

Modul 3

Transformasi Geometri

I. Tugas Pendahuluan

1. Jelaskan dengan singkat apa yang disebut sebagai translasi, scaling, dan rotasi dalam transformasi geometri!
2. Gambarkan dengan tangan grafik transformasi titik (2, 3) yang ditranslasi sejauh (3, -4)!
3. Gambarkan dengan tangan grafik transformasi titik (3, 3) yang dirotasi sejauh 90 derajat terhadap sumbu koordinat!
4. Gambarkan dengan tangan grafik transformasi titik (3, 2) yang di-scaling sebesar (2, 1.5) terhadap sumbu koordinat!

II. Pengantar

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 struktur geometri, sehingga disebut transformasi geometri. Terdapat banyak jenis operasi transformasi geometri: translasi, refleksi, rotasi, scaling, shearing.

OpenGL memiliki 3 perintah transformasi:

- a. `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)`.
- b. `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)`.
- c. `glRotated(alpha, i, j, k)`: melakukan rotasi sebesar α . α 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 diinginkan. Contoh: jika ingin merotasi obyek sebesar 90 derajat pada sumbu x , maka perintahnya adalah: `glRotated(90.0, 1, 0, 0)`.

*huruf d diakhir perintah transformasi merupakan kependekan dari double; yang berarti argumen $a, b, c, d, e, f, \alpha, i, j$, dan k adalah angka pecahan presisi ganda (double). Selain double (d), pilihan jenis argumen yang dapat digunakan adalah: i (integer/bilangan bulat) dan f (float/bilangan pecahan presisi tunggal), contoh: `glTranslatetf(m, n, o)`.

Proses transformasi di OpenGL bersifat melekat: sekali sebuah perintah transformasi dieksekusi, perintah tersebut akan selalu dilakukan untuk semua perintah yang ada dibawahnya. Contoh: jika pada program terdapat perintah `glTranslated(10.0, 0.0, 0.0)` pada baris ke 25, maka perintah-perintah `glVertex` pada baris ke 26 dan seterusnya akan selalu ditranslasi pada sumbu x sejauh 10. Program 5.1 menunjukkan contoh translasi.

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

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

    //translasi ke 20, 20
    glTranslated(20.0, 20.0, 0);
    glRecti(0,0, 10, 10);

    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 5.1 Translasi Program

5.2 menunjukkan contoh Scaling.

```

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0,0.0,0.5);

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

    //Scaling kotak yang digambar di ke 20, 20 sebesar 1.5
    kali    glScaled(1.5, 1.5, 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 5.2 Scaling

Yang perlu diperhatikan disini adalah bahwa proses scaling dilakukan dari sumbu koordinat yang terletak di sudut kiri bawah jendela. Hal inilah yang menyebabkan tampilan pada program 5.2 diatas terlihat cenderung lebih ke kanan atas jendela.

Program 5.3 menunjukkan contoh rotasi.

```

void display()
{
    glClear(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 sebesar 15 derajat terhadap sumbu
koordinat(titik kiri bawah)   glRotated(15, 0, 0, 1.0);
    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 5.3 Rotasi

Yang perlu diperhatikan dari program 5.3 adalah bahwa rotasi dilakukan terhadap titik koordinat yang terletak pada ujung kiri bawah jendela. Supaya rotasi terjadi pada titik tengah obyek, perlu dilakukan kombinasi perintah transformasi.

Kombinasi transformasi

Operasi-operasi transformasi yang berbeda dapat dikombinasikan. Contoh: jika ingin melakukan operasi-operasi berikut pada sebuah obyek:

- translasi sebesar (3, 4)
- lalu rotasi sebesar 30° pada sumbu z
- lalu skala sebesar (1.5, 1.5)
- lalu translasi lagi sebesar (0, 0.5)
- dan terakhir rotasi sebesar 45° maka perintah-perintahnya adalah:

```

glRotated(45, 0, 0, 1); glTranslated(0.0, 0.5, 0.0); glScaled(1.5, 1.5, 0.0); glRotated(30.0,
0, 0, 1); glTranslated(3.0, 4.0, 0.0);

```

Yang perlu diperhatikan disini adalah urutan perintah. OpenGL melakukan perintah transformasi mulai dari yang paling bawah.

Perlu diingat pula bahwa karena pada dasarnya operasi transformasi dilakukan dengan menggunakan operasi perkalian matrix yang tidak bersifat komutatif ($AB \neq BA$), maka urutan operasi transformasi sangat berpengaruh.

Program 5.4 menunjukkan contoh kombinasi transformasi

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

    // Gambar kotak pertama di kuadran 1
    glRecti(0,0, 2, 2);

    // Gambar kotak kedua
    glRotated(45,
0, 0, 1);    glTranslated(0.0, 0.5, 0.0);
    glScaled(1.5, 1.5, 0.0);
    glRotated(30.0, 0, 0, 1);
    glTranslated(3.0, 4.0, 0.0);

    glRecti(0,0, 2, 2);

    glLoadIdentity();

    //Gambar sumbu koordinat
    glColor3f(1.0, 1.0, 0.0);
    glBegin(GL_LINES);    glVertex3f(-
10.0,0.0,0.0);

    glVertex3f(10.0,0.0,0.0);
    glVertex3f(0.0,-10.0,0.0);
    glVertex3f(0.0,10.0,0.0);
    glEnd();

    glFlush();
}
void myinit()
{
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-10.0, 10.0, -10.0, 10.0);
    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 5.4 Kombinasi Transformasi

Program 5.4 memiliki baris perintah `glLoadIdentity()` sebelum menggambar sumbu koordinat. Hal ini disebabkan, tiap kali perintah transformasi diberikan semua baris perintah vertex di bawahnya akan terpengaruh. Perintah `glLoadIdentity()` digunakan untuk menetralkan efek ini; sehingga, pemanggilan perintah vertex di bawahnya akan kembali normal seperti sebelum terjadi transformasi.

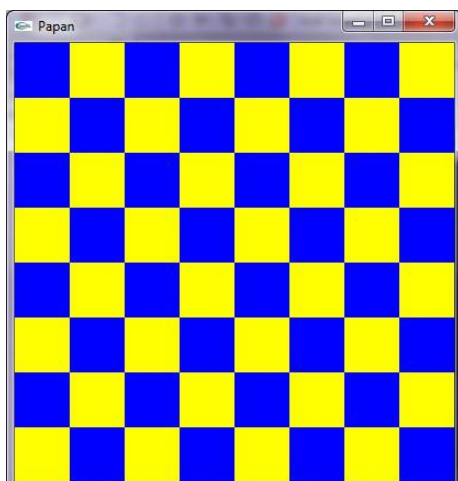
III. Percobaan

1. Untuk program 5.1, modifikasi parameter `glTranslated`, lalu amati perubahan tampilannya. Kombinasikan dengan memodifikasi parameter pada `glRecti`; amati perubahannya juga.
2. Untuk program 5.2, modifikasi parameter `glScaled`, lalu amati perubahan tampilannya. Kombinasikan dengan memodifikasi parameter pada `glRecti`; amati perubahannya juga.
3. Untuk program 5.3, modifikasi parameter `glRotated`, lalu amati perubahan tampilannya. Kombinasikan dengan memodifikasi parameter pada `glRecti`; amati perubahannya juga.
4. Untuk program 5.4, coba hapus baris perintah `glLoadIdentity()`, lalu amati perubahan tampilannya.
5. Untuk program 5.4, modifikasi parameter fungsi transformasi, lalu amati perubahan tampilannya. Amati juga efek urutan pemanggilan fungsi transformasi.
6. Untuk program 5.4, tambahkan/bermain-mainlah dengan perintah `glRecti()`, `glLoadIdentity()`, dan transformasi. Amati perubahan-perubahan yang terjadi hingga paham cara kerja masing-masing perintah.

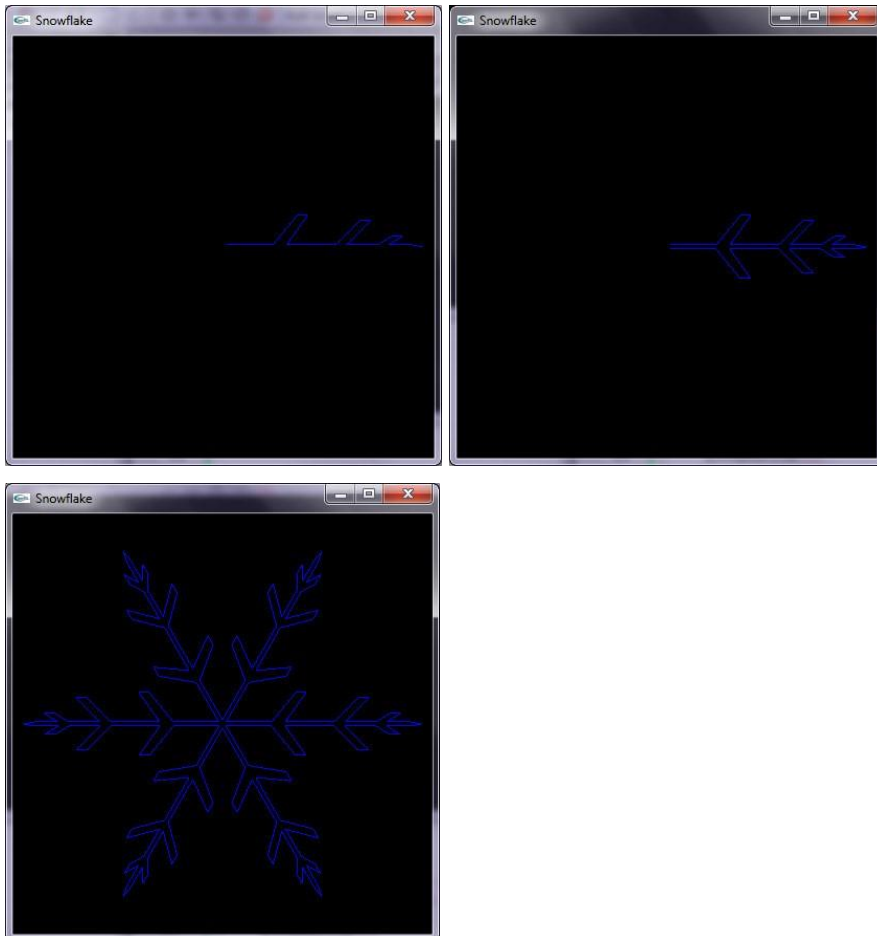
IV. Tugas

Selesaikan soal-soal berikut ini:

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

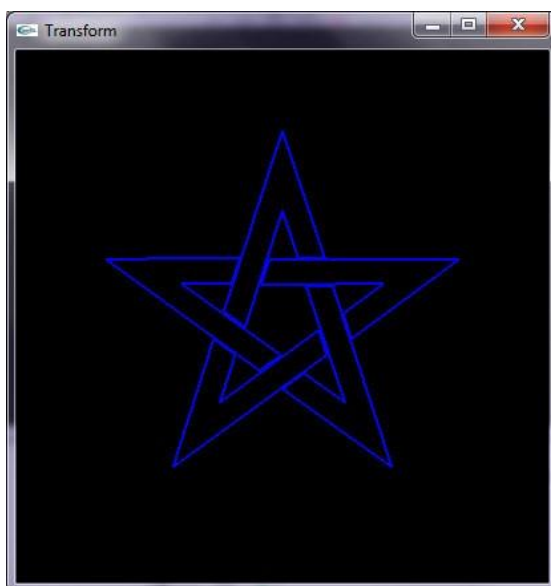


2. Buat snow flake (bunga salju) berikut menggunakan transformasi.



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