixel yang masih ada */
    glClear (GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    gabunganObject();
    glPopMatrix();
    glPushMatrix();
    glRotated(180,0,0,1);
    glTranslated(0,-30,0);

```

```

gabunganObject();
glPopMatrix();
glPushMatrix();
glTranslated(0,-10,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glRotated(180,0,0,1);
glTranslated(0,-20,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glTranslated(0,-20,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glRotated(180,0,0,1);
glTranslated(0,-10,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glTranslated(0,-30,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glRotated(180,0,0,1);
glTranslated(0,0,0);
gabunganObject();
glPopMatrix();
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;
}

```

## 2) 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);
        glPushMatrix();
        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();
        glPopMatrix();
        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;
}

```

```
}
```

### 3) 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 display(void)
```

```

{
    /* bersihkan layar dari titik pixel yang masih ada */
    glClear (GL_COLOR_BUFFER_BIT);
    glScaled(5, 5, 0);
    glPushMatrix();
    object();
    glTranslated(0.35,1,0);
    glRotated(70,0,0,1);
    object();
    glPopMatrix();
    glPushMatrix();
    glTranslated(-1.07,-0.02,0);
    glRotated(290,0,0,1);
    object();
    glPopMatrix();
    glPushMatrix();
    glTranslated(-1.46,1,0);
    glRotated(220,0,0,1);
    object();
    glPopMatrix();
    glPushMatrix();
    glTranslated(-0.55,1.73,0);
    glRotated(145,0,0,1);
    object();
    glPopMatrix();

    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;
}

```

## 2. 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 menentukan 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, kemudian melakukan transformasi glTranslated atau pergeseran agar bentuk object yang dibuat sesuai dengan tugas yaitu membentuk papan catur.

```

void display(void)
{
    /* bersihkan layar dari titik pixel yang masih ada */
    glClear (GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    gabunganObject();
    glPopMatrix();
    glPushMatrix();
    glRotated(180,0,0,1);
    glTranslated(0,-30,0);
    gabunganObject();
}

```

```
glPopMatrix();
glPushMatrix();
glTranslated(0,-10,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glRotated(180,0,0,1);
glTranslated(0,-20,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glTranslated(0,-20,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glRotated(180,0,0,1);
glTranslated(0,-10,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glTranslated(0,-30,0);
gabunganObject();
glPopMatrix();
glPushMatrix();
glRotated(180,0,0,1);
glTranslated(0,0,0);
gabunganObject();
glPopMatrix();
glFlush ();
}
```

Kemudian pada fungsi display dipanggil fungsi gabungan object, dimana fungsi tersebut diletakkan didalam glPushMatrix dan glPopMatrix agar fungsi selanjutnya yang akan dipanggil akan kembali ke titik koordinat awal. Hal ini dilakukan sebanyak 8 kali agar terbentuk papan catur seperti pada soal.

2) 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();
}
```



Membuat 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 melakukan transformasi pada object yang ada di fungsi object agar bentuknya sesuai, setelah ditransformasi menggunakan glRotate dan glTranslate, dipanggil kembali fungsi object 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);
    glPushMatrix();
    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();
    glPopMatrix();
    glFlush ();
}

```

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

### 3) 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 display(void)

```
{
    /* bersihkan layar dari titik pixel yang masih ada */
    glClear (GL_COLOR_BUFFER_BIT);
    glScaled(5, 5, 0);
    glPushMatrix();
    object();
    glTranslated(0.35,1,0);
    glRotated(70,0,0,1);
    object();
    glPopMatrix();
    glPushMatrix();
    glTranslated(-1.07,-0.02,0);
    glRotated(290,0,0,1);
    object();
    glPopMatrix();
    glPushMatrix();
    glTranslated(-1.46,1,0);
    glRotated(220,0,0,1);
    object();
    glPopMatrix();
    glPushMatrix();
    glTranslated(-0.55,1.73,0);
    glRotated(145,0,0,1);
```

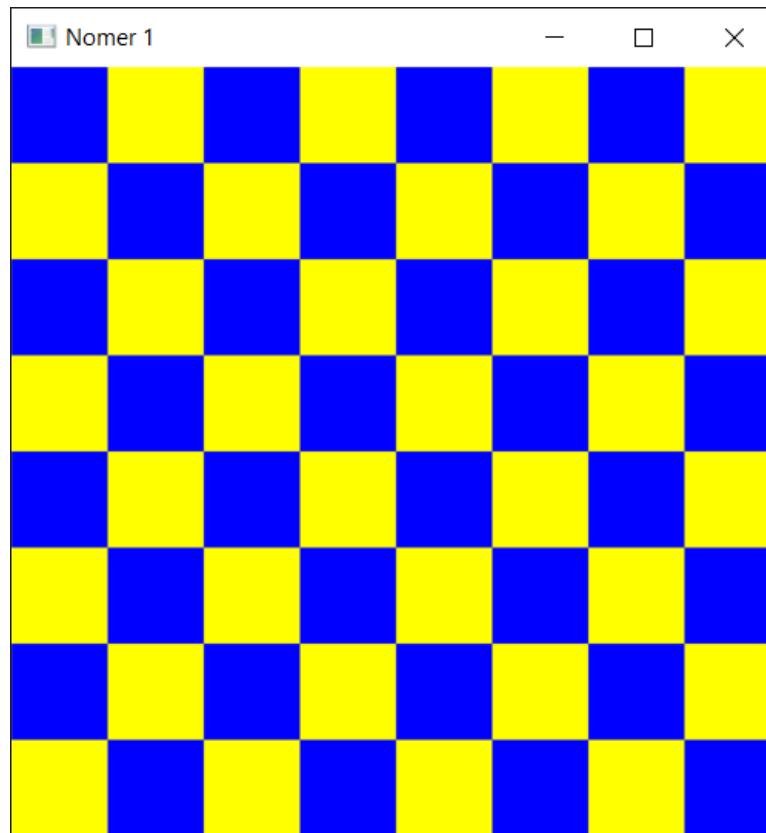
```
object();  
glPopMatrix();
```

```
glFlush ();  
}
```


Kemudian pada fungsi display melakukan transformasi `glScaled` sebesar 5 pada sumbu x dan y, dan transformasi `glTranslated` agar object yang dihasilkan bergeser ke tengah, selanjutnya memanggil fungsi `Object` dan diletakkan diantara `glPushMatrix` dan `glPopMatrix` agar fungsi selanjutnya ayang akan dipanggil kembali ke titik koordinat awal, dan akhirnya object yang digambar atau dibuat tampil sesuai output pada soal.

### 3. Hasil Running Program

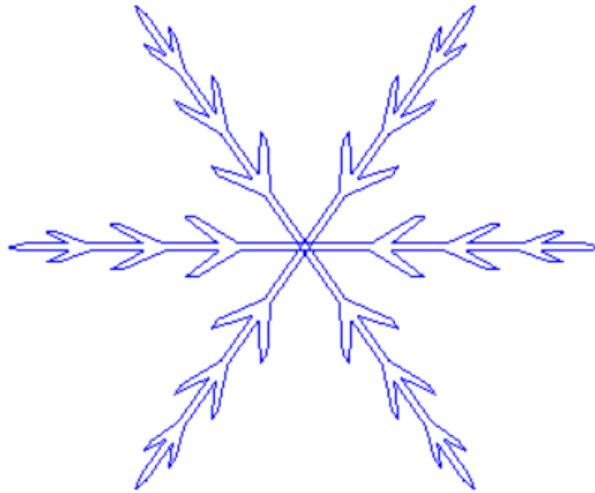
#### 1) Nomor 1



2) Nomor 2

 Nomer 2

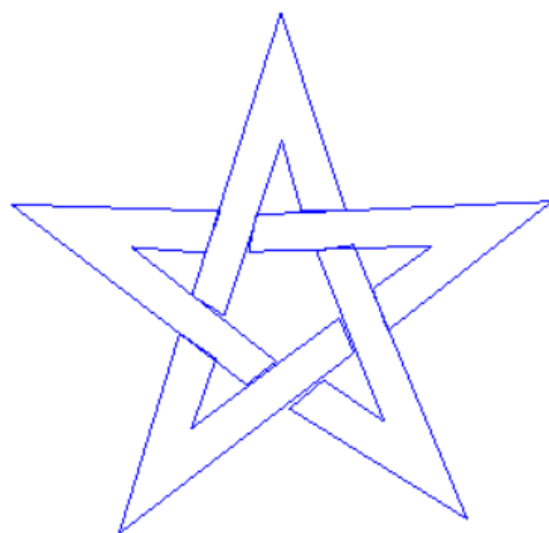
— □ ×



3) Nomor 3

Nomer 3

— □ ×



## **BAB III**

### **PENUTUP**

#### **A. Kesimpulan**

1. Mahasiswa mampu melakukan transformasi pada object yang dibuat menggunakan `glPushMatrix` dan `glPopMatrix`.
2. Terdapat 3 fungsi transformasi yang dapat digunakan, dan masing – masing transformasi dapat digabungkan dalam pemakaiannya.
3. `glPushMatrix` dan `glPopMatrix` sangat penting untuk digunakan.

#### **B. Saran**

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