

### Praktikum 3 Translasi dan Rotasi

Tujuan :

1. Mengetahui koordinat melalui proses transformasi

Pelaksanaan :

1. Membuat fungsi Translasi
2. Membuat fungsi Rotasi
3. Penggunaan Stack

Sebelumnya, praktikum sebelumnya terdapat perintah perintah object primitif.

Berikut rangkumannya :

Perintah	Arti	Keterangan
glVertex2i(x,y);	Lokasi titik berada di (x,y)	Tipe argumennya adalah integer dan 2 dimensi yaitu x dan y
glVertex2f(x,y);	Lokasi titik berada di (x,y)	Tipe argumennya adalah float dan 2 dimensi yaitu x dan y
glVertex3i(x,y,z);	Lokasi titik berada di (x,y,z)	Tipe argumennya adalah integer dan 3 dimensi yaitu x, y dan z
glVertex3f(x,y,z);	Lokasi titik berada di (x,y,z)	Tipe argumennya adalah float dan 3 dimensi yaitu x, y dan z
glClearColor(R, G, B, $\alpha$ );	Warna latar belakang	Empat komponen warna yaitu Red, Green, Blue dan alpha
glColor3f(R, G, B);	Warna latar muka (pena)	Tiga komponen warna yaitu Red, Green dan Blue
glColor4f(R, G, B, $\alpha$ );	Warna latar muka (pena)	Empat komponen warna yaitu Red, Green, Blue dan alpha
glPointSize(k);	Ukuran titik k piksel	Besar kecilnya ukuran titik tergantung pada k (integer)
glBegin(GL_POINTS);	Titik	Objek primitif
glBegin(GL_LINES);	Garis	Objek primitif
glBegin(GL_LINE_STRIP);	Poligaris	Objek primitif
glBegin(GL_LINE_LOOP);	Poligaris tertutup (polygon)	Objek primitif
glBegin(GL_TRIANGLES);	Segitiga	Objek primitif
glBegin(GL_TRIANGLE_STRIP);	Segitiga	Objek primitif
glBegin(GL_TRIANGLE_FAN);	Segitiga	Objek primitif
glBegin(GL_QUADS);	Segiempat	Objek primitif
glBegin(GL_QUAD_STRIP);	Segiempat	Objek primitif
glBegin(GL_POLYGON);	Poligon	Objek primitif
glBegin(GL_LINE_STIPPLE);	Garis putus-putus	Objek primitif
glBegin(GL_POLY_STIPPLE);	Poligon dengan pola tertentu	Objek primitif
glRect(GLint x1, GLint y1, GLint x2, GLint y2);	Segiempat siku-siku	Objek primitif dan ukuran segiempat ditentukan oleh dua titik yaitu (x1,y1) dan (x2,y2)
glEnd();	Akhir perintah OpenGL	-

## Translasi Objek

Sekarang kita akan belajar memindahkan objek yang telah kita dibuat, dengan mengubah model pandangan atau sumbu utama koordinat, fungsi yang digunakan untuk mengubah sumbu utama adalah `glTranslatef(sumbu x, sumbu y, sumbu z)`.

```
#include <gl/glut.h>
void SegiEmpat(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glTranslatef(-0.50, 0.50, 0);
    glColor3f(1.0f, 0.0f, 0.0f);
    glRectf(-0.18, 0.18, 0.18, -0.18);
    glFlush();
}
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE |
    GLUT_RGBA); glutCreateWindow("Segi Empat");
    glutDisplayFunc(SegiEmpat);
    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
    glutMainLoop();
    return 0;
}
```

Kita baru saja memindahkan segiempat yang di buat dari sumbu utamanya. Pada posisi sebenarnya segiempat ada di koordinat ( -18, 18, 18, -18 ) yang berarti titik diagonal pertama ada di  $x=-18$ ,  $y=18$ , dan titik diagonal kedua ada di  $x=18$ ,  $y=-18$  dengan sumbu utama di  $(x=0,y=0,z=0)$  dengan member fungsi `glTranslatef(-0.50, 0.50, 0)`; maka sumbu utama bergeser ke titik  $(x=-50, y=50, z=0)$ .

## Rotasi Objek

Setelah belajar memindahkan objek, sekarang kita akan merotasi objek, perintah yang digunakan adalah `glRotated(rotasi, x,y, 1.0)`

```
#include <gl/glut.h>
void Segitiga(void)
{
    glTranslatef( 0.25, -0.25, 0);
    glRotated(60.0, 0.0, 0.0, 1.0);
    glBegin(GL_TRIANGLES);
        glColor3f(1.0f, 0.0f, 0.0f); glVertex3f( -0.05,-0.05, 0.00);
        glColor3f(0.0f, 1.0f, 0.0f); glVertex3f( 0.15,-0.05, 0.00);
        glColor3f(0.0f, 0.0f, 1.0f); glVertex3f( -0.05, 0.05, 0.00);
    glEnd();
}
int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
```

```

glutInitWindowSize(640,480);
glutInitDisplayMode(GLUT_SINGLE |
GLUT_RGBA); glutCreateWindow("Segi Empat");
glutDisplayFunc(Segitiga);
glClearColor(0.0f, 0.0f, 1.0f,
1.0f); glutMainLoop();
return 0;
}

```

glRotated(60.0, 0.0, 0.0, 1.0); berfungsi merotasi objek, 60 merupakan nilai derajat kemiringan objek yang dirotasi, nilai 0 pertama merupakan sumbu putar x, dan nilai 0 kedua merupakan sumbu putar y terhadap objek, kita gunakan sumbu (0,0) memutar terhadap objek, dan nilai 1 terakhir merupakan opsi untuk skala, bila bernilai 0 maka fungsi rotate berubah menjadi skala, jadi harus bernilai 1 untuk rotasi.

### Stack object

Untuk membuat banyak objek sekaligus menggunakan stack, perintah yang digunakan adalah glPushMatrix(); & glPopMatrix(); fungsinya adalah mengelompokkan perintah, sehingga fungsi yang digunakan oleh objek sebelumnya tidak terbawa ke objek berikutnya.

```

#include <gl/glut.h>
#include <gl/gl.h>
#include <math.h>
void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
        glLineWidth(2.0f);
        glBegin(GL_LINES);
            glColor3f(1.0f, 1.0f, 1.0f);
            glVertex3f( 0.00, 0.20, 0.0);
            glVertex3f( 0.00, -0.20, 0.0);
        glEnd();
    glPopMatrix();

    glPushMatrix();
        glPointSize(5.0f);
        glTranslatef(0.35, 0.35,
0.0); glBegin(GL_POINTS);
            glColor3f(1.0f, 1.0f, 1.0f);
            glVertex3f( 0.25, 0.25, 0.0);
        glEnd();
    glPopMatrix();

    glPushMatrix();
        glTranslatef( 0.50,0.50,0.00);
        glBegin(GL_TRIANGLE_STRIP);
            glColor3f(1.0f, 0.0f, 0.0f); glVertex3f( -0.05,-0.05, 0.00);
            glColor3f(0.0f, 1.0f, 0.0f); glVertex3f( 0.15,-0.05, 0.00);
            glColor3f(0.0f, 0.0f, 1.0f); glVertex3f( -0.05, 0.05, 0.00);
        glEnd();
    glPopMatrix();
}

```

```

        glColor3f(1.0f, 1.0f, 0.0f); glVertex3f( 0.15, 0.05,
        0.00); glEnd();
    glPopMatrix();

    glPushMatrix();
    #define PI 3.1415926535898
    glBegin(GL_LINE_LOOP);
        GLint circle_points =
        100; int i;
        float angle;
        for (i = 0; i < circle_points; i++) {
            angle = 2*PI*i/circle_points;
            glVertex2f(cos(angle), sin(angle));
        }
    glEnd();
    glPopMatrix();

    glFlush();
}

int main(int argc, char* argv[])
{
    glutInit(&argc, argv);
    glutInitWindowSize(640,480);
    glutInitDisplayMode(GLUT_SINGLE |
    GLUT_RGBA); glutCreateWindow("Simple");
    glutDisplayFunc(RenderScene);
    glClearColor(0.0f, 0.0f, 1.0f, 1.0f);
    glutMainLoop();
    return 0;
}

```

Pertanyaan :

1. Tunjukkan bagian mana yang disebut stack dari kode di atas ?
2. Apa Fungsi Push dan Pop yang ada pada stack? Mengapa harus menggunakan stack?
3. Apa perbedaan rotasi/translasi di dalam stack dan diluar stack ?

Tugas :

1. Buatlah sebuah lingkaran.
2. Buatlah sebuah mobil 2D yang memanfaatkan object2 primitif dan transformasi (translasi dan rotasi). Penggunaan stack disarankan!

**Catatan :** buatlah mobil dengan rodanya. Detail mobil dan objek-objek tambahan akan mendapatkan nilai tambah.