

Java GUI by Example

Created as supplement material for Class of Object Oriented Programming

Kautsar Sophan
University of Trunojoyo
Madura
Mei 2018

Daftar Isi

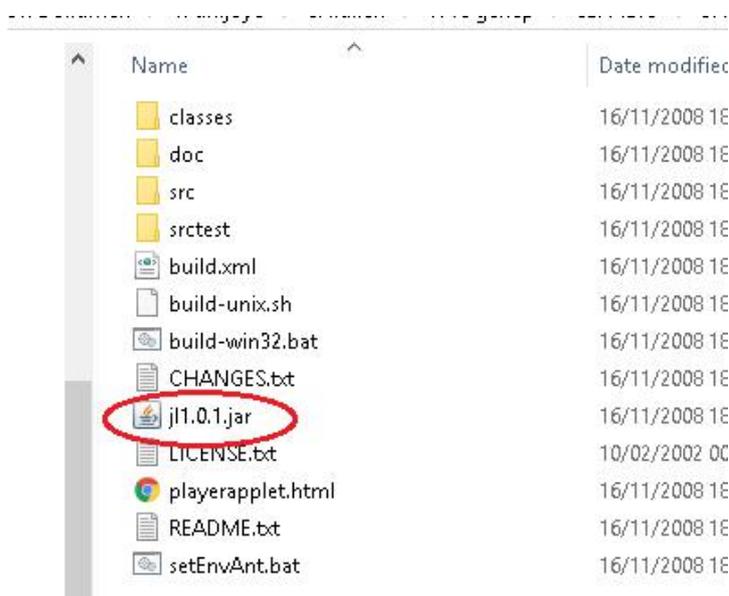
Daftar Isi.....	2
Memainkan file MP3.....	3
Memainkan file video mpg.....	7
Menggambar Garis.....	10
Menggerakkan tombol dengan kursor.....	13
Klik Picture.....	15

Memainkan file MP3

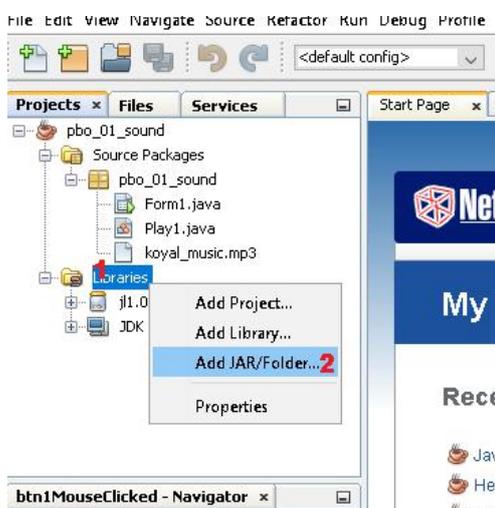
Pada contoh berikut, kita akan membuat aplikasi java untuk memainkan file mp3. Kita memanfaatkan library JLayer - mp3 library yang bisa kita download di <http://www.javazoom.net/javalayer/sources.html>

Langkah

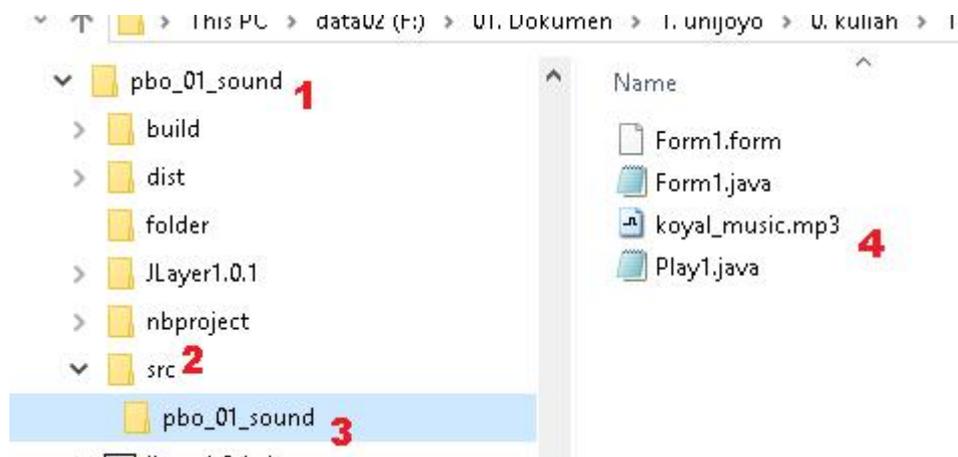
1. Buat project baru di netbeand
2. Download library jLayer di <http://www.javazoom.net/javalayer/sources.html>
3. Hasil download adalah file zip, lakukan extact file tersebut. Didalam folder tersebut terdapat file jl1.0.1.jar, seperti pada gambar berikut



4. Dari netbean, dari project anda, di bagian folder library, klik kanan, pilih 'add jar/folder' seperti pada contoh berikut



5. Selanjutnya, pilih file library **jl1.0.1.jar**
6. Sebelumnya, kita menyiapkan file mp3 yang akan dijalankan. File mp3 harus di letakkan di bawah folder src, yang ada di dalam folder project kita, seperti pada contoh berikut



7. Untuk menjalankan file mp3, kita membuat method play, dan untuk memberhentikan file mp3, kita menggunakan method stop. Buat class baru dengan nama bebas (dalam contoh ini, dibuat class dengan nama '**Play1.java**'), kemudian tambahkan method play dan close seperti pada contoh berikut

```

private static String filename;
private static Player player;
public static void play() {
    try {
        FileInputStream fis =
            new FileInputStream(
                "src/pbo_01_sound/koyal_music.mp3");
        BufferedInputStream bis = new BufferedInputStream(fis);
        player = new Player(bis);
    }
    catch (Exception e) {
        System.out.println("Problem playing file " + filename);
        System.out.println(e);
    }

    // run in new thread to play in background
    new Thread() {
        @Override
        public void run() {
            try { player.play();
                //player.play();
            }
            catch (Exception e) { System.out.println(e); }
        }
    }.start();

}

public static void close()
{
    player.close();
}

```

8. Pada bagian

```

FileInputStream fis =
    new FileInputStream(
        "src/pbo_01_sound/koyal_music.mp3");

```

Kita perhatikan fileInputStrem berisi parameter yang menunjukkan folder file mp3 yang akan di jalankan.

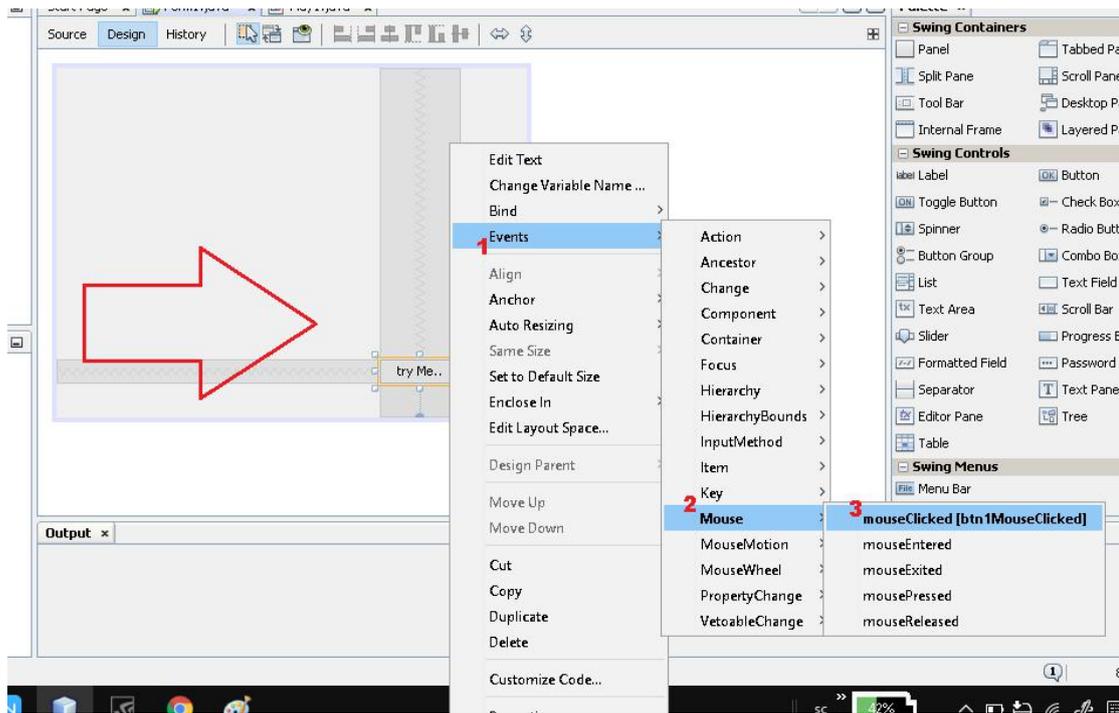
9. Pada bagian atas class, jangan lupa juga diimport libaray yang berkaitan, seperti pada contoh berikut

```

import java.io.BufferedInputStream;
import java.io.FileInputStream;
import javazoom.jl.player.Player;

```

10. Selanjutnya, buat class baru JFrame yang akan berisi Aplikasi GUI kita. Siapkan tombol dengan layout seperti berikut



11. Pada bagian event mouse click, kita siapkan listing program

```
private void btn1MouseClicked(java.awt.event.MouseEvent evt) {
    if (isplay == false)
    {
        Play1.play();
        isPlay = true;
    }
    else
    {
        Play1.close();
        isPlay = false;
    }
}
```

pada listing diatas, ada instance variabel isPlay yang di inialisasi dengan nilai false. Variabel ini sebagai penanda apakah mp3 nya sedang posisi play atau tidak. Jika mp3 tidak sedang prosisi play, maka file mp3 di play, begitu juga sebaliknya.

12. Selanjutnya, class GUI bisa dijalankan dan di test.

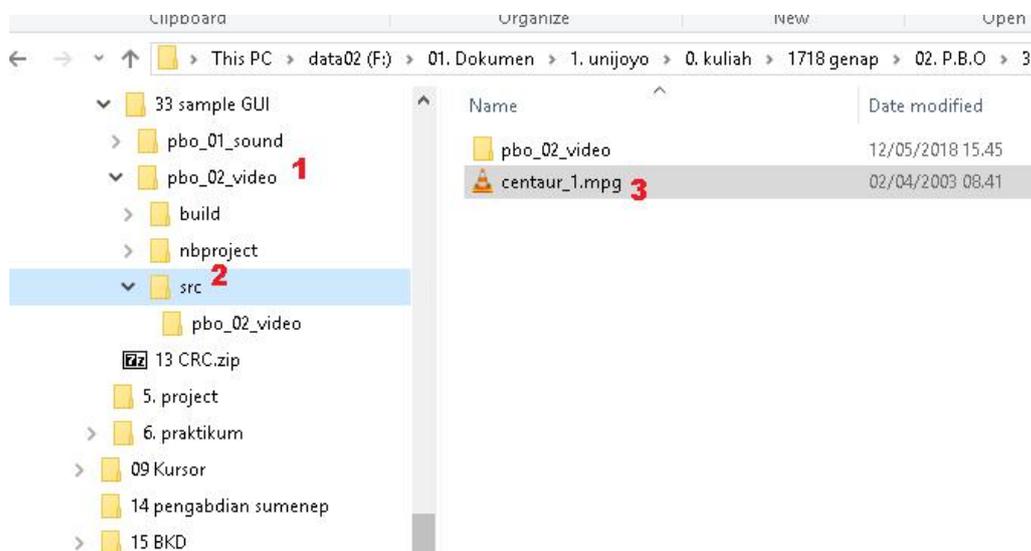
Memainkan file video mpg

Pada contoh berikut, kita akan membuat aplikasi GUI untuk memainkan file video dengan ekstensi mpg. Pada contoh ini, kita menggunakan library JMF (Java Media Framework) API, di

<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-140239.html>

Langkah

1. Buat project baru
2. Download JMF, dan install
3. Untuk file yang akan kita play, kita letakkan di dalam folde src, seperti pada contoh berikut



4. Buat class baru, dalam contoh ini diberi nama **Play1.java**. Class ini akan berisi program untuk memanggil JMF
5. Listing untuk import library di class ini

```
import java.awt.BorderLayout;
import java.awt.Component;
import java.net.MalformedURLException;
import java.net.URL;
import javax.media.Manager;
import javax.media.MediaLocator;
import javax.media.Player;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

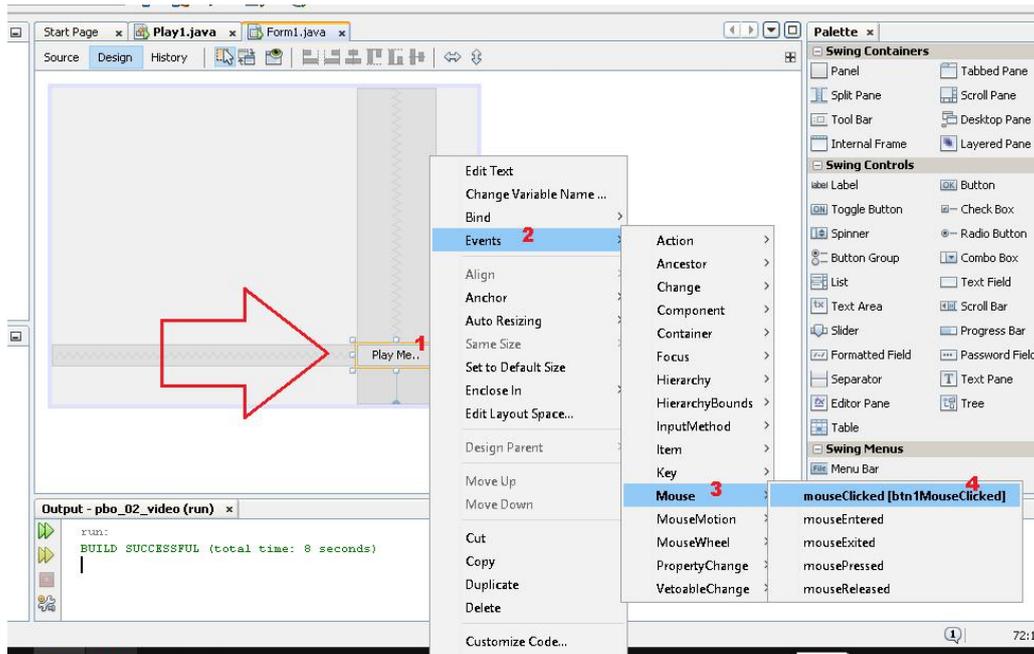
6. Selanjutnya, kita buat constructor dengan parameter URL. URL ini akan mengacu ke file mpg yang akan di jalankan. Listingnya

```

public Play1(URL mediaUrl) {
    //initComponents();
    setLayout(new BorderLayout());
    try {
        Player mediaPlayer=Manager.createRealizedPlayer(
            new MediaLocator(mediaUrl));
        Component video=mediaPlayer.getVisualComponent();
        Component control=mediaPlayer.getControlPanelComponent();
        if (video!=null) {
            add(video, BorderLayout.CENTER);
        }
        add(control, BorderLayout.SOUTH);
        mediaPlayer.start();
    } catch (Exception e) {
    }
}
}

```

7. Selanjutnya, untuk GUI kita buat class JFrame dengan 1 tombol untuk menjalankan videonya. Video yang dijalankan di dalam class JFrame baru. Contoh tampilan GUI tampak seperti pada gambar berikut



8. Listing untuk event mouse click di btn1 sebagai berikut

```

private void btnMouseClicked(java.awt.event.MouseEvent evt) {
    URL mediaUrl = null;
    try {
        mediaUrl = new URL(new URL("file:"), "./src/centaur_1.mpg");
    } catch (MalformedURLException ex) {
        Logger.getLogger(Form1.class.getName()).log(Level.SEVERE, null, ex);
    }

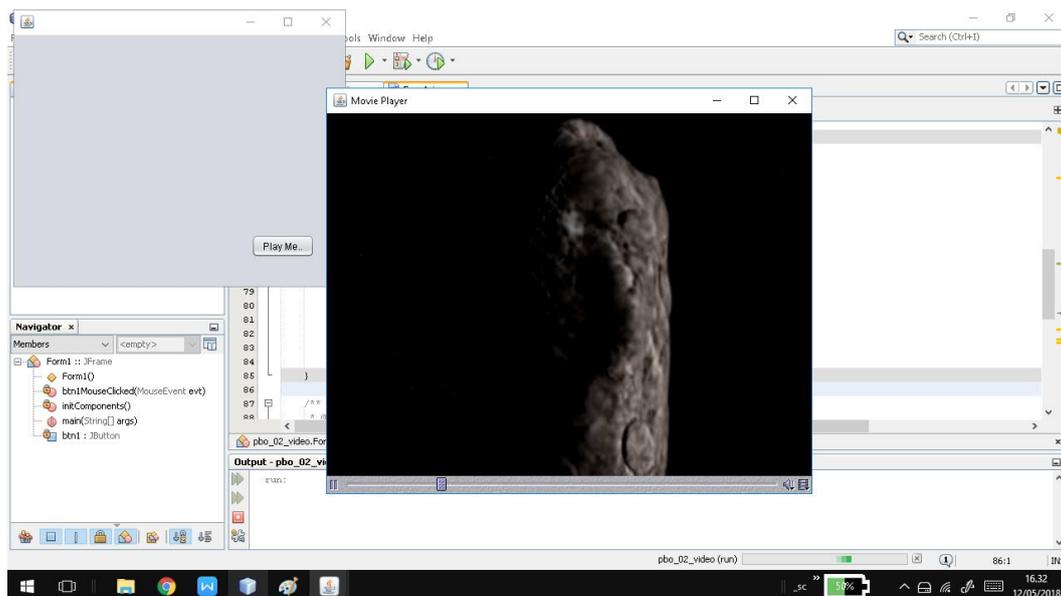
    JFrame mediaTest = new JFrame( "Movie Player" );
    mediaTest.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
    Playl mediaPanel = new Playl( mediaUrl );

    mediaTest.add( mediaPanel );
    mediaTest.setSize( 600, 500 ); // set the size of the player
    mediaTest.setLocationRelativeTo(null);
    mediaTest.setVisible( true );
}

```

9. Di bagian mediaURL, itu menunjukkan lokasi file mpg yang akan dijalankan.

10. Contoh hasil running tampak pada gambar berikut



Menggambar Garis

Untuk menggambar di java, kita memanfaatkan library graphics. Dalam contoh berikut, kita menggambar dengan menggunakan 2 metode, yang pertama menggambar dengan di JPanel, dan yang kedua kita menggambar di JFrame.

Langkah

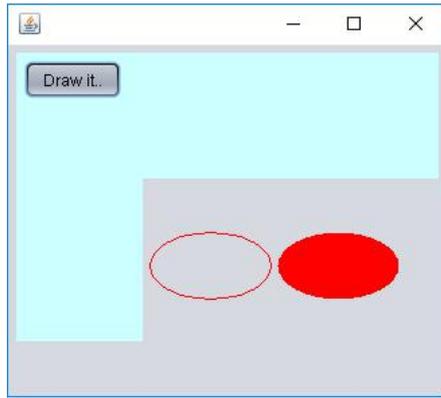
1. Buat project baru di netbeans
2. Buat class java dengan nama **Draw1.java**. Pada class ini, kita extends dari class JPanel.
 - a) Buat konstruktor. Pada bagian ini kita menentukan ukuran panel, dan lokasi panel
 - b) Kita buat metode override paint. Method ini harus kita buat, pada bagian ini kita tuliskan grafik yang akan kita gambarkan. Pada contoh berikut, kita menggambar oval. Listing lengkap tampak pada gambar berikut

```
public class Draw1 extends JPanel {  
    public Draw1()  
    {  
        setSize (400,165);  
        setLocation(100,100);  
        setVisible (true);  
    }  
  
    @Override  
    public void paint(Graphics graph)  
    {  
        super.paint (graph);  
        graph.setColor(Color.RED);  
        graph.drawOval (5, 40, 90, 50);  
        graph.fillOval(100, 40, 90, 50);  
    }  
}
```

3. Selanjutnya, kita buat class GUI nya, dengan button dan event mouse kliknya.
4. Untuk menampilkan gambar, kita cukup membuat object dari class **Draw1**, kemudian objek ini kita letakkan di GUI yang telah kita buat, dengan listing berikut

```
private void btnMouseClicked(java.awt.event.MouseEvent evt) {  
    Draw1 test1 = new Draw1();  
    this.setContentPane(test1);  
}
```

- a) Hasil running tampak pada contoh berikut



5. Pada contoh berikutnya, kita akan membuat gambar namun dituliskan pada class JFrame.

6. Buat class baru, pada contoh berikut kita beri nama **Draw2.java**

- a) Pada class ini kita extends pada class JFrame. JFrame adalah class yang berupa window / forms, sehingga, jika kita extends dari class JFrame, maka hasil gambar kita bisa langsung tampil.
- b) Pada contoh ini kita menggambar 2 garis, dan kemudian beberapa garis dengan menggunakan for
- c) Contoh listing dari class ini tampak pada gambar berikut

```
public class Draw2 extends JFrame{
    public Draw2()
    {
        super ("Menggambar garis");
        setSize (400,165);
        setVisible (true);
    }

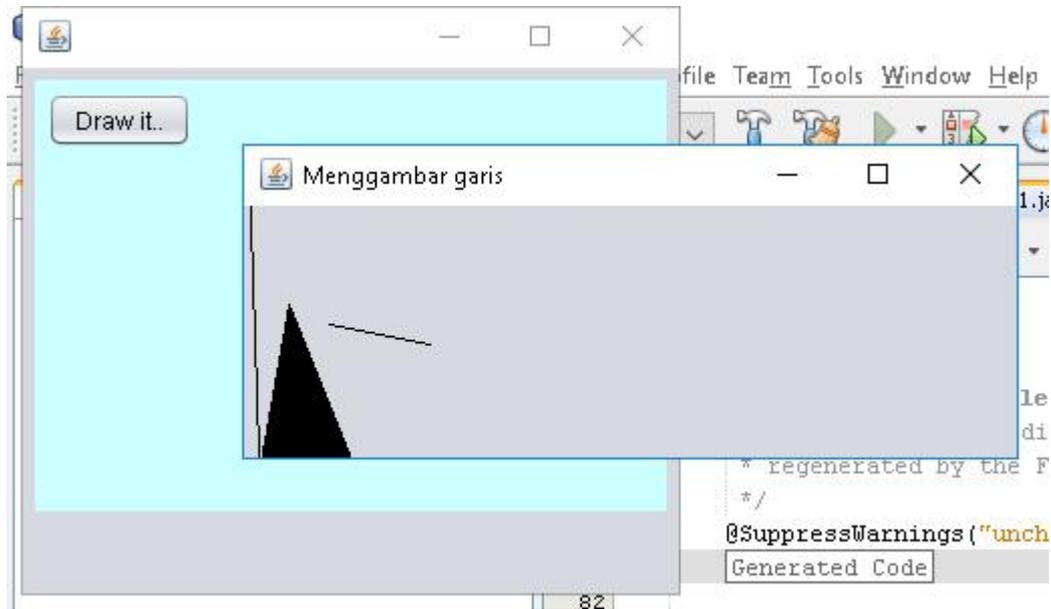
    @Override
    public void paint (Graphics g)
    {
        super.paint (g);
        int h = 0;
        g.drawLine(50,90,100,100);
        g.drawLine(10, 10, 20, 300);

        for (int i = 0; i < 100; i += 2) {
            g.drawLine(30, 80, i, 250);
        }
    }
}
```

- d) Selanjutnya, kita buat GUI dengan tombol dan event mouse clicknya. Scriptnya cukup membuat object dari class **Draw2**.
- e) Listing program di GUI tampak pada contoh berikut

```
private void btnMouseClicked(java.awt.event.MouseEvent evt) {  
    Draw2 test1 = new Draw2();  
}
```

f) Hasil running tampak pada contoh berikut



7. Dari contoh 2 cara tersebut bisa dikembangkan untuk membuat gambar di Java.

Menggerakkan tombol dengan kursor

Pada contoh ini kita akan menggerakkan tombol dengan menggunakan kursor panah. Program akan kita set agar selalu focus di tombol, kemudian kita akan merespon tombol yang ditekan melalui event keyPressed.

Posisi tombol kita ubah melalui nilai x dan y.

Langkah

1. Buat project baru di netbeans
2. Buat class JFrame
3. Buat masukkan tombol dari class JButton. Dalam contoh ini diberi nama **btn1**.
4. Pada event windowOpened, kita pastikan bahwa btn1 selalu mendapat focus / selalu terpilih. Kita menggunakan perintah requestFocus, seperti pada contoh berikut

```
private void formWindowOpened(java.awt.event.WindowEvent evt) {  
    btn1.requestFocus();  
}
```

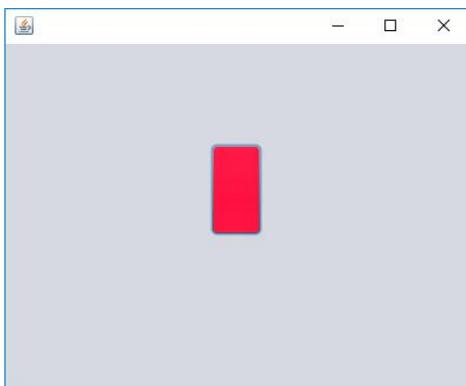
5. Selanjutnya, di event keyPressed, kita menangkap kode keyboard yang ditekan, kemudian di cek apakah panah atas, bawah, kiri, atau kanan. Selanjutnya, posisi btn1 kita sesuaikan sesuai dengan panah yang ditekan. Listing untuk event ini tampak pada contoh berikut

```

private void btnKeyPressed(java.awt.event.KeyEvent evt) {
    int a = evt.getKeyCode();
    int x, y;
    x = btnl.getX();
    y = btnl.getY();
    switch (a)
    {
        case KeyEvent.VK_UP:
            btnl.setLocation(x, y-10);
            break;
        case KeyEvent.VK_DOWN:
            btnl.setLocation(x, y+10);
            break;
        case KeyEvent.VK_LEFT:
            btnl.setLocation(x-10, y);
            break;
        case KeyEvent.VK_RIGHT:
            btnl.setLocation(x+10, y);
            break;
        default:
            break;
    }
    btnl.requestFocus();
}
}

```

6. Hasil running tampak pada contoh berikut



tombol akan bergerak ke atas / bawah / kanan / kiri sesuai dengan tombol yang di tekan.

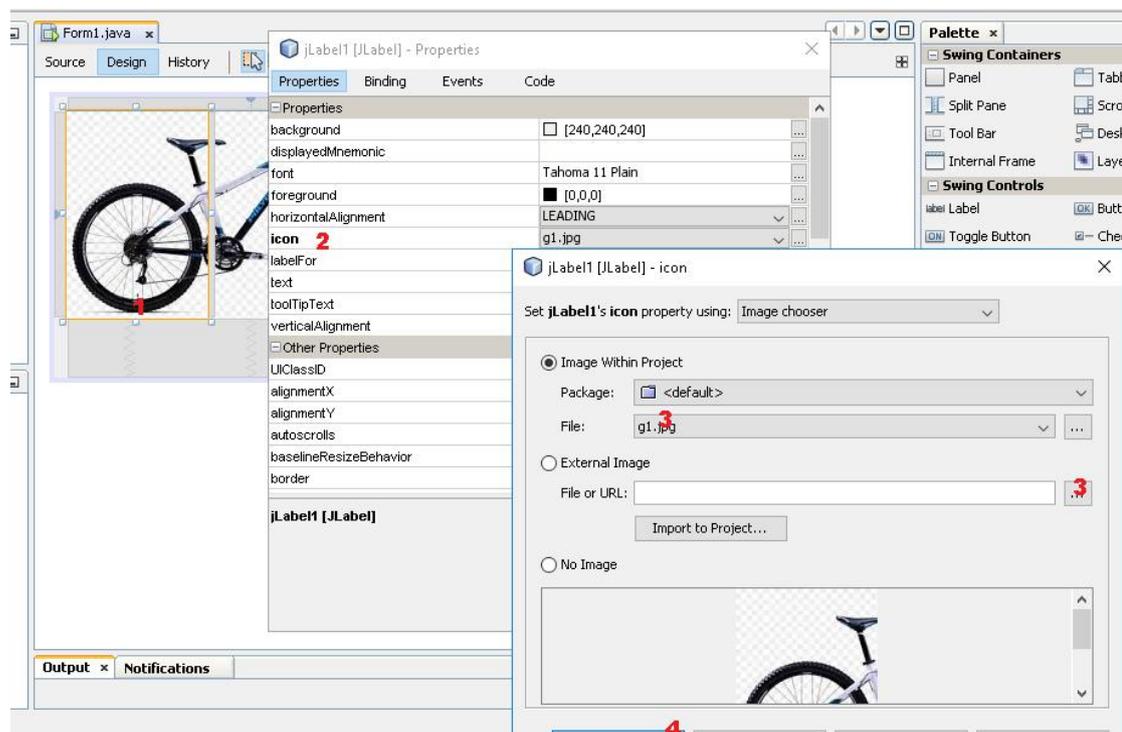
Klik Picture

Pada program ini kita akan membuat aplikasi yang menampilkan sebuah gambar, dan kemudian kita bisa mengklik bagian gambar tertentu, kemudian nanti bisa muncul informasi sesuai dengan gambar yang diklik.

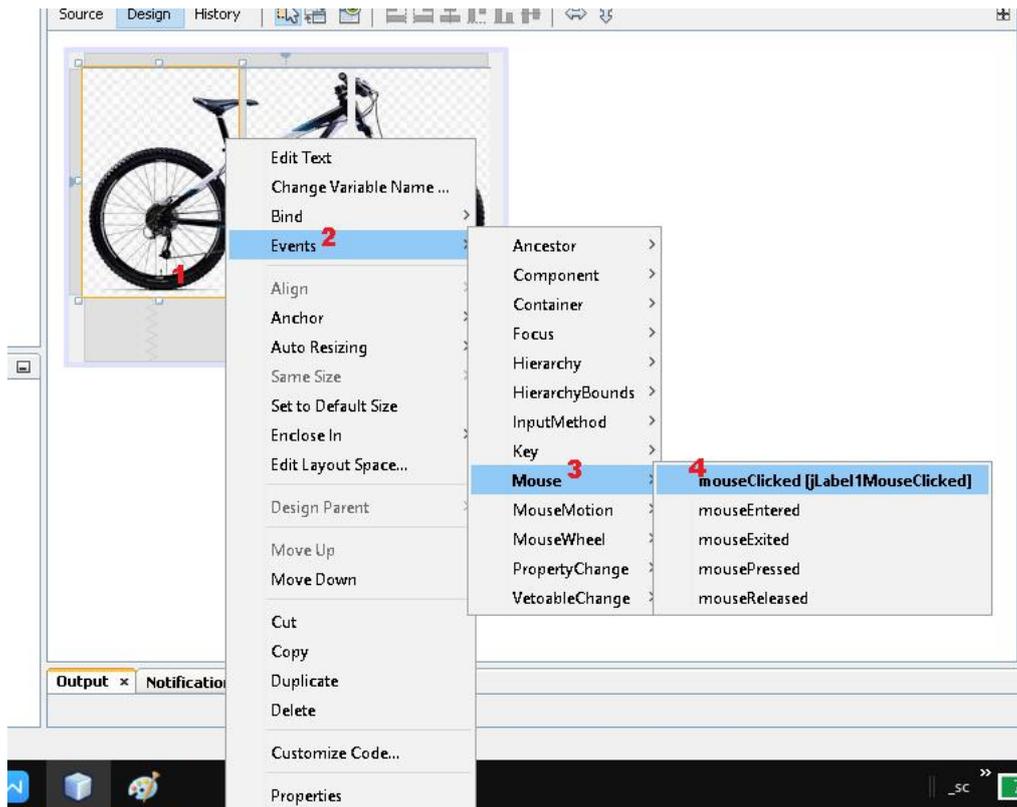
Untuk menampilkan gambar, kita bisa menggunakan button ataupun toggle button ataupun label, dan gambar yang ditampilkan bisa kita set dari properties icon yang ada di komponen tersebut.

Langkah

1. Buat project baru
2. Buat class dari JFrame
3. Masukkan JLabel. Untuk memberikan gambar, buka propertis, dan klik bagian icon, kemudian pilih gambar yang akan dimasukkan, seperti pada contoh berikut



4. Kita bisa memasukkan beberapa gambar di label yang terpisah.
5. Selanjutnya, dari label yang sudah ada gambarnya, kita bisa masukkan event mouseclick, dan kemudian memberikan program untuk menampilkan informasi tertentu yang berkaitan dengan gambar tersebut. Informasi bisa berupa textual, suara, atau membuka frame baru.
6. Contoh listing untuk mouseclick di label1 tampak pada gambar berikut



```
private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
    JOptionPane.showMessageDialog(this, "something....");
}
```

7. Contoh hasil running tampak pada gambar berikut

