

Modul 4

Transformasi Geometri Lanjut

I. Tugas Pendahuluan

1. Jika titik A(1,2) ditranslasi (2,2) kemudian ditranslasi lagi (1,-1), dikoordinat manakah posisi akhir titik A?
2. Jika titik B(1,2) ditranslasi (1,-1) kemudian ditranslasi lagi (2,2), dikoordinat manakah posisi akhir titik B?
3. Samakah hasil jawaban soal nomor 1 dan 2?
4. Jika titik C(1,2) ditranslasi (2,2) kemudian dirotasi 90 derajat, dikoordinat manakah posisi akhir titik C?
5. Jika titik D(1,2) dirotasi 90 derajat kemudian ditranslasi (2,2), dikoordinat manakah posisi akhir titik D?
6. Samakah hasil jawaban soal nomor 4 dan 5? Kenapa demikian?

II. Pengantar

Salah satu kegunaan kombinasi Transformasi adalah untuk melakukan rotasi pada obyek terhadap arbitrary point/sembarang titik(bukan terhadap sumbu koordinat). Metode rotasi terhadap arbitrary point adalah pertama-tama mentranslasikan obyek untuk berhimpit dengan sumbu koordinat, diikuti dengan rotasi, dan terakhir men-translasikan kembali obyek pada posisinya semula.

Program 6.1 menunjukkan contoh kombinasi transformasi.

```
void display()
{
    glClearColor(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0,0.0,0.5);

    // Gambar kotak pertama di sudut kiri bawah
    glRecti(0,0, 10, 10);

    //rotasi kotak kedua terhadap titik tengah
    kotak    glTranslated(25.0, 25.0, 0);
    glRotated(45, 0, 0, 1.0);    glTranslated(-
25.0, -25.0, 0);    glRecti(20, 20, 30, 30);

    glFlush();
}
void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0,50.0,0.0,50.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(1.0,1.0,1.0,1.0);
    glColor3f(0.0,0.0,0.0);
}
int main(int argc, char* argv[])
```

```
{
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400,400);
    glutInitWindowPosition(100,100);
    glutCreateWindow("Transform");
    glutDisplayFunc(display);
    myinit();
    glutMainLoop();

    return 0;
}
```

Program 6.1 Rotasi terhadap sembarang titik

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)

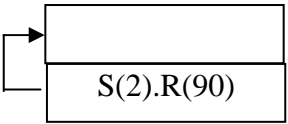
- 3. Rotate terhadap x by 90 derajat

```
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);
```

S(2).R(90)Tr(a)
S(2).R(90)

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

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

S(2).R(90)Tr(a)
S(2).R(90)

- 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.

Program 6.2 menunjukkan contoh penggunaan glPushMatrix() dan glPopMatrix().

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0,0.0,1.0);
    glLoadIdentity();
    glPointSize(6);

    //Point pertama
    glBegin(GL_POINTS);
    glVertex2f(15,15);
    glEnd();

    glPushMatrix();

    glScalef(2,1,1);

    //Point pertama yang di
    Scale      glBegin(GL_POINTS);
    glVertex2f(15,15);      glEnd();

    //Point kedua yang sudah di
    Scale      glColor3f(0.0,1.0,0.0);
    glBegin(GL_POINTS);
    glVertex2f(10,25);      glEnd();

    glPopMatrix();

    //Point kedua tanpa Scale
    glBegin(GL_POINTS);
    glVertex2f(10,25);
    glEnd();

    glRotatef(10, 0, 0, 1);

    //Point pertama dengan
    rotate
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POINTS);
    glVertex2f(15,15);      glEnd();

    glPushMatrix();

    glTranslatef(5,0,0);
```

```

        //Point pertama dengan rotate dulu dan setelah itu di
translate    glBegin(GL_POINTS);    glVertex2f(15,15);
glEnd();

        glFlush();
    }
void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 50.0, 0.0, 50.0);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(1.0, 1.0, 1.0, 1.0);
    glColor3f(0.0, 0.0, 0.0);
}
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Transform");
    glutDisplayFunc(display);
    myinit();
    glutMainLoop();

    return 0;
}

```

Program 6.2 Contoh penggunaan glPushMatrix() dan glPopMatrix().

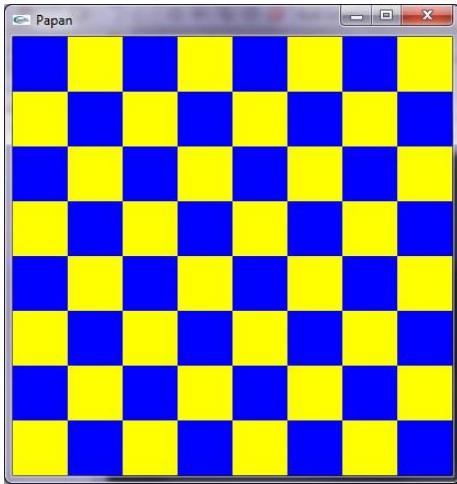
III. Percobaan

1. Untuk program 6.1, modifikasi parameter fungsi transformasi, lalu amati perubahan tampilannya. Amati juga efek urutan pemanggilan fungsi transformasi.
2. Berdasarkan program 6.1, buat sebuah program yang melakukan Scaling kotak kedua terhadap titik tengah kotak kedua, bukan terhadap titik pusat koordinat.
3. Untuk program 6.2, modifikasi transformasi dan push/pop matrix, lalu amati perubahan tampilannya.
4. Untuk program 6.2, hapus semua glPushMatrix() dan glPopMatrix, ganti dengan glLoadIdentity() sedemikian rupa sehingga tampilan program tetap sama. Posisi vertex-vertexnya harus tetap, sedangkan nilai transformasi boleh diubah.

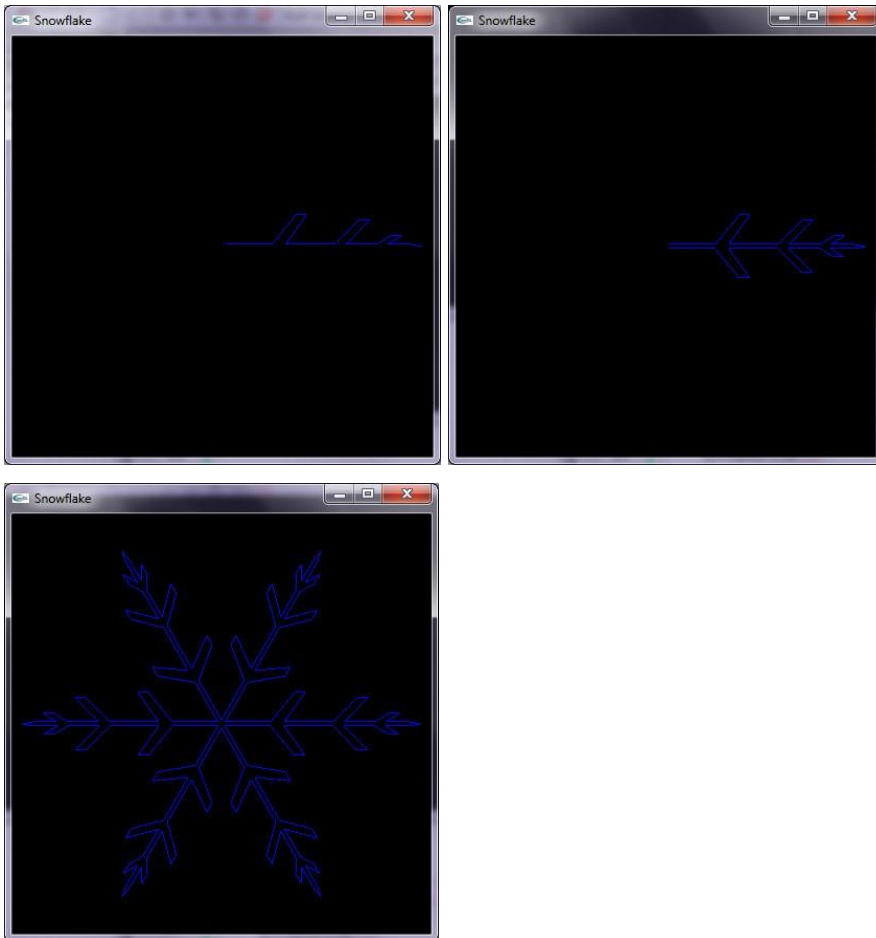
IV. Tugas

Tugas modul 6 sama dengan tugas modul 5. Bedanya, gunakan glPushMatrix() dan glPopMatrix:

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

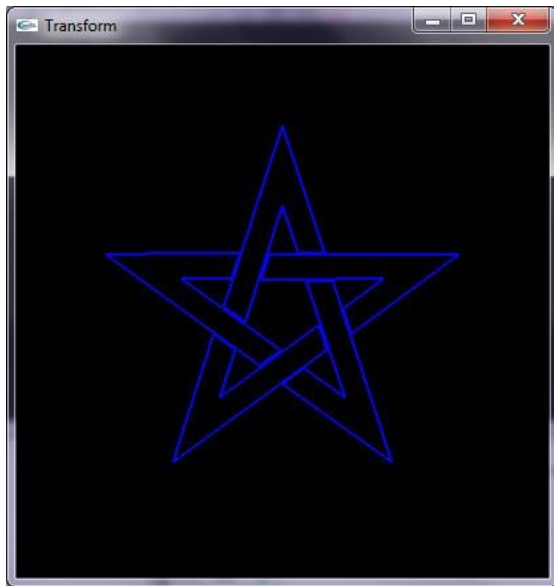


2. Buat snow flake (bunga salju) berikut menggunakan transformasi.



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.

V. Referensi

1. Edward Angel, "Interactive Computer Graphics Sixth Edition", Pearson, 2012, ch 3, p 115 – 194
2. F. S. Hill, Jr., Stephen M. Kelley, "Computer Graphics Using OpenGL Third Edition", Prentice Hall, 2007, ch 3, p 115 – 194
3. Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis, "OpenGL Programming Guide Fifth Edition", Addison-Wesley, 2006, ch 3, p 103-163