

TECHNICAL REPORT
GRAFIKA KOMPUTER
MODUL 7



Disusun Oleh :

TGL. PRAKTIKUM	: Jum'at, 4 Juni 2021
NAMA	: Achmad Farid Alfa Waid
NRP	: 190411100073
KELOMPOK	: 2
DOSEN	: Ach. Khozaimi, S.Kom., M.Kom

TELAH DISETUJUI TANGGAL :

.....
ASISTEN PRAKTIKUM

Rizky Alfriansyah
(180411100068)

JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

TUJUAN DAN DASAR TEORI

A. TUJUAN

Mahasiswa dapat mengetahui dan memahami cara membuat inputan keyboard dan animasi pada object.

B. DASAR TEORI

Callback merupakan istilah untuk function yang di passing ke dalam function lain sebagai argument, yang kemudian di eksekusi oleh function yang membungkus function callback tersebut. Dalam penggunaan glutKeyboardFunc dimungkinkan untuk mendeteksi input dari keyboard. Fungsi ini diletakkan pada fungsi main dari program, dan parameternya adalah callback function.

Dalam ilmu komputer, Komputasi Interaktif merupakan interaksi antara perangkat lunak yang menerima masukkan atau input dari pengguna(user) saat program dijalankan. Simulasi merupakan suatu proses peniruan dari yang nyata yang berada di sekelilingnya (state of affairs). simulasi ini secara umum menggambarkan sifat sifat karakteristik kunci dari kelakuan sistem fisik atau sistem

BAB II

PEMBAHASAN

A. SOAL

1. Buatlah program untuk menampilkan gambar bintang segi 4 dengan warna yang dapat diubah dengan menggunakan tombol panah ↑ dan ↓
2. Modifikasi program 7.4 supaya
 - a. ketika tombol mouse kiri ditekan, kotak berotasi terhadap sumbu z (rotasi yang sekarang)
 - b. ketika tombol mouse tengah ditekan, kotak berotasi terhadap sumbu y
 - c. ketika tombol mouse kanan ditekan, kotak berotasi terhadap sumbu x
 - d. start dan stop pergerakan dilakukan dengan menekan tombol 'p' atau 'P'
3. Modifikasi program 7.4 dengan mengganti kotak yang diputar dengan checkerboard 8 x 8
4. Modifikasi program 7.4 supaya
 - a. Ketika ditekan tombol 'u', ukuran kotak mengecil menjadi 0.75 ukuran semula
 - b. Ketika ditekan tombol 'U', ukuran kotak membesar menjadi 1.5 ukuran semula
 - c. Ketika ditekan tombol 'v', kecepatan putaran kotak jadi melambat setengah kali lipat dari kecepatan semula
 - d. Ketika ditekan tombol 'V', kecepatan putaran kotak jadi lebih cepat dua kali lipat dari kecepatan semula

B. JAWABAN

1. Code Program

1) Nomor 1

```
#include <windows.h>
```

```
#ifdef __APPLE__
```

```

#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>

void init(void) {
glClearColor(0.0,0.0, 0.0, 0.0);
glScaled(3.0,3.0,0.0);
}

void Bintang(){
    glBegin(GL_POLYGON);

    glVertex2f(10.0,60.0);
    glVertex2f(30.0,50.0);
    glVertex2f(10.0,40.0);
    glVertex2f(0.0,20.0);
    glVertex2f(-10.0,40.0);
    glVertex2f(-30.0,50.0);
    glVertex2f(-10.0,60.0);
    glVertex2f(0.0,80.0);
glEnd();

}

void Display(){
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    Bintang();
    glPopMatrix();
    glFlush();
}

void mySpecialKeyboard(int key, int x, int y) {
    switch (key)
    {
        case GLUT_KEY_UP:

```

```

        glColor3f(0.1,0.7,0.2);
        glutPostRedisplay();
        break;
    case GLUT_KEY_DOWN:
        glColor3f(0.2,0.1,0.7);
        glutPostRedisplay();
        break;
    }
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(100,100);
    glutInitWindowSize (400, 400);
    glutCreateWindow ("No 1");
    init();
    gluOrtho2D(-320,320,-320,320);
    glutDisplayFunc (Display);
    glutSpecialFunc(mySpecialKeyboard);
    glutMainLoop();
    return 0;
}

```

2) Nomor 2

```

#include<windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>

static GLfloat spin = 0.0;
float x;
float y;
float z;

void init(void) {

```

```

        glClearColor (0.0, 0.0, 0.0, 0.0);
        glShadeModel (GL_FLAT);
    }

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    glRotatef(spin, 0.0, 0.0, 1.0);
    glColor3f(1.0, 1.0, 1.0);
    glRectf(-25.0, -25.0, 25.0, 25.0);
    glPopMatrix();
    glutSwapBuffers();
}

void spinDisplay(void) {
    spin = spin + 0.01;
    if (spin > 360.0)
        spin = spin - 360.0;
    glutPostRedisplay();
}

void reshape(int w, int h) {
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void keyboard(unsigned char key, int x, int y) {
    switch (key) {
        case 'p':
            glutIdleFunc(spinDisplay);
            break;
        case 'P':
            glutIdleFunc(NULL);
            break;
        default:
            break;
    }
}

```

```
    }  
}
```

```
void mouse(int button, int state, int x, int y) {  
    switch (button)  
    {  
        case GLUT_LEFT_BUTTON:  
            if (state==GLUT_DOWN)  
                glutIdleFunc(spinDisplay);  
            break;  
        case GLUT_MIDDLE_BUTTON:  
            if (state==GLUT_DOWN)  
                glRotatef(spin, 0.0, 1.0, 0.0);  
                glutIdleFunc(spinDisplay);  
            break;  
        case GLUT_RIGHT_BUTTON:  
            if (state==GLUT_DOWN)  
                glRotatef(spin, 1.0, 0.0, 0.0);  
                glutIdleFunc(spinDisplay);  
            break;  
        default:  
            break;  
    }  
}
```

```
int main(int argc, char** argv) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);  
    glutInitWindowSize (400, 400);  
    glutInitWindowPosition (100, 100);  
    glutCreateWindow ("No 2");  
    init ();  
    glutDisplayFunc(display);  
    glutReshapeFunc(reshape);  
    glutMouseFunc(mouse);  
    glutKeyboardFunc(keyboard);  
    glutMainLoop();  
    return 0;  
}
```

3) Nomor 3

```
#include<windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>

static GLfloat spin = 0.0;

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    glRotatef(spin, 0.0, 0.0, 1.0);

    glColor3f(0.0, 0.0, 1.0);
    glTranslated(10.0, 0.0, 0);
    glRecti(0,0, 10, 10);
    glTranslated(20.0, 0.0, 0);
    glRecti(0,0, 10, 10);
    glTranslated(20.0, 0.0, 0);
    glRecti(0,0, 10, 10);
    glTranslated(20.0, 0.0, 0);
    glRecti(0,0, 10, 10);
    int a;
    for (a=1; a<=7; a++){
        if (a%2==1) {
            glTranslated(-70.0, 10.0, 0);
            glRecti(0,0, 10, 10);
            glTranslated(20.0, 0.0, 0);
            glRecti(0,0, 10, 10);
            glTranslated(20.0, 0.0, 0);
            glRecti(0,0, 10, 10);
            glTranslated(20.0, 0.0, 0);
            glRecti(0,0, 10, 10);
        }
        else {
```



```

        glTranslated(-50.0, 10.0, 0);
        glRecti(0,0, 10, 10);
        glTranslated(20.0, 0.0, 0);
        glRecti(0,0, 10, 10);
        glTranslated(20.0, 0.0, 0);
        glRecti(0,0, 10, 10);
        glTranslated(20.0, 0.0, 0);
        glRecti(0,0, 10, 10);
    }

}

glPopMatrix();
glutSwapBuffers();
}

void spinDisplay(void) {
    spin = spin + 0.01;
    if (spin > 360.0)
        spin = spin - 360.0;
    glutPostRedisplay();
}

void init(void) {
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_FLAT);
}

void reshape(int w, int h) {
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-100.0, 100.0, -100.0, 100.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void mouse(int button, int state, int x, int y) {
    switch (button) {
        case GLUT_LEFT_BUTTON:
            if(state==GLUT_DOWN)
                glutIdleFunc(spinDisplay);
            break;
    }
}

```

```

case GLUT_MIDDLE_BUTTON:
case GLUT_RIGHT_BUTTON:
if (state==GLUT_DOWN)
glutIdleFunc(NULL);
break;
default:
break;
}
}

int main(int argc, char** argv) {
glutInit(&argc, argv);
glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
glutInitWindowSize (400, 400);
glutInitWindowPosition (100, 100);
glutCreateWindow ("No 3");
init ();
glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutMouseFunc(mouse);
glutMainLoop();
return 0;
}

```

4) Nomor 4

```

#include<windows.h>
#ifdef __APPLE__
#include <GLUT/glut.h>
#else
#include <GL/glut.h>
#endif

#include <stdlib.h>

static GLfloat spin = 0.0;

void display(void) {
glClear(GL_COLOR_BUFFER_BIT);
glPushMatrix();
glRotatef(spin, 0.0, 0.0, 1.0);

```

```

glColor3f(1.0, 1.0, 1.0);
glRectf(-25.0, -25.0, 25.0, 25.0);
glPopMatrix();
glutSwapBuffers();
}

void spinDisplay(void) {
    spin = spin + 0.01;
    if (spin > 360.0)
        spin = spin - 360.0;
    glutPostRedisplay();
}

void init(void) {
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_FLAT);
}

void reshape(int w, int h) {
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-100.0, 100.0, -100.0, 100.0, -20.0, 20.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void mouse(int button, int state, int x, int y) {
    switch (button) {
        case GLUT_LEFT_BUTTON:
            if(state==GLUT_DOWN)
                glutIdleFunc(spinDisplay);
            break;
        case GLUT_MIDDLE_BUTTON:
        case GLUT_RIGHT_BUTTON:
            if(state==GLUT_DOWN)
                glutIdleFunc(NULL);
            break;
        default:
            break;
    }
}

```

```

}

void keyboard(unsigned char key, int x, int y){
    switch (key){
        case 'u':
            glScaled(0.75,0.75,0.75);
            glutIdleFunc(spinDisplay);
            break;
        case 'U':
            glScaled(1.5,1.5,1.5);
            glutIdleFunc(spinDisplay);
            break;
        case 'v':
            spin = spin + 0.005;
            glutIdleFunc(spinDisplay);
        case 'V':
            spin = spin + 0.02;
            glutIdleFunc(spinDisplay);
    }
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (400, 400);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("No 4");
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(mouse);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

2. Penjelasan Code Program

1) Nomor 1

```

void Bintang(){
    glBegin(GL_POLYGON);

```

```

    glVertex2f(10.0,60.0);
    glVertex2f(30.0,50.0);
    glVertex2f(10.0,40.0);
    glVertex2f(0.0,20.0);
    glVertex2f(-10.0,40.0);
    glVertex2f(-30.0,50.0);
    glVertex2f(-10.0,60.0);
    glVertex2f(0.0,80.0);
    glEnd();

}

```

Membuat sebuah fungsi bintang untuk menampilkan sebuah bintang.

```

void Display(){
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    Bintang();
    glPopMatrix();
    glFlush();
}

```

Pada fungsi display di dalam glPushMatrix dan glPopMatrix terdapat pemanggilan fungsi Bintang agar ditampilkan menjadi sebuah output.

```

void mySpecialKeyboard(int key, int x, int y) {
    switch (key)
    {
    case GLUT_KEY_UP:
        glColor3f(0.1,0.7,0.2);
        glutPostRedisplay();
        break;
    case GLUT_KEY_DOWN:

```

```

        glColor3f(0.2,0.1,0.7);
        glutPostRedisplay();
        break;
    }
}

```

Membuat sebuah fungsi mySpecialKeyboard untuk inputan keyboard, agar ketika terdapt inputan di user, program akan melakukan sesuatu. Pada kasus kali ini, ketika user menginputkan key up atau tanda panah atas, maka akan mengganti warna pada object menjadi hijau, sedangkang ketika menginputkan key down atau tanda panah ke bawah, maka akan menggani warna pad object menjadi warna biru.

```

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition(100,100);
    glutInitWindowSize (400, 400);
    glutCreateWindow ("No 1");
    init();
    gluOrtho2D(-320,320,-320,320);
    glutDisplayFunc (Display);
    glutSpecialFunc(mySpecialKeyboard);
    glutMainLoop();
    return 0;
}

```

Pada fungsi main, terdapat sebuah fungsi glutSpecialFunc agar fungsi inputan keyboard yang dibuat dapat dijalankan.

2) Nomor 2

```

static GLfloat spin = 0.0;
float x;

```

```
float y;
```

```
float z;
```

Mendeklarasikan nilai spin, x, y, dan z.

```
void display(void) {  
    glClear(GL_COLOR_BUFFER_BIT);  
    glPushMatrix();  
    glRotatef(spin, 0.0, 0.0, 1.0);  
    glColor3f(1.0, 1.0, 1.0);  
    glRectf(-25.0, -25.0, 25.0, 25.0);  
    glPopMatrix();  
    glutSwapBuffers();  
}
```

Di dalam fungsi display digunakan untuk menampilkan object, dimana diletakkan antara glPushMatrix dan glPopMatrix, kemudian dilakukan rotate dan pemberian warna.

```
void spinDisplay(void) {  
    spin = spin + 0.01;  
    if (spin > 360.0)  
        spin = spin - 360.0;  
    glutPostRedisplay();  
}
```

Membuat sebuah fungsi spinDisplay untuk mengubah nilai dari variable spin, agar animasi yang dijalankan dapat berjalan.

```
void reshape(int w, int h) {  
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);  
    glMatrixMode(GL_PROJECTION);  
    glLoadIdentity();  
    glOrtho(-50.0, 50.0, -50.0, 50.0, -1.0, 1.0);  
    glMatrixMode(GL_MODELVIEW);  
    glLoadIdentity();  
}
```

```
}
```

Membuat sebuah fungsi reshape ntuk memntukan titik object sebelah mana dan viewport yang akan ditampilkan.

```
void keyboard(unsigned char key, int x, int y) {  
    switch (key)    {  
        case 'p':  
            glutIdleFunc(spinDisplay);  
            break;  
        case 'P':  
            glutIdleFunc(NULL);  
            break;  
        default:  
            break;  
    }  
}
```

Membuat sebuah fungsi keyboard agar ketikan menerima inputan keyboard “p” amaka akan menjalankan animasi dengan fungsi spinDisplay, sedangkan ketika inputan keyboard “P” akan memberhentikan animasi.

```
void mouse(int button, int state, int x, int y) {  
    switch (button)  
    {  
        case GLUT_LEFT_BUTTON:  
            if(state==GLUT_DOWN)  
                glutIdleFunc(spinDisplay);  
            break;  
        case GLUT_MIDDLE_BUTTON:  
            if (state==GLUT_DOWN)  
                glRotatef(spin, 0.0, 1.0, 0.0);  
                glutIdleFunc(spinDisplay);  
    }
```



```

        break;
case GLUT_RIGHT_BUTTON:
    if (state==GLUT_DOWN)
        glRotatef(spin, 1.0, 0.0, 0.0);
        glutIdleFunc(spinDisplay);
        break;
default:
    break;
}
}

```

Membuat sebuah fungsi mouse, agar ketika menerima inputan dari mouse nilai rotasi dapat berubah, dan pergerakan animasi ikut berubah.

```

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (400, 400);
    glutInitWindowPosition (100, 100);
    glutCreateWindow ("No 2");
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(mouse);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}

```

Di dalam fungsi main terdapat syntax `glutReshapeFunc(reshape);` untuk menjalankan fungsi reshape, sedangkan syntax

glutMouseFunc(mouse); untuk menjalankan fungsi mouse dan syntax
glutKeyboardFunc(keyboard); untuk menjalankan fungsi keyboard.

3) Nomor 3

```
void display(void) {  
    glClear(GL_COLOR_BUFFER_BIT);  
    glPushMatrix();  
    glRotatef(spin, 0.0, 0.0, 1.0);
```

```
    glColor3f(0.0, 0.0, 1.0);  
    glTranslated(10.0, 0.0, 0);  
    glRecti(0,0, 10, 10);  
    glTranslated(20.0, 0.0, 0);  
    glRecti(0,0, 10, 10);  
    glTranslated(20.0, 0.0, 0);  
    glRecti(0,0, 10, 10);  
    glTranslated(20.0, 0.0, 0);  
    glRecti(0,0, 10, 10);  
    int a;  
    for (a=1; a<=7; a++){  
        if (a%2==1) {  
            glTranslated(-70.0, 10.0, 0);  
            glRecti(0,0, 10, 10);  
            glTranslated(20.0, 0.0, 0);  
            glRecti(0,0, 10, 10);  
            glTranslated(20.0, 0.0, 0);  
            glRecti(0,0, 10, 10);  
            glTranslated(20.0, 0.0, 0);  
            glRecti(0,0, 10, 10);  
        }  
        else {
```

```

        glTranslated(-50.0, 10.0, 0);
        glRecti(0,0, 10, 10);
        glTranslated(20.0, 0.0, 0);
        glRecti(0,0, 10, 10);
        glTranslated(20.0, 0.0, 0);
        glRecti(0,0, 10, 10);
        glTranslated(20.0, 0.0, 0);
        glRecti(0,0, 10, 10);
    }
}

glPopMatrix();
glutSwapBuffers();
}

```

Di dalam fungsi display diunakan untuk membuat chekcboard.\

```

void spinDisplay(void) {
    spin = spin + 0.01;
    if (spin > 360.0)
        spin = spin - 360.0;
    glutPostRedisplay();
}

```

Membuat sebuah fungsi spinDisplay untuk mengubah nilai dari variable spin, agar animasi yang dijalankan dapat berjalan.

```

void reshape(int w, int h) {
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-100.0, 100.0, -100.0, 100.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

```

Membuat sebuah fungsi reshape ntuk memntukan titik object sebelah mana dan viewport yang akan ditampilkan.

```
void mouse(int button, int state, int x, int y) {  
    switch (button)  
    {  
        case GLUT_LEFT_BUTTON:  
            if(state==GLUT_DOWN)  
                glutIdleFunc(spinDisplay);  
            break;  
        case GLUT_MIDDLE_BUTTON:  
            if (state==GLUT_DOWN)  
                glRotatef(spin, 0.0, 1.0, 0.0);  
                glutIdleFunc(spinDisplay);  
            break;  
        case GLUT_RIGHT_BUTTON:  
            if (state==GLUT_DOWN)  
                glRotatef(spin, 1.0, 0.0, 0.0);  
                glutIdleFunc(spinDisplay);  
            break;  
        default:  
            break;  
    }  
}
```

Membuat sebuah fungsi mouse, agar ketika menerima inputan dari mouse nilai rotasi dapat berubah, dan pergerakan animasi ikut berubah.

```
int main(int argc, char** argv) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);  
    glutInitWindowSize (400, 400);
```

```

    glutInitWindowPosition (100, 100);
    glutCreateWindow ("No 3");
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(mouse);
    glutMainLoop();
    return 0;
}

```

Di dalam fungsi main terdapat syntax `glutReshapeFunc(reshape);` untuk menjalankan fungsi reshape, sedangkan syntax `glutMouseFunc(mouse);` untuk menjalankan fungsi mouse dan syntax `glutKeyboardFunc(keyboard);` untuk menjalankan fungsi keyboard.

4) Nomor 4

```

void display(void) {
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    glRotatef(spin, 0.0, 0.0, 1.0);
    glColor3f(1.0, 1.0, 1.0);
    glRectf(-25.0, -25.0, 25.0, 25.0);
    glPopMatrix();
    glutSwapBuffers();
}

```

Di dalam fungsi display digunakan untuk menampilkan object, dimana diletakkan antara `glPushMatrix` dan `glPopMatrix`, kemudian dilakukan rotate dan pemberian warna.

```

void spinDisplay(void) {
    spin = spin + 0.01;
    if (spin > 360.0)
        spin = spin - 360.0;
}

```

```

    glutPostRedisplay();
}

```

Membuat sebuah fungsi spinDisplay untuk mengubah nilai dari variable spin, agar animasi yang dijalankan dapat berjalan.

```

void reshape(int w, int h) {
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(-100.0, 100.0, -100.0, 100.0, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

```

Membuat sebuah fungsi reshape ntuk memntukan titik object sebelah mana dan viewport yang akan ditampilkan.

```

void mouse(int button, int state, int x, int y) {
    switch (button)
    {
    case GLUT_LEFT_BUTTON:
        if(state==GLUT_DOWN)
            glutIdleFunc(spinDisplay);
        break;
    case GLUT_MIDDLE_BUTTON:
        if (state==GLUT_DOWN)
            glRotatef(spin, 0.0, 1.0, 0.0);
            glutIdleFunc(spinDisplay);
        break;
    case GLUT_RIGHT_BUTTON:
        if (state==GLUT_DOWN)
            glRotatef(spin, 1.0, 0.0, 0.0);
            glutIdleFunc(spinDisplay);
    }
}

```

```

        break;
    default:
        break;
    }
}

```

Membuat sebuah fungsi mouse, agar ketika menerima inputan dari mouse nilai rotasi dapat berubah, dan pergerakan animasi ikut berubah.

```

void keyboard(unsigned char key, int x, int y){
    switch (key){
        case 'u':
            glScaled(0.75,0.75,0.75);
            glutIdleFunc(spinDisplay);
            break;
        case 'U':
            glScaled(1.5,1.5,1.5);
            glutIdleFunc(spinDisplay);
            break;
        case 'v':
            spin = spin + 0.005;
            glutIdleFunc(spinDisplay);
        case 'V':
            spin = spin + 0.02;
            glutIdleFunc(spinDisplay);
    }
}

```

Membuat fungsi keyboar agar ketika programn menerima inputan “u” maka akan discale sebanyak 0.75 tiap koordinat, dan ketikan menerima inputan “U” maka akan disclae sebanyak 1.5 pada setiap koordinat. Dan ketika menerima inputan “v” maka kecepatan

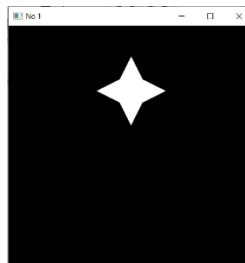
perputaran animasi ditambah sebanyak 0.005 dan ketika menerima inputan “v” kecepatan animasi akan ditambah sebanyak 0.02.

```
int main(int argc, char** argv) {  
    glutInit(&argc, argv);  
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);  
    glutInitWindowSize (400, 400);  
    glutInitWindowPosition (100, 100);  
    glutCreateWindow ("No 4");  
    init ();  
    glutDisplayFunc(display);  
    glutReshapeFunc(reshape);  
    glutMouseFunc(mouse);  
    glutKeyboardFunc(keyboard);  
    glutMainLoop();  
    return 0;  
}
```

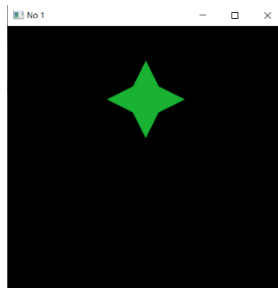
Di dalam fungsi main terdapat syntax `glutReshapeFunc(reshape);` untuk menjalankan fungsi reshape, sedangkan syntax `glutMouseFunc(mouse);` untuk menjalankan fungsi mouse dan syntax `glutKeyboardFunc(keyboard);` untuk menjalankan fungsi keyboard.

3. Hasil Running Program

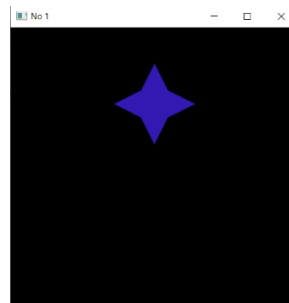
1) Nomor 1



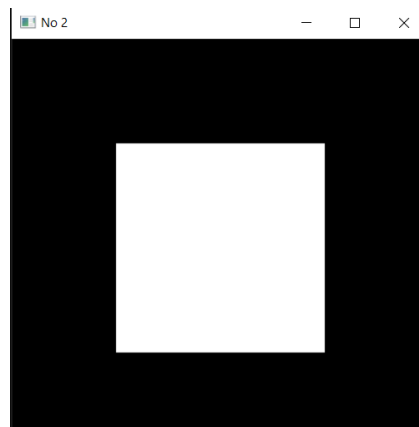
Ketika menerima inputan arah panah atas.



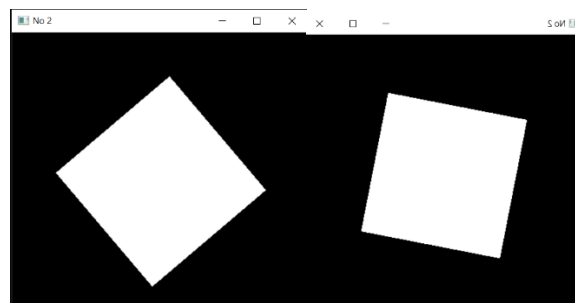
Ketika menerima inputan arah panah atas.



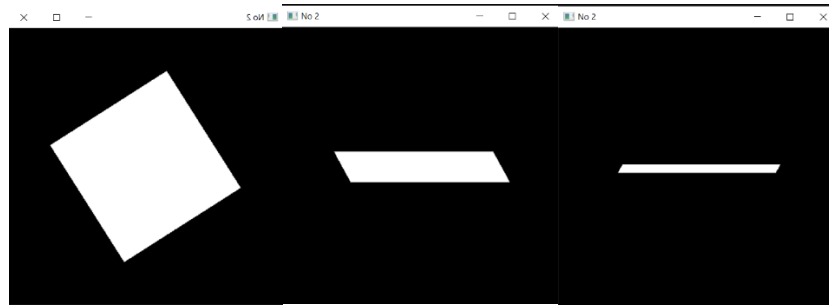
2) Nomor 2



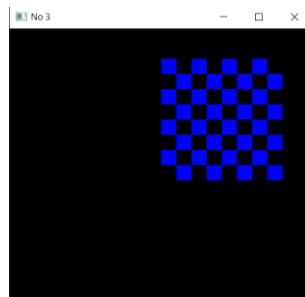
Ketika animasi dijalankan menggunakan inputan "p" dan berhenti menggunakan inputan "P"



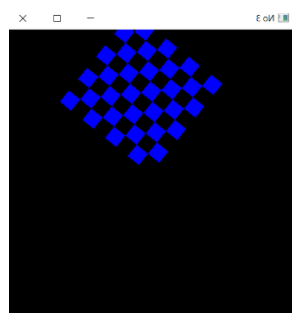
Ketika menerima inputan mouse maka akan berotasi



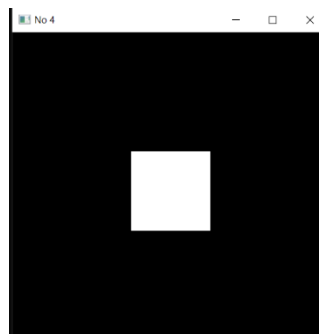
3) Nomor 3



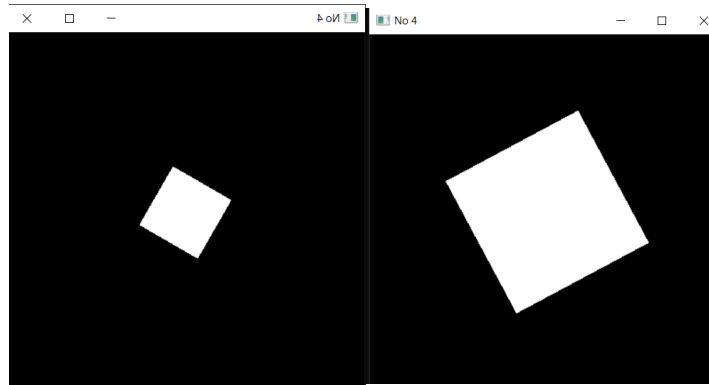
Ketika animasi dijalankan



4) Nomor 4



Ketika menerima inputan “u” maka akan diperkecil dan menerima inputan “U” akan diperbesar. Dan ketika menerima inputan “v” dan “V” maka kecepatan animasi akan dirubah sesuai dengan intruksi pada modul.



BAB III

PENUTUP

A. Kesimpulan

1. Mahasiswa mampu menerapkan inputan keyboard dan mouse pada object yang telah dibuat, dan mahasiswa juga dapat menganimasikan object.
2. Inputan object sangat berguna, sehingga membuat sebuah program dan hasil menjadi interaktif oleh user.
3. Pembuatan objek 3 dimensi secara manual harus dibuat satu persatu lalu digabung dengan objek.

B. Saran

Diperlukan banyak latihan dan mencoba megotak-atik sendiri agar mahasiswa lebih memahami dan mendalami fungsi – fungsi yang ada pada openGL.