

Modul Praktikum 4

Transformasi 2D & 3D

A. Tujuan

1. Mahasiswa mampu menjelaskan tentang konsep transformasi.
2. Mahasiswa mampu menjelaskan tentang perhitungan matriks transformasi 2D.
3. Mahasiswa mampu menjelaskan tentang perhitungan matriks transformasi 3D.
4. Mahasiswa mampu menerapkan transformasi 2D dan 3D dengan OpenGL.

B. Dasar Teori

Transformasi berarti mengubah posisi. Transformasi dasar dalam komputer grafis diantaranya translasi, scaling, rotasi, shear untuk 2D dan 3D.

Translasi

Translasi berarti merubah posisi obyek dari koordinat yang satu ke koordinat yang lain. Dilakukan dengan menambahkan jarak translasi (t_x , t_y , t_z) pada posisi awal (x , y , z) untuk memindahkan benda ke posisi baru (x' , y' , z'). Jarak translasi (t_x , t_y , t_z) disebut juga vektor translasi atau vektor perpindahan. Rumus Translasi 3D:

$$\begin{aligned}x' &= x + t_x \\y' &= y + t_y \\z' &= z + t_z\end{aligned}$$

Atau bila direpresentasikan dengan matriks, translasi 3D dapat dirumuskan: $P' = T \cdot P$
Apabila dituliskan dalam koordinat homogen:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Scaling

Scaling berarti mengubah ukuran obyek. Dilakukan dengan mengalikan factor skala (s_x , s_y , s_z) pada posisi awal (x , y , z) untuk menghasilkan ukuran baru di koordinat (x' , y' , z'). Rumus Scaling 3D:

$$x' = \dots$$

$$y' = \dots$$

$$z' = \dots$$

Bila direpresentasikan dengan matriks, scaling 3D dapat dirumuskan:

$$P' = S \cdot P$$

Apabila dituliskan dalam koordinat homogen:

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rotasi

Rotasi berarti mengubah posisi terhadap jalur melingkar pada bidang x-y-z. Dilakukan dengan menentukan sudut rotasi dan titik rotasi (rotation point / pivot point) (x, y, z) untuk menghasilkan posisi baru pada koordinat (x', y', z'). Bila $\theta > 0$: rotasi berlawanan jarum jam. Bila $\theta < 0$: rotasi searah jarum jam.

Rumus rotasi terhadap sumbu-Z:

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

$$z' = z$$

Atau $P' = R_z(\theta) \cdot P$.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rumus rotasi terhadap sumbu-X:

$$x' = x$$

$$y' = y \cos \theta - z \sin \theta$$

$$z' = y \sin \theta + z \cos \theta$$

Atau $P' = R_x(\theta) \cdot P$.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Rumus rotasi terhadap sumbu-Y:

$$x' = x \cos \theta + y \sin \theta$$

$$y' = -x \sin \theta + y \cos \theta$$

$$z' = z$$

Atau $\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ -\sin \theta & 0 & \cos \theta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

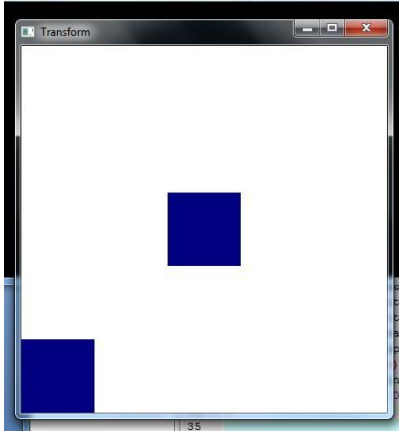
C. Praktikum

Translasi

Jalankan kode program berikut:

```
main.cpp
1  #include<windows.h>
2  #include<gl/glut.h>
3  void display() //Translasi
4  {
5      glClear(GL_COLOR_BUFFER_BIT);
6      glColor3f(0.0,0.0,0.5);
7      // Gambar kotak pertama di sudut kiri bawah
8      glRecti(0,0, 10, 10);
9      //translasi ke 20, 20
10     glTranslated(20.0, 20.0, 0);
11     glRecti(0,0, 10, 10);
12     glFlush();
13 }
14 void myinit()
15 {
16     glMatrixMode(GL_PROJECTION);
17     glLoadIdentity();
18     gluOrtho2D(0.0,50.0,0.0,50.0);
19     glMatrixMode(GL_MODELVIEW);
20     glClearColor(1.0,1.0,1.0,1.0);
21     glColor3f(0.0,0.0,0.0);
22 }
23 int main(int argc, char* argv[])
24 {
25     glutInit(&argc,argv);
26     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
27     glutInitWindowSize(400,400);
28     glutInitWindowPosition(100,100);
29     glutCreateWindow("Transform");
30     glutDisplayFunc(display);
31     myinit();
32     glutMainLoop();
33     return 0;
34 }
```

Hasil proses *compile and run* pada *list* program diatas, ditunjukkan pada Gambar berikut.

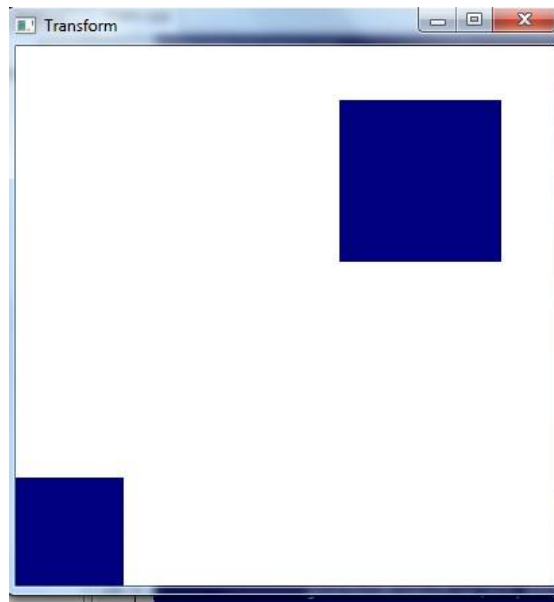


Scaling

Jalankan kode program berikut:

```
main.cpp
1  #include<windows.h>
2  #include<gl/glut.h>
3  void display() // Scaling
4  {
5      glClear(GL_COLOR_BUFFER_BIT);
6      glColor3f(0.0,0.0,0.5);
7
8      // Gambar kotak pertama di sudut kiri bawah
9      glRecti(0,0, 10, 10);
10
11     //Scaling kotak yang digambar di ke 20, 20 sebesar 1.5 kali
12     glScaled(1.5, 1.5, 0.0);
13     glRecti(20,20, 30, 30);
14     glFlush();
15 }
16 void myinit()
17 {
18     glMatrixMode(GL_PROJECTION);
19     glLoadIdentity();
20     gluOrtho2D(0.0,50.0,0.0,50.0);
21     glMatrixMode(GL_MODELVIEW);
22     glClearColor(1.0,1.0,1.0,1.0);
23     glColor3f(0.0,0.0,0.0);
24 }
25 int main(int argc, char* argv[])
26 {
27     glutInit(&argc,argv);
28     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
29     glutInitWindowSize(400,400);
30     glutInitWindowPosition(100,100);
31     glutCreateWindow("Transform");
32     glutDisplayFunc(display);
33     myinit();
34     glutMainLoop();
35     return 0;
36 }
37
```

Hasilnya :

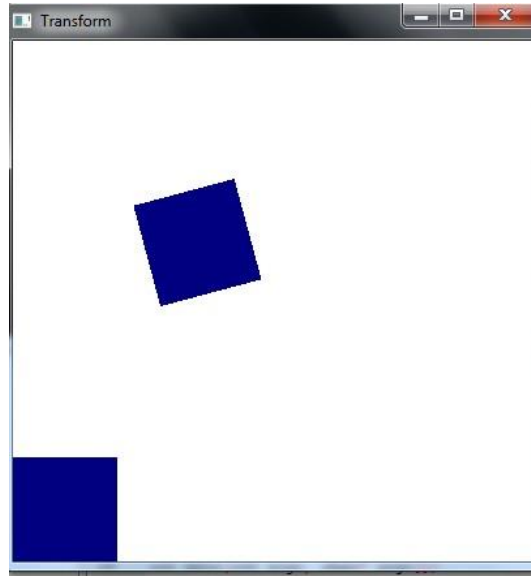


Rotasi

Jalankan kode program berikut:

```
[*] main.cpp
1  #include<windows.h>
2  #include<gl/glut.h>
3  void display() //Rotasi
4  {
5      glClear(GL_COLOR_BUFFER_BIT);
6      glColor3f(0.0,0.0,0.5);
7
8      // Gambar kotak pertama di sudut kiri bawah
9      glRecti(0,0, 10, 10);
10
11     //rotasi kotak kedua sebesar 15 derajat terhadap sumbu koordinat (titik kiri bawah)
12     glRotated(15, 0, 0, 1.0);
13     glRecti(20,20, 30, 30);
14
15     glFlush();
16 }
17 void myinit()
18 {
19     glMatrixMode(GL_PROJECTION);
20     glLoadIdentity();
21     gluOrtho2D(0.0,50.0,0.0,50.0);
22     glMatrixMode(GL_MODELVIEW);
23     glClearColor(1.0,1.0,1.0,1.0);
24     glColor3f(0.0,0.0,0.0);
25 }
26
27 int main(int argc, char* argv[])
28 {
29
30     glutInit(&argc,argv);
31     glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
32     glutInitWindowSize(400,400);
33     glutInitWindowPosition(100,100);
34     glutCreateWindow("Transform");
35     glutDisplayFunc(display);
36     myinit();
37     glutMainLoop();
38
39     return 0;
40 }
```

Hasilnya adalah seperti berikut.



TUGAS MINGGUAN

1. Untuk NPM (Ganjil) rubah bentuk objek menjadi segitiga;
2. Untuk NPM (Genap) rubah bentuk objek menjadi persegi panjang;
3. Lakukan Translasi, lalu coba rubah koordinat translasi dari yang semulanya (20.0, 20.0, 0) dengan menggunakan koordinat yang lain;
4. Lakukan Scaling, lalu coba rubah ukuran menjadi 2 kali;
5. Lakukan Rotasi, lalu coba lakukan rotasi menjadi 20 derajat;
6. Rubahlah nama windownya dengan (NPM & Nama Masing-masing)
7. Buat laporannya dalam format pdf, lengkap dengan identitas dan cover.
8. Waktu pengerjaan 1 Minggu.