# BAB 3
# EVENT HANDLING

## 3.1 Pendahuluan

Sebuah game sangat membutuhkan penanganan aksi yang terjadi pada mouse dan keyboard yang akan memberikan dampak atau perubahan kepada suatu objek. Hampir keseluruhan game komputer menggunakan dua item ini, dimana didalam java dikenal dengan istilah “event handling”. Sebenarnya event handling di java tidak hanya sebatas pada mouse dan keyboard saja tetapi juga objek-objek lain seperti item listener, componen listener dst, tetapi disini hanya membahas tentang masalah penanganan mouse dan keyboard. Sebagai contoh, apabila ingin mengetahui tombol apa yang ditekan maka disinilah fungsi event handling, tidak hanya itu tetapi masih banyak lagi seperti jika tombol space ditekan maka bom akan meledak, apabila tombol enter ditekan maka permainan akan dimulai dan masih banyak lagi.

## 3.2 Tujuan

Setelah mempelajari modul ini peserta diharapkan dapat:

- Mendengarkan action dari keyboard
- Menerima input dari keyboard
- Mengetahui kode tombol-tombol keyboard pada KeyEvent.
- Membaca pergerakan mouse
- Mendeteksi tombol mouse
- Menerima input dari mouse.

### 3.3 Input dari Keyboard

Java menyediakan Interface KeyListener untuk mendengarkan aksi dari keyboard dan mengirimkan action tersebut melalui method yang diikutsertakan dalam program. Adapun method-methodnya adalah *keyPressed,keyReleased, dan keyTyped* . Ketiga parameter dari method ini hanya mempunya satu parameter yaitu *KeyEvent.*

Ketika menulis program yang menggunakan KeyListener, maka harus menambahkan definisi dari class yang telah dibuat dengan menggunakan kata kunci *implements*, sebagai contoh:

```
Public class TestKeyboard extends JFrame
implements Keylistener{
      ......................
      ......................
}
```

Fitur Yang paling menarik dari *implements* pada java adalah dapat mengimplementasikan lebih dari satu interface didalam program dengan menggunakan tanda koma(,) sebagai pemisah dari masing-masing interface.

### 3.4 Mendengarkan Action dari Keyboard

Sebuah program membutuhkan method addKeyListener untuk mendengarkan aksi dari *keyboard,* agar key event akan di kirim oleh Java Runtime Environment(JRE) kepada program yang dibuat. Parameter dari method ini akan di turunkan pada class yang akan direpresentasikan dengan kata kunci (*this*). Bisanya untuk pemanggilan addKeyListener(this) dilakukan di dalam program dimana konstruktor akan di panggil pertama kali ketika sebuah program dijalankan.

Selanjutnya, mengimplementasikan 3(tiga) aksi dari keyboard di dalam program agar interface dari keyListener dapat bekerja.

```
public void keyPressed(KeyEvent  e)
public void keyReleased(KeyEvent  e)
public void keyTyped(KeyEvent  e)
```

Ada tiga jalan untuk menentukan key yang telah ditekan atau dilepaskan dengan menggunakan parameter KeyEvent. Jika ingin mengetahui karakter tombol yang ditekan, dapat menggunakan method getKeyChar yang mana akan mengirimkan nilai bertype char, tetapi jika ingin mengetahui kode apapun ditekan berdasarkan kode tombol maka dapat menggunakan method getKeyCode. Sebagai contoh ketika program mendengarkan aksi dari keyboard dan kemudian tombol A ditekan, maka method getKeychar akan mengembalikan hasil berupa “a”(atau “A” jika menggunakan huruf besar), tetapi method getKeyCode akan mengembalikan nilai berupa kode tombol virtual yaitu VK_A. Semua dari kode tombol virtual terdapat di dalam class KeyEvent. Tabel dibawah ini menampilkan sebagian daftar dari kode tombol virtual yang biasanya umum digunakan didalam pembuatan game.

| Kode Tombol | Penjelasan |
|---|---|
| VK_LEFT | Panah kiri |
| VK_RIGHT | Panah kanan |
| VK_UP | Panah atas |
| VK_DOWN | Panah bawah |
| VK_0 ..... VK_9 | Tombol Numerik |
| VK_A .... VK_Z | Tombol Alphabet |
| VK_F1 .. VK_F12 | Tombol Fungsi |
| VK_KP_LEFT | Numerik left |
| VK_KP_RIGHT | Numerik right |
| VK_KP_UP | Numerik atas |
| VK_KP_DOWN | Numerik bawah |
| VK_ENTER | Tombol Enter |
| VK_BACK_SPACE | Tombol Backspace |
| VK_TAB | Tombol Tab |

Sekarang membuat class TestKeyboard yang mengimplementasi interface KeyListener seperti pada kode program berikut.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.JFrame;

public class TestKeyboard extends JFrame implements KeyListener{

	int KodeTombol;
	char KarakterTombol;

	public static void main(String[] args) {
		TestKeyboard tk= new TestKeyboard();
	}
	TestKeyboard(){
		setTitle("Test Keyboard");
		addKeyListener(this);
		setSize(300, 200);
		setVisible(true);
		setDefaultCloseOperation(this.EXIT_ON_CLOSE);
	}
	public void paint(Graphics g){
		g.setColor(Color.WHITE);
		g.clearRect(0, 0, 300, 200);
		g.setColor(Color.BLACK);
		g.drawString("Tekan sebuah tombol : ", 20, 20);
		g.drawString("Kode Tombol         : "+KodeTombol, 20, 50);
		g.drawString("Karakter Tombol     : "+KarakterTombol, 20, 70);
	}

	public void keyPressed(KeyEvent e) {
		KodeTombol=e.getKeyCode();
		KarakterTombol=e.getKeyChar();
		repaint();
	}
	public void keyReleased(KeyEvent e) {	}
	public void keyTyped(KeyEvent e) {  }

}
```

Kode program diatas akan menampilkan kode tombol dan karakter tombol yang kita tekan disetiap kita menekan tombol pada keyboard. Method clearRect() digunakan untuk menghapus tulisan ataupun gambar sebelumnya sedangkan method repaint() dipanggil ketika ada perubahan atau ketika tombol ditekan yang berfungsi untuk memanggil method paint().

### 3.5 Input dari Mouse

Mendengarkan action dari mouse sama prosesnya seperti mendengarkan action dari keyboard. Java menangani masukan dari mouse menggunakan aksi yang dihasilkan oleh Java Runtime Environment (JRE) dan mengirimkan ke program jika mengimplementasikan MouseListener.

Langkah pertama yang harus diambil untuk menangani aksi dari mouse di dalam program adalah memanggil kedua fungsi yang berikan oleh JRE yaitu addMouseListener() dan addMouseMotionListener(). Kedua fungsi tersebut harus diinialisasi, fungsi ini sama seperti yang telah dipelajari pada saat menangani keyboard.

```
addMouseListener(this);
addMouseMotionListener(this);
```

Java menyediakan sebuah class interface untuk pergerakan mouse dan aksi tombol yang hampir sama seperti interface pada keyboard. Class MouseListener adalah class abstrak yang disediakan kepada program dengan sebuah interface dengan 5(lima) method yang harus diimplementasi di dalam walupun tidak digunakan.

```
public void mouseClicked(MouseEvent  e)
public void mouseEntered(MouseEvent  e)
public void mouseExited(MouseEvent  e)
public void mousePressed(MouseEvent  e)
public void mouseReleased(MouseEvent  e)
```

Interface dari MouseListener menangani tombol mouse, posisi mouse di dalam window, dan lokasi mouse ketika kursor mouse kedalam dan keluar dari window.

Selain MouseListener juga terdapat class interface yang sangat berbeda, yang digunakan untuk menangani pergerakan mouse. Posisi mouse

dapat dibaca melalui MouseListener, tetapi untuk menerima aksi dari pergerakan mouse membutuhkan interface yang lain. Untuk menerima aksi pergerakan mouse melalui jendela window harus menggunakan interface MouseMotionListener. Ada 2(dua) aksi didalam interface ini yaitu :

```
public void mouseDragged(MouseEvent  e)
public void mouseMoved(MouseEvent  e)
```

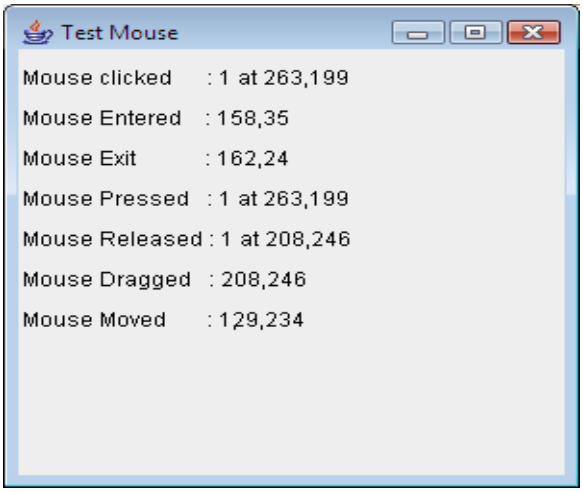
### 3.6 Mendeteksi Tombol Mouse

Beberapa aksi akan di beritahukan oleh JRE ketika tombol mouse diklik,ditekan, atau dilepaskan. Salah satu method yang yang tidak terhubung dengan tombol mouse adalah mouseEntered, mouseExited, dan mouseMoved, ketiga method itu terhubung ketika terjadi aksi terhadap posisi dan pergerakan. Semua method dari (mouseClicked, mouseEntered, mouseExited, mousePressed, dan mouseReleased) berjalan apabila tombol mouse ditekan.

Semua action hanya mempunyai 1 (satu) parameter yang disebut sebagai MouseEvent. Parameter ini sebenarnya adalah sebuh class, yang kemudian digunakan oleh JRE untuk mengirimkan action yang terjadi pada mouse. Pada class ini terdapat mthod untuk mengetahui posisi mouse dan nilai dari tombol mouse. Untuk nilai posisi mouse x dan y, dapat menggunakan MouseEvent.getX() dan mouseEvent.getY(). Parameter biasanya didefinisikan menjadi (MouseEvent  e), untuk itu pada prakteknya gunakan e.getX() dan e.getY() untuk mengetahui posisi dari mouse. Demikian juga, MouseEvent akan memberitahu tombol mana yang telah ditekan. Pada MouseEvent ada sebuah method getButton() yang fungsinya sama tergantung dari tombol yang telah ditekan. Method getButton() sangat berguna ketika ingin mendeteksi satu tombol yang ditekan.

```
public void mouseClicked(MouseEvent e) {
      if(e.getButton()== MouseEvent.BUTTON1){
            System.out.println("button 1");
      }else if(e.getButton()== MouseEvent.BUTTON2){
            System.out.println("button 2");
      }else if(e.getButton()== MouseEvent.BUTTON3){
            System.out.println("button 3");
      }
}
```

Class TestMouse dibawah ini memperlihatkan semua action dari mouse yang baru saja dipelajari. program ini menggunakan kode class Graphics, method drawString dan rangkaian varible untuk menampilkan status dari semua aksi mouse secara satu persatu. Gambar 3.13 dibawah ini menampilkan hasil dari kode program TestMouse.




Gambar 3.13 Tampilan Hasil, Semua Action dari Mouse

```
import java.awt.Graphics;
import java.awt.event.*;
import javax.swing.JFrame;

public class TestMouse extends JFrame implements
MouseListener,MouseMotionListener{

      int clickx, clicky;
      int pressx, pressy;
      int releasex, releasey;
      int enterx, entery;
      int exitx, exity;
      int dragx, dragy;
      int movex, movey;
      int mousebutton;
```

Konstruktor adalah method pertama yang dijalankan di dalam sebuah program ketika class tersebut dipanggil. Oleh sebab itu di method inilah dimana variabel dan object game diinialisasi, dan disini juga dapat ditempatkan listener untuk semua program peralatan masukan(Input Device) yang dibutuhkan. Apabila program tidak menghiraukan keyboard atau mouse, periksalah konstruktor utama untuk memastikan apakah listener telah ditambahkan atau tidak.

```
TestMouse(){
      setTitle("Test Mouse");
      addMouseListener(this); // listener tombol mouse
      addMouseMotionListener(this);// listener pergerakan mouse
      setSize(300, 200);
      setVisible(true);
      setDefaultCloseOperation(this.EXIT_ON_CLOSE);
}
```

Method paint() adalah method yang dipanggil ketika program merefresh jendela. Sejak paint() dengan parameter (Graphics g), object ini dapat digunakan untuk menggambar di dalam jendela program. Untuk menampilkan teks didalam program ini kami telah menggunakan method Graphics.drawString().

```
public void paint(Graphics g){
            g.setColor(Color.WHITE);
            g.clearRect(0, 0, 300, 200);
            g.setColor(Color.BLACK);
            g.drawString("Mouse clicked : "+mousebutton+" at
"+clickx+","+clicky, 10,50);
            g.drawString("Mouse Entered : "+enterx+","+entery, 10,
75);
            g.drawString("Mouse Exit    : "+exitx+","+exity, 10,
100);
            g.drawString("Mouse Pressed : "+mousebutton+" at
"+pressx+","+pressy, 10,125);
            g.drawString("Mouse Released: "+mousebutton+" at
"+releasex+","+releasey,10,150);
            g.drawString("Mouse Dragged : "+dragx+","+dragy, 10,
175);
            g.drawString("Mouse Moved   : "+movex+","+movey, 10,
200);
      }
```

Bagian kode selanjutnya adalah method checkButton(), adalah program yang kami tulis untuk menangani aksi pada mouse. Method chekButton() ini memeriksa tombol yang baru saja di tekan dan menyimpannya kedalam variabel (mousebutton) sebagai nilai yang merepresentasikan tombol yang ditekan.

```
public void checkButton(MouseEvent e){
      switch(e.getButton()){
            case MouseEvent.BUTTON1:
                  mousebutton=1;
                  break;
            case MouseEvent.BUTTON2:
                  mousebutton=2;
                  break;
            case MouseEvent.BUTTON3:
                  mousebutton=3;
                  break;
            default:
                  mousebutton=0;
      }
}
```

Method mouseClicked() adalah bagian dari interface MouseListener. Ketika mengimplementasi interface ini, harus mengikutsertakan semua action dari Mouse yang didefinisikan dalam interface ini atau akan mendapatkan pesan kesalahan(error). Action ini dipanggil kapanpun ketika menekan tombol mouse di dalam jendela window, artinya ketika menekan atau melepaskan tombol mouse. Action ini tidak biasanya dibutuhkan didalam mousePressed() dan mouseReleased() sendiri.

```
 public void mouseClicked(MouseEvent e) {
       // Menyimpan posisi mouse
       clickx=e.getX();
       clicky=e.getY();
       // Mengupdate Tombol
       checkButton(e);
       // refresh tampilan
       repaint();
 }
```

Selanjutnya adalah aksi kedua method mouse, mouseEntered() dan mouseExited() yang dipanggil ketika kursor mouse memasuki atau meninggalkan jendela aplikasi. Kedua aksi ini tidak sering dibutuhkan dalam pembuatan game.

```
public void mouseEntered(MouseEvent e) {
      enterx=e.getX();
      entery=e.getY();
      repaint();
}
public void mouseExited(MouseEvent e) {
      exitx=e.getX();
      exity=e.getY();
      repaint();
}
```

Method mousePressed() dan mouseReleased() akan dipanggil ketika menekan dan melepaskan tombol mouse. Ketika kedua aksi tersebut terjadi, posisi akhir dari mouse dapat diambil baik ketika ditekan atau dilepaskan.

```
public void mousePressed(MouseEvent e) {
      pressx=e.getX();
      pressy=e.getY();
      repaint();
}
public void mouseReleased(MouseEvent e) {
      releasex=e.getX();
      releasey=e.getY();
      repaint();
}
```

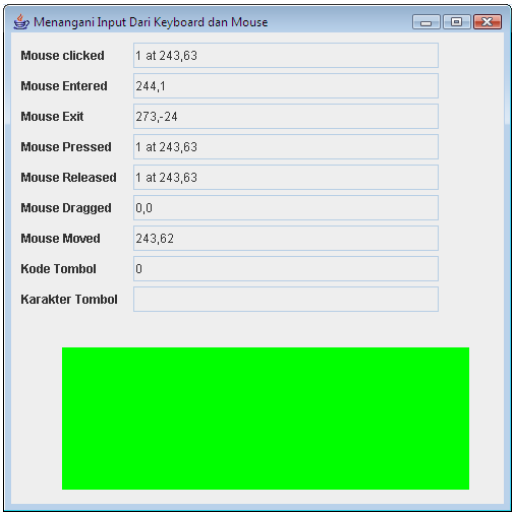
Interface MouseMotionListener mendefinisikan 2(dua) aksi yaitu mouseDragged() dan mouseMoved(). Kedua aksi tersebut sangat membantu ketika ingin mengetahui kapan mouse bergerak di dalam jendela window.

```
public void mouseDragged(MouseEvent e) {
      dragx=e.getX();
      dragy=e.getY();
      repaint();
}

public void mouseMoved(MouseEvent e) {
      movex=e.getX();
      movey=e.getY();
      repaint();
}
```

### 3.7 Membuat Class Untuk Handling Keyboard dan Mouse

Pada pokok bahasan sebelumnya penanganan aksi dari keyboard dan mouse selalu di dalam sebuah class yang sama, sekarang akan dibuat class sendiri untuk menangani aksi yang terjadi pada keyboard dan mouse yang mana akan mempermudah user untuk menggunakannya. Berikut adalah tampilan dari program yang dibuat.




Gambar 3.14 Tampilan hasil Class PercobaanKeyboardMouse

Pada Class PercobaanKeyboardMouse diatas menampilkan koor-dinat, tombol mouse beserta kode dan karakter tombol pada keyboard dengan menggunakan 2 buah panel, 9 buah label dan textfield.

```
JTextField tf[]=new JTextField[9];
JLabel l[]=new JLabel[9];
HandlingKeyboardMouse hkm;
String label[]={"Mouse clicked","Mouse Entered","Mouse Exit",
                "Mouse Pressed","Mouse Released","Mouse Dragged",
                "Mouse Moved","Kode Tombol","Karakter Tombol"};

.....................
.....................
.....................

for(int i=0;i<9;i++) {
      tf[i]=new JTextField(30);
      tf[i].setEditable(false);
      tf[i].setFocusable(false);
      l[i]=new JLabel(label[i]);

      p1.add(l[i]).reshape(10, i*30, 100, 25);
```

```
        p1.add(tf[i]).reshape(120, i*30, 300, 25);
}
```

Kode diatas mendefinisikan array textfield dan label sebanyak 9 buah kemudian memberikan nilai false pada method setEditable() agar tidak dapat di edit dan method setFocusable() supaya kursor tidak terarah langsung pada textfield ketika program di jalankan. Selanjutnya membuat object yang digunakan untuk menangani keyboard dan mouse dengan melewatkan textfield yang di buat tadi dari konstruktornya.

```
hkm=new HandlingKeyboardMouse(tf);
p2.addMouseListener(hkm);
p2.addMouseMotionListener(hkm);
addKeyListener(hkm);
```

Berikut ini adalah konstruktor dari class HandlingKeyboardMouse.

```
HandlingKeyboardMouse(JTextField a[]){
        for(int i=0;i<9;i++){
                tf[i]=new JTextField(30);
                tf[i]=a[i];
        }
}
```

Textfield yang dilewatkan pada konstruktor HandlingKeyboardMouse diatas kemudian diproses sesuai kebutuhan dimana pada class ini texfield tersebut digunakan untuk menampilkan koordinat pointer pada mouse juga karakter dan kode tombol pada keyboard yang berada di dalam methode update(), mothod ini selalu di panggil ketika ada aksi yang terjadi pada keyboard ataupun mouse. Berikut adalah potongan kode programnya.

```
public void update(){
        tf[0].setText(mousebutton+" at "+clickx+","+clicky);
        tf[1].setText(enterx+","+entery);
        tf[2].setText(exitx+","+exity);
        tf[3].setText(mousebutton+" at "+pressx+","+pressy);
        tf[4].setText(mousebutton+" at "+releasex+","+releasey);
        tf[5].setText(dragx+","+dragy);
        tf[6].setText(movex+","+movey);
        tf[7].setText(""+KodeTombol);
        tf[8].setText(""+KarakterTombol);
}
```

```
.....
.....
public void mouseMoved(MouseEvent e) {
      movex=e.getX();
      movey=e.getY();
      update();
}
.....
.....
public void keyPressed(KeyEvent e) {
      KodeTombol=e.getKeyCode();
      KarakterTombol=e.getKeyChar();
      update();
}
```