

PRAKTIKUM GRAFIK KOMPUTER
TUGAS PERTEMUAN 4 – TRANSFORMASI 2D & 3D

“Disusun Untuk Memenuhi Tugas Mata Kuliah Praktikum Grafik Komputer”

Dosen pengampu : Sri Rahayu, S.T. M.Kom



Disusun Oleh :

Yoga Agustiansyah

2206050

Teknik Informatika B

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI GARUT

2023

PENDAHULUAN

Transformasi adalah teknik penting dalam grafika komputer yang digunakan untuk mengubah tampilan objek-objek grafis. Transformasi 2D mencakup translasi (pergeseran), scaling (perbesaran/persempitan), dan rotasi objek dalam dua dimensi, sedangkan transformasi 3D melibatkan perubahan dalam tiga dimensi.

Transformasi 2D adalah dasar dari banyak aplikasi grafis, termasuk permainan, animasi, dan aplikasi desain. Transformasi 2D memungkinkan penggunaan objek-objek dalam dua dimensi untuk menciptakan tampilan yang dinamis. Transformasi ini dapat digunakan untuk mengubah ukuran, posisi, atau orientasi objek.

Sementara itu, transformasi 3D adalah penting dalam pembuatan grafis 3D yang kompleks, seperti perangkat lunak modeling 3D, simulasi, dan permainan video. Dalam transformasi 3D, objek dapat dipindahkan, diubah skala, dan dirotasi dalam tiga dimensi, menciptakan tampilan yang realistis.

Dalam praktikum ini, kami akan membandingkan program-program OpenGL yang menerapkan transformasi 2D dan 3D.

PEMBAHASAN

A. Translasi

1) Source Code

```
#include <GL/glut.h>

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRecti(0, 0, 20, 5);
    glTranslated(10.0, 10.0, 0);
    glRecti(0, 0, 20, 5);
    glFlush();
}

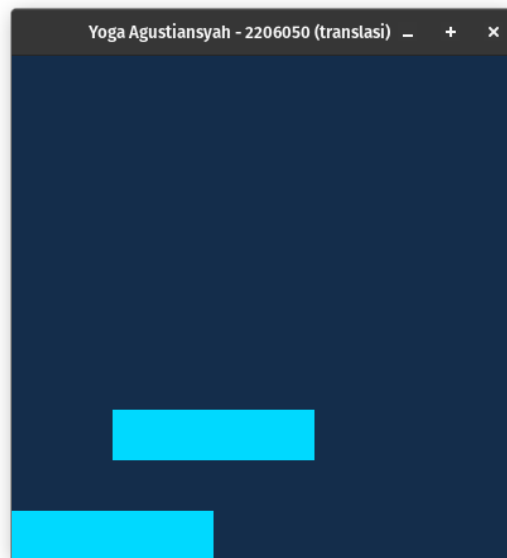
void myInit() {
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 50, 0, 50);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glColor3f(0, 0, 0);
}

void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

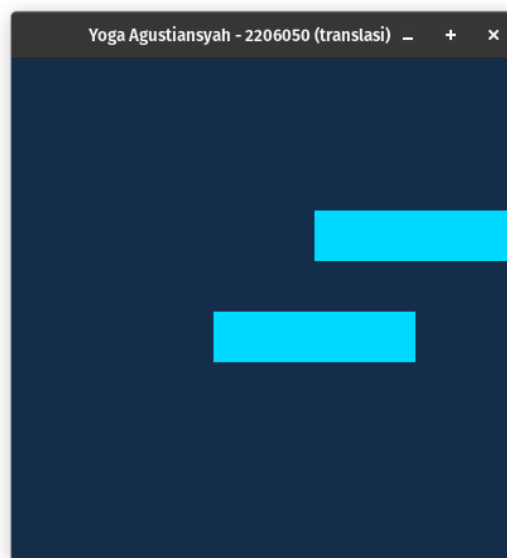
int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050
(translasi)");
    glutDisplayFunc(display);
    myInit();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}
```

2) Output

Saat program dijalankan



Saat di translasi satukali



3) Penjelasan

Program ini membuat jendela sederhana dengan dua persegi panjang dengan ukuran panjang 20 pixel dan lebar 5 pixel yang diberi efek translasi saat tombol kiri mouse ditekan. Saat menekan klik kiri pada mouse di window output, kedua objek akan digambar ulang dengan posisi X dan Y masing masing ditambah 10 pixel, hal ini membuat efek objek seakan akan berpindah secara langsung, padahal sebenarnya objek hanya digambar ulang di titik koordinat yang berbeda.

B. Eskalasi

1) Source Code

```
#include <GL/glut.h>

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRecti(0, 0, 10, 5);
    glScaled(2, 2, 0);
    glRecti(5, 5, 15, 10);
    glFlush();
}

void myInit() {
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 50, 0, 50);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glColor3f(0, 0, 0);
}

void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

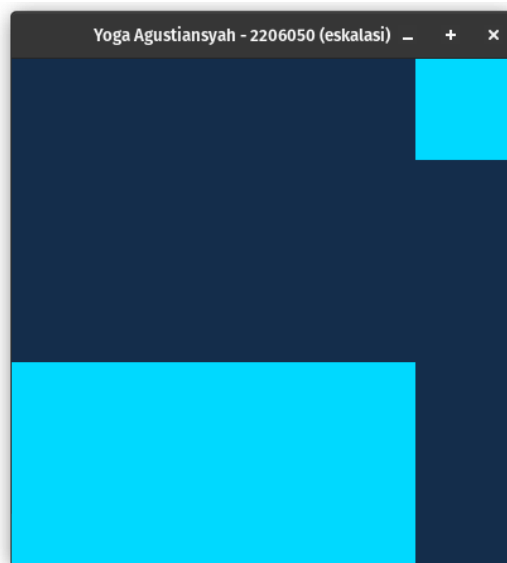
int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050
(eskalasi)");
    glutDisplayFunc(display);
    myInit();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}
```

2) Output

Saat program pertama kali dijalankan



Saat di eskalasi satu kali



3) Penjelasan

Program ini membuat jendela sederhana dengan dua persegi panjang dengan ukuran panjang 10 pixel dan lebar 5 pixel yang diberi efek skalasi. Setelah menggambar persegi panjang pertama, objek kedua diberi efek skalasi dengan faktor x2 pada kedua dimensi X dan Y, sehingga menjadi dua kali lebih besar daripada yang pertama. Setiap menekan klik kiri pada mouse di area window output, kedua objek akan di eskalasi sebesar 2x sehingga ukurannya dapat terus membesar.

1) Source Code

```
#include <GL/glut.h>

void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.850, 1);
    glRecti(0, 0, 20, 5);
    glRotated(20, 0, 0, 1);
    glRecti(20, 20, 40, 15);
    glFlush();
}

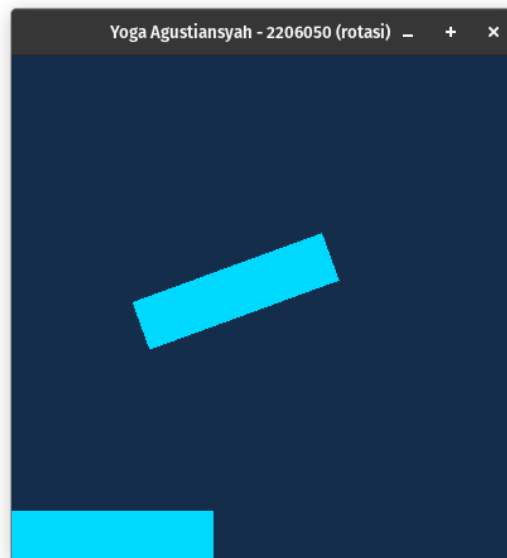
void myInit() {
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 50, 0, 50);
    glMatrixMode(GL_MODELVIEW);
    glClearColor(0.078, 0.178, 0.294, 1.0);
    glColor3f(0, 0, 0);
}

void mouseClicked(int button, int state, int x, int y) {
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        // Memanggil fuction display saat klik kiri mouse
        display();
        // Meminta glut untuk merender ulang tampilan
        glutPostRedisplay();
    }
}

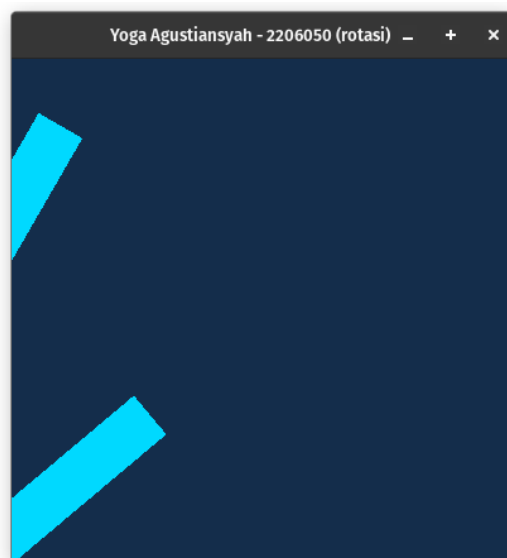
int main(int argc, char*argv[]){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(400, 400);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Yoga Agustiansyah - 2206050
(rotasi)");
    glutDisplayFunc(display);
    myInit();
    glutMouseFunc(mouseClick);
    glutMainLoop();
    return 0;
}
```

2) Output

Saat program pertama kali dijalankan



saat dirotasi sekali



3) Penjelasan

Program ini membuat objek sederhana berbentuk dua persegi panjang dengan ukuran panjang 20 pixel dan lebar 5 pixel. Objek kedua mengalami rotasi sebesar 20 derajat pada sumbu Z, setiap kali mengklik kiri mouse di area window output, akan memberikan efek rotasi pada kedua objek tersebut.

D. Penjelasan Fungsi Tambahan

```
void mouseClicked(int button, int state, int x, int y) {  
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {  
        display();  
        glutPostRedisplay();  
    }  
}
```

Fungsi `mouseClick` ini adalah sebuah callback function yang akan dipanggil oleh GLUT ketika ada sebuah interaksi klik mouse yang terjadi pada window OpenGL. Fungsi ini memiliki empat parameter:

1. `button`: Parameter ini mengidentifikasi tombol mouse yang ditekan atau dilepas. Dalam kasus ini, kita memeriksa apakah tombol yang ditekan adalah tombol kiri (`GLUT_LEFT_BUTTON`).
2. `state`: Parameter ini menunjukkan apakah tombol mouse ditekan atau dilepas. Jika `state` sama dengan `GLUT_DOWN`, itu berarti tombol mouse ditekan.
3. `x` dan `y`: Parameter ini menyimpan koordinat lokasi di mana klik mouse terjadi pada window.

Dalam function ini terdapat beberapa proses, yaitu

- Kode pertama memeriksa apakah tombol mouse yang ditekan adalah tombol kiri (`GLUT_LEFT_BUTTON`) dan apakah tombol mouse dalam keadaan "ditekan" (`GLUT_DOWN`). Hal ini dilakukan untuk memastikan bahwa kita hanya menangani klik mouse kiri saat tombol tersebut ditekan.
- Jika kondisi di atas terpenuhi, maka fungsi `display()` dipanggil. Ini akan menggambar ulang tampilan dalam window OpenGL.
- Selanjutnya, dengan menggunakan `glutPostRedisplay()`, kita meminta GLUT untuk merender ulang tampilan. Ini diperlukan untuk memastikan bahwa perubahan yang dilakukan dalam fungsi `display()` diterapkan secara visual ke tampilan OpenGL.

penambahan function `mouseClick` ini dikarenakan saat menjalankan program di lingkungan Linux, tidak terjadi perubahan apapun ketika window output di klik, sedangkan saat menggunakan Windows, semua program mendeteksi input saat window output di klik, sehingga dapat melakukan redisplay secara otomatis.

Dengan demikian, fungsi `mouseClick` ini digunakan sebagai pemicu untuk melakukan redisplay setelah objek di transformasi. Saat kita mengklik mouse, objek yang kita gambar akan ditransformasi (sesuai dengan kode dalam fungsi `display()`) dan tampilan diperbarui untuk mencerminkan perubahan tersebut. Hal ini membuat program lebih interaktif dan responsif terhadap input pengguna.

KESIMPULAN

Dalam praktikum ini, kita telah mempelajari konsep dasar tentang transformasi 2D dan 3D dalam grafik komputer. Transformasi adalah teknik penting yang digunakan untuk mengubah posisi, rotasi, atau skala objek dalam ruang dua dimensi maupun tiga dimensi.

Transformasi translasi memungkinkan kita untuk menggeser objek dari satu posisi ke posisi lain dalam koordinat grafis. Dalam praktikum, kita menciptakan efek translasi dengan menggeser objek secara horizontal dan vertikal saat tombol mouse ditekan. Ini memberikan ilusi pergerakan objek tanpa perlu menciptakan objek yang sama berulang kali.

Transformasi eskalasi memungkinkan kita untuk mengubah ukuran objek dengan mengubah faktor skala pada setiap sumbu (X, Y, dan Z). Dalam percobaan ini, kita memperbesar kedua objek dengan faktor 2x setiap kali tombol mouse ditekan, menciptakan efek perbesaran objek yang terus meningkat.

Transformasi rotasi memungkinkan kita untuk memutar objek sekitar sumbu tertentu. Kita mengaplikasikan rotasi pada objek kedua sebesar 20 derajat pada sumbu Z setiap kali tombol mouse ditekan, menciptakan efek rotasi yang mengubah orientasi kedua objek.

Melalui praktikum ini, kami memahami bagaimana mengimplementasikan dan menerapkan transformasi 2D dan 3D dalam program grafik komputer. Transformasi ini adalah alat yang kuat untuk menciptakan efek visual dan animasi yang menarik dalam dunia grafik komputer.

Selain itu, kita juga memahami bahwa transformasi bersifat akumulatif, dan setiap kali suatu transformasi diterapkan, ia mempengaruhi posisi atau ukuran objek untuk transformasi berikutnya. Ini memberikan fleksibilitas dan kreativitas dalam menciptakan efek yang beragam dalam grafik komputer.