

# Modul 4

## Interaksi dengan Keyboard dan Mouse

### A. KOMPETENSI DASAR

- Memahami prinsip-prinsip deteksi input berupa interaksi dari keyboard.
- Membuat objek 2D yang dikendalikan dengan keyboard.
- Memahami prinsip-prinsip pembuatan interaksi objek 2D menggunakan *mouse*.
- Membuat interaksi objek 2D menggunakan *mouse*.

### B. ALOKASI WAKTU

2 JS (2x50 menit)

### C. PETUNJUK

- Awali setiap aktivitas dengan do'a, semoga berkah dan mendapat kemudahan.
- Pahami Tujuan, dasar teori, dan latihan-latihan praktikum dengan baik dan benar.
- Kerjakan tugas-tugas dengan baik, sabar, dan jujur.
- Tanyakan kepada asisten/dosen apabila ada hal-hal yang kurang jelas.

### D. DASAR TEORI

#### INTERAKSI KEYBOARD

##### 1. Fungsi dasar pembuatan animasi dengan menggunakan *KeyboardFunction*

```
GLUTAPI void APIENTRY glutKeyboardFunc(void (GLUTCALLBACK  
*func)(unsigned char key, int x, int y));
```

Dalam penggunaan glutKeyboardFunc dimungkinkan untuk mendeteksi input dari keyboard. Fungsi ini diletakkan pada fungsi main dari program, dan parameternya adalah callback function yang telah didefinisikan berupa fungsi dengan 3 parameter, seperti contoh di bawah ini.

```
void myKeyboard(unsigned char key, int x, int y){  
    if(key == 'a') glTranslatef(4,0,0); //seleksi tombol yang ditekan  
}  
  
void mySpecialKeyboard(int key, int x, int y){  
    switch(key){  
        case GLUT_KEY_??? : ...; break;  
    }  
}
```

```
}
```

Agar fungsi keyboard ini dapat dideteksi terus maka fungsi untuk **animasi (update) harus telah disertakan.**

Untuk fungsi callback yang memanggil tombol keyboard normal/biasa adalah

```
glutKeyboardFunc(myKeyboard); //hanya memanggil fungsi myKeyboard
```

sedangkan untuk mendeteksi tombol-tombol keyboard yang bersifat spesial seperti tombol F1, arah panah, Home, Enter, dsb dapat menggunakan callback function

```
glutSpecialFunc(mySpecialKeyboard); //hanya memanggil fungsi mySpecialKeyboard
```

untuk tombol-tombol spesialnya adalah sebagai berikut

GLUT_KEY_F1	F1 function key	GLUT_KEY_LEFT	Left function key
GLUT_KEY_F2	F2 function key	GLUT_KEY_RIGHT	Up function key
GLUT_KEY_F3	F3 function key	GLUT_KEY_UP	Right function
GLUT_KEY_F4	F4 function key	key	
GLUT_KEY_F5	F5 function key	GLUT_KEY_DOWN	Down function key
GLUT_KEY_F6	F6 function key	GLUT_KEY_PAGE_UP	Page Up function
GLUT_KEY_F7	F7 function key	key	
GLUT_KEY_F8	F8 function key	GLUT_KEY_PAGE_DOWN	Page Down
GLUT_KEY_F9	F9 function key	function key	
GLUT_KEY_F10	F10 function key	GLUT_KEY_HOME	Home function key
GLUT_KEY_F11	F11 function key	GLUT_KEY_END	End function key
GLUT_KEY_F12	F12 function key	GLUT_KEY_INSERT	Insert

## INTERAKSI MOUSE

### 1. Fungsi dasar pembuatan interaksi dengan menggunakan *MouseFunction*

```
GLUTAPI void APIENTRY glutMouseFunc(void (GLUTCALLBACK *func)(int
button, int state, int x, int y));
```

Paramater *func* adalah fungsi yang akan ditangani dengan event klik mouse.

```
GLUTAPI void APIENTRY glutMotionFunc(void *func(int x, int y));
```

Fungsi di atas adalah fungsi pelengkap dari fungsi interaksi mouse untuk mendeteksi gerakan mouse.

### 2. Inisialisasi dalam penggunaan *MouseFunction*

```
void mouse(int button, int state, int x,int y){
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
        drawDot(x,480-y);
    if(button==GLUT_RIGHT_BUTTON && state==GLUT_DOWN)
        drawDot2(x,480-y);
    if(button==GLUT_MIDDLE_BUTTON && state==GLUT_DOWN)
        drawDot3(x,480-y);
}
```

```
void motion(int x,int y){
}
```

GLUT\_LEFT\_BUTTON untuk inialisasi button mouse kiri.

GLUT\_RIGHT\_BUTTON untuk inialisasi button mouse kanan.

GLUT\_MIDDLE\_BUTTON untuk inialisasi button mouse tengah.

Fungsi dari GLUT\_DOWN adalah untuk inialisasi ketika tombol mouse ditekan.

Fungsi dari GLUT\_UP adalah untuk inialisasi ketika tombol mouse dilepaskan.

Kemudian pada *main* program perlu menambahkan fungsi untuk *callback* fungsi *MouseFunction*.

```
glutMouseFunc(mouse);
glutMotionFunc(motion);
```

## E. AKTIFITAS PRAKTIKUM

### PRAKTIKUM INTERAKSI KEYBOARD

Berikut adalah script dasar untuk kegiatan praktikum interaksi keyboard

```
glutInitWindowSize(640, 480);
glutInitWindowPosition(100, 100);
```

dan

```
glutTimerFunc(50,timer,0);
glutDisplayFunc(display);
glutKeyboardFunc(myKeyboard);
glutSpecialFunc(mySpecialKeyboard);
```

1. Buatlah project baru pada Visual Studio dengan nama prak4-keyboard1. Berikut adalah fungsi-fungsi yang harus ditambahkan. Sediakan fungsi `drawQuad()`; untuk menggambar sebuah kotak.

```
void renderScene() {
    glClear(GL_COLOR_BUFFER_BIT);
    drawQuad(30,10);
    glFlush();
}

void myKeyboard(unsigned char key, int x, int y){
    if(key == 'a') glTranslatef(-4,0,0);
    else if(key == 'd') glTranslatef(4,0,0);
}

void update(int value){
    glutPostRedisplay();
    glutTimerFunc(50,update,0);
}
```

Fungsi `myKeyboard` adalah callback function yang akan dipanggil oleh `glutKeyboardFunc(myKeyboard)`; dan fungsi tersebut berada di dalam fungsi `main`.

**Berikan kesimpulan dari hasil kegiatan (mengacu pada fungsi `myKeyboard`)!**

2. Modifikasi program pada latihan 1 dan tambahkan program pada fungsi **myKeyboard** sehingga dapat menggerakkan objek ke atas dan ke bawah. Tampilkan source codenya. **Berikan kesimpulan!**

3. Buatlah project baru pada Visual Studio dengan nama prak4-keyboard2. Berikut adalah cara untuk mendeteksi tombol-tombol keyboard yang memiliki fungsi spesial. **Berikan kesimpulan!**

```
void mySpecialKeyboard(int key, int x, int y){
    switch(key){
        case GLUT_KEY_LEFT : //deteksi tombol tanda panah kiri
            glRotatef(4,0,0,1);
            break;
    }
}
```

Dan pada fungsi main tambahkan

`glutSpecialFunc(mySpecialKeyboard);` **Beri Kesimpulan!**

4. Modifikasi kode program pada latihan 3 dan tambahkan program pada fungsi **mySpecialKeyboard** sehingga dapat memutar objek kebalikan putaran dengan menekan tombol panah kanan. Tampilkan source codenya. **Berikan kesimpulan!**
5. Tambahkan pada fungsi **myKeyboard**, untuk merubah warna obyek sebanyak 4 pilihan merah, hijau, biru, kuning (deteksi 4 tombol keyboard (normal) lain)!

## PRAKTIKUM INTERAKSI MOUSE

Berikut adalah script dasar untuk kegiatan praktikum interaksi mouse.

```
int w = 480, h = 480; //variabel global
```

```
glutInitWindowSize(w,h);
gluOrtho2D(-w/2,w/2,-h/2,h/2);
```

1. Buatlah project baru pada Visual Studio dengan nama prak4-mouseMotion. Fungsi **myDisplay** *callback function* yang di dalamnya menggunakan fungsi **Points** untuk memindah posisi objek sesuai *pointer mouse*.

```
float x=0,y=0,z=0;
void myDisplay(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glPushMatrix();
    glTranslatef(x,y,z);
    drawQuad(5,5);
    glPopMatrix();
    glFlush();
}
```

Fungsi mouse untuk inisialisasi *mouse event*.

```
void mouse(int button, int state, int xmouse, int ymouse)
{
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN) {
        x=xmouse-(w/2);
        y=(h/2)-ymouse;
    }
}
```

2. Sempurnakan program di atas sehingga tombol **tengah** ditekan akan menyebabkan objek memiliki skala yang membesar (1.1) dan tombol **kanan** ditekan akan menyebabkan skala mengecil (0.9). Berikan **source code** program.
3. Ubahlah program sehingga tombol kanan ditekan skala membesar (2.0) dan ketika tombol kanan dilepaskan (UP) skala mengecil (0.5). Berikan **source code** program.
4. Buatlah nilai parameter yang ada pada fungsi  
`gluOrtho2D(-100,100,-100,100);`  
Jelaskan apa yang terjadi dan beri kesimpulan (**gambar sistem koordinat**) hubungan antara sistem koordinat openGL dan sistem koordinat yang digunakan oleh *mouse pointer*.
5. Buatlah project baru pada Visual Studio dengan nama prak4-mouseMotion1. Gunakan fungsi yang sama dengan no 1 dan lengkapi dengan fungsi **Motion**. Buatlah fungsi untuk deteksi motion. **Berikan kesimpulan!**

```
void motion(int xmouse,int ymouse)
{
    x = xmouse-(w/2);
    y = (h/2)-ymouse;
}
```

### **CONTOH-CONTOH PROGRAM**

Grafika komputer telah berhasil mengembangkan interaksi dengan sarana keyboard dan light-pen yang ditemukan oleh Ivan Sutherland dengan sistem penggambaran interaktif.

#### **Input primitif grafika dan pirantinya**

Sebagaimana banyak piranti dan cara untuk pemaparan output grafika komputer, demikian pula untuk piranti input yaitu

- Keyboard (untuk memasukkan karakter atau string)
- Grafik Tablets
- Tombol
- Joystick dan trackball
- Mouse
- Knobs
- Space balls
- Data gloves

#### **Fungsi dasar pembuatan animasi dengan menggunakan *keyboardFunction***

Dalam penggunaan `glutKeyboardFunc` dimungkinkan untuk mendeteksi input dari keyboard. Fungsi ini diletakkan pada fungsi main dari program, dan parameternya adalah callback function yang telah didefinisikan berupa fungsi dengan 3 parameter, seperti contoh di bawah ini.

##### **1. Fungsi untuk karakter dan angka pada keyboard**

```
void myKeyboard(unsigned char key, int x, int y){  
    if(key == 'char ataupun angka')  
        //kode-kode y=untuk event yang akan di lakukan  
}
```

Misal :

```
void myKeyboard(unsigned char key, int x, int y){  
    if(key == 'a')  
        glRotatef ( 90,0,1,0 );
```

```
}
```

Agar fungsi tersebut dapat dideteksi secara berkala, maka pada fungsi main perlu ditambahkan fungsi : `glutKeyboardFunc (myKeyboard);`

## 2. Fungsi untuk tombol special

Tombol spesial meliputi :

- F1, F2, F3, F4,
- tombol navigasi panah keatas, panah kebawah, panah kekiri, dan panah kekanan,
- tombol page Down, page Up, Home, End, Insert

```
void mySpecialKeyboard(int key, int x, int y){
    switch(key){
        case GLUT_KEY_<opsional> :
            ...; // kode event yang akan dilakukan
            break;
    }
}
```

Misal :

```
void mySpecialKeyboard(int key, int x, int y){
    switch(key){
        case GLUT_KEY_LEFT> :
            glRotatef (90, 1, 0, 0);
            break;
    }
}
```

Agar fungsi tersebut dapat dideteksi secara berkala, maka pada fungsi main perlu ditambahkan fungsi : `glutSpecialFunc (mySpecialKeyboard);`

Contoh program :

```
#include <stdlib.h>
#include <glut.h>

void segitiga (){
    glBegin (GL_TRIANGLES);
    glColor3f (1,0,0);
    glVertex2f (0.,0.);
    glVertex2f (150,150);
```

```
    glVertex2f (-150,150);
    glColor3f (1,0,0);
    glVertex2f (0.,0.);
    glVertex2f (-150,-150);
    glVertex2f (150,-150);
    glEnd();
}

void myKeyboard(unsigned char key, int x, int y){
    if(key=='c') glRotatef (10,0,0,-4);
    else if(key=='u') glRotatef (10,0,0,4);
}

void renderScene(){
    glClear(GL_COLOR_BUFFER_BIT);
    segitiga();
    glFlush();
}

void timer(int value){
    glutPostRedisplay();
    glutTimerFunc(50,timer,0);
}

void main(int argc, char **argv){
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DEPTH | GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(640,480);
    glutCreateWindow("latihan 6");
    gluOrtho2D(-320.,320.,-340.,340.);
    glutDisplayFunc(renderScene);
    glutKeyboardFunc(myKeyboard);
    glutTimerFunc(1,timer,0);
    glClearColor(1.,1.,1.,0); // untuk memberikan background putih
    glutMainLoop();
}
```

Hasil :





### Triangle Keyboard

```
#include "stdlib.h"
#include "gl/glut.h"

void Triangles(){
    glBegin(GL_TRIANGLES);
    //glColor3f(0.5,0.5,0.5);
    glVertex2f (0.,0.);
    glVertex2f(10.,30.);
    glVertex2f(-10.,30.);
    glEnd();
}

void renderScene(void){
    glClear(GL_COLOR_BUFFER_BIT);
    Triangles();
    glFlush();
}

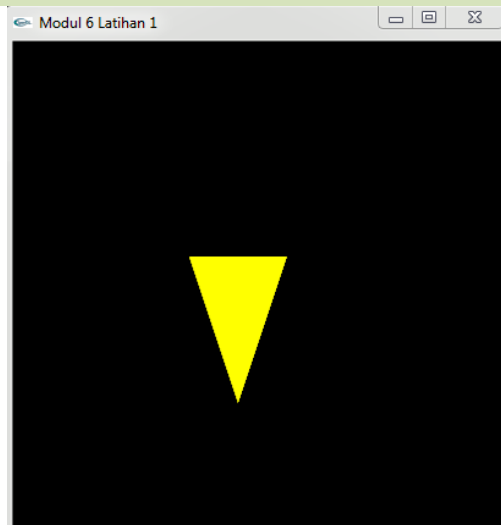
void myKeyboard(unsigned char key,int x,int y){
    if(key == 'a')glTranslatef(-4,0,0);
    else if (key == 'd')glTranslatef(4,0,0);
    else if (key == 'w')glTranslatef(0,4,0);
    else if (key == 's')glTranslatef(0,-4,0);
    else if (key == 'm')glColor3f(1.,0.,0.);
    else if (key == 'h')glColor3f(0.,1.,0.);
    else if (key == 'b')glColor3f(0.,0.,1.);
    else if (key == 'k')glColor3f(1.,1.,0.);
    else if (key == 'u')glColor3f(1.,0.,1.);
    else if (key == 'p')glColor3f(1.,1.,1.);
}
```

```
}

void mySpecialKeyboard(int key, int x, int y){
    switch(key){
        case GLUT_KEY_LEFT:
            glRotatef(4,0,0,1);
            break;
        case GLUT_KEY_RIGHT:
            glRotatef(-4,0,0,1);
            break;
    }
}

void update(int value){
    glutPostRedisplay();
    glutTimerFunc(50,update,0);
}

void main(int argc, char **argv){
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(400,400);
    glutCreateWindow("Modul 6 Latihan 1");
    gluOrtho2D(-50.,50.,-50.,50.);
    glutDisplayFunc(renderScene);
    glutTimerFunc(1,update,0);
    glutKeyboardFunc(myKeyboard);
    glutSpecialFunc(mySpecialKeyboard);
    glutMainLoop();
}
```



### Zooming Keyboard 2 Dimensi

```
//Memanggil Library
```

```
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <GL/glut.h>

//Deklarasi Variable
float z_pos=0.0f;
float rot=0.0f;

//Membuat Method
void mydisplay(){
    glClear(GL_COLOR_BUFFER_BIT);
    glLoadIdentity();
    glTranslatef(0.0,0.0f,z_pos);
    glRotatef(rot, 0, 0, 1);
    glBegin(GL_POLYGON);
    glColor3f(0, 1, 0);
    glVertex3f(-0.5, -0.5, -5);
    glColor3f(0, 0, 1);
    glVertex3f(-0.75, 0, -5);
    glColor3f(1, 0, 0);
    glVertex3f(-0.5, 0.5, -5);
    glColor3f(0, 1, 0);
    glVertex3f(0, 0.75, -5);
    glColor3f(0, 0, 1);
    glVertex3f(0.5, 0.5, -5);
    glColor3f(1, 0, 0);
    glVertex3f(0.75, 0, -5);
    glColor3f(0, 1, 0);
    glVertex3f(0.5, -0.5, -5);
    glColor3f(0, 0, 1);
    glVertex3f(0,-0.75, -5);
    glEnd();
    glFlush();
    glutSwapBuffers();
}

//Membuat Method init
void init( void )
```

```
{
    glClearColor( 1.0, 0.0, 0.0, 1.0 ); // A Background Clear Color
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(45, (GLdouble)500.0/(GLdouble)500.0, 0, 100);
    glMatrixMode(GL_MODELVIEW);
}

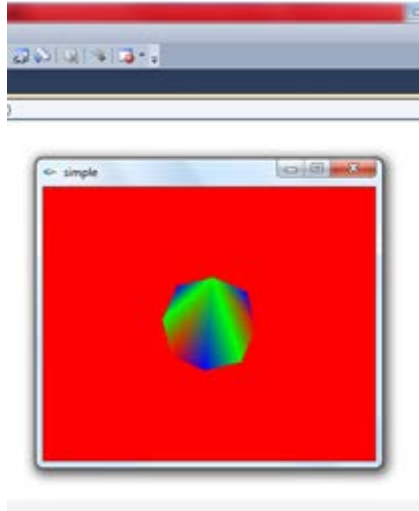
//Membuat Method resize
void resize( int w, int h )
{
    glViewport( 0, 0, (GLsizei) w, (GLsizei) h );
    glMatrixMode( GL_PROJECTION );
    glLoadIdentity();
    gluPerspective(45, (GLdouble)w/(GLdouble)h, 0, 100);
    glMatrixMode( GL_MODELVIEW );
}

//Membuat Method timeout
void myTimeOut(int id)
{
    // called if timer event
    // ...advance the state of animation incrementally...
    rot+=10;
    glutPostRedisplay(); // request redisplay
    glutTimerFunc(100, myTimeOut, 0); // request next timer event
}

//Membuat Method keyboard
void myKeyboard(unsigned char key,int x, int y)
{
    if((key=='<')||(key==',')) z_pos-=0.1f;
    if((key=='>')||(key=='.')) z_pos+=0.1f;
}

//Program Utama
int main(int argc, char** argv)
{
    glutInit(&argc,argv);
    //glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
```

```
glutCreateWindow("simple");  
// callbacks  
glutDisplayFunc(mydisplay);  
glutKeyboardFunc(myKeyboard);  
glutTimerFunc(100, myTimeOut, 0);  
glutReshapeFunc(resize);  
init();  
glutMainLoop();  
}
```

**Penjelasan :**

- Function `glMatrixMode()` digunakan untuk memberikan inisialisasi matriks.
- Function `glLoadIdentity()` digunakan untuk memanggil matriks identitas dan dijadikan status matriks (proyeksi) saat ini.
- Function `gluPerspective()` digunakan untuk mengatur perspektif yaitu perubahan atau pergeseran dari proyeksi.
- Function `glutPostRedisplay()` digunakan untuk mengirimkan perintah saat mengaktifkan display secara berkala (looping).
- Function `glutTimerFunc()` digunakan untuk membuat animasi yang dapat dikontrol oleh waktu.
- Function `glTranslatef()` digunakan untuk memindahkah objek.
- Function `glRotatef()` digunakan untuk memutar objek.
- Function `glutInitDisplayMode()` digunakan untuk untuk menentukan apakah object akan menggunakan model pewarnaan RGB atau indeks warna.
- Function `glutInitWindowSize()` digunakan untuk menentukan ukuran window.
- Function `glutInitWindowPosition()` digunakan untuk menentukan letak posisi sebuah window.
- Function `glutCreateWindow()` digunakan untuk membuat window.
- Function `glutDisplayFunc()` digunakan untuk memanggil fungsi display atau method untuk membuat objek primitive.

- Function `glutKeyboardFunc()` digunakan untuk memanggil fungsi inputan dari keyboard.
- Function `glutTimerFunc()` digunakan untuk memanggil fungsi animate.
- Function `glutMainLoop()` adalah event yang menandakan pemrosesan glut terjadi.

Jika Source Code tersebut dijalankan maka akan menghasilkan objek primitif segidelapan yang dapat berputar dan dapat berinteraksi dengan inputan melalui sebuah keyboard yang sudah di setting pada kode program seperti contoh 2 buah gambar dibawah ini. Output objek primitif segidelapan dihasilkan melalui method `mydisplay()`. Dalam method tersebut terdapat fungsi untuk memindahkan objek primitif sesuai dengan sumbu z (`glTranslatef(0.0,0.0f,z_pos)`) dan fungsi untuk memutar objek primitif yang berputar searah dengan sumbu z. Pada objek tersebut tiap-tiap titik diberikan warna yang berbeda.

Method `init()` digunakan untuk menginisialisasi seperti warna background

Method `myTimeOut()` digunakan untuk mengatur waktu berputarnya object

Method `myKeyboard` dengan menggunakan simbol `>` dan `<` untuk memperbesar atau memperkecil, dan simbol koma dan titik.

## Interaksi Mouse

```
#include "stdlib.h"
#include "stdio.h"
#include "gl/glut.h"

int x=0,y=0,z=0;
int w=480, h=480;

void drawQuad() {
    glBegin(GL_POLYGON);
    glColor3f(0,0,1);
    glVertex2f(-20.,-20.);
    glVertex2f(20.,-20.);
    glVertex2f(20.,20.);
    glVertex2f(-20.,20.);
    glEnd();
}

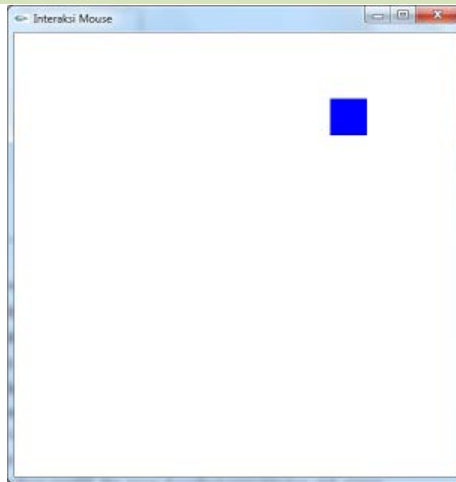
void mouse(int button, int state, int xmouse, int ymouse){
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN){

        x = xmouse-(w/2);
        y = (h/2)-ymouse;
        printf("    x = %d    y = %d",x,y);

    }
}

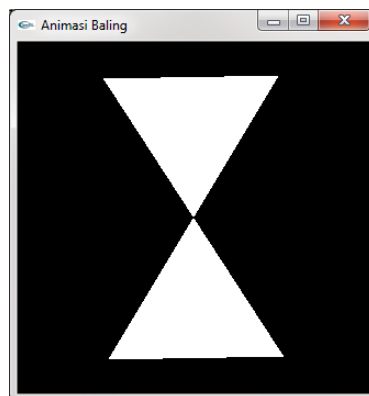
void renderScene(void){
    glClear(GL_COLOR_BUFFER_BIT);
    glClearColor(1,1,1,1);
    glPushMatrix();
    glTranslatef(x,y,z);
    drawQuad();
}
```

```
    glPopMatrix();  
    glFlush();  
}  
  
void timer(int value){  
    glutPostRedisplay();  
    glutTimerFunc(50,timer,0);  
}  
  
void main (int argc, char **argv){  
    glutInit(&argc, argv);  
    glutInitWindowPosition(100,100);  
    glutInitWindowSize(w,h);  
    glutCreateWindow("Interaksi Mouse");  
    gluOrtho2D(-w/2,w/2,-h/2,h/2);  
    glutDisplayFunc(renderScene);  
    glutMouseFunc(mouse);  
    glutTimerFunc(1,timer,0);  
    glutMainLoop();  
}
```

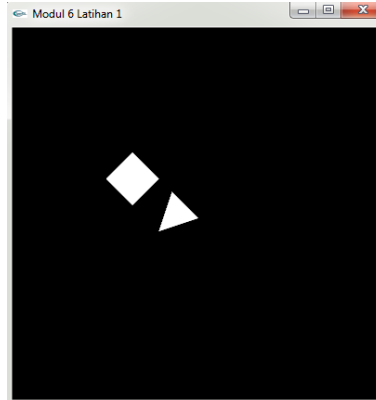


## F. TUGAS

1. Buatlah sebuah baling-baling yang bisa berputar secara **clockwise** dan **unclockwise** yang pusatnya berada pada pusat koordinat. Dengan kontrol dari tombol 'k' dan 'l'.



2. Buatlah 2 benda (kotak dan segitiga) yang dapat dikendalikan secara individual, dengan memanfaatkan tombol untuk tangan kanan dan tombol untuk tangan kiri. ('a', 's', 'd', 'w' untuk kotak dan 'tombol panah atas, bawah, kiri, kanan' untuk segitiga) atau dengan metode yang lain



3. Tambahkan intraksi keyboard menggunakan baling-baling berubah menjadi warna hijau bila ditekan huruf "p" dan menjadi kuning bila ditekan huruf "o"! ..

