



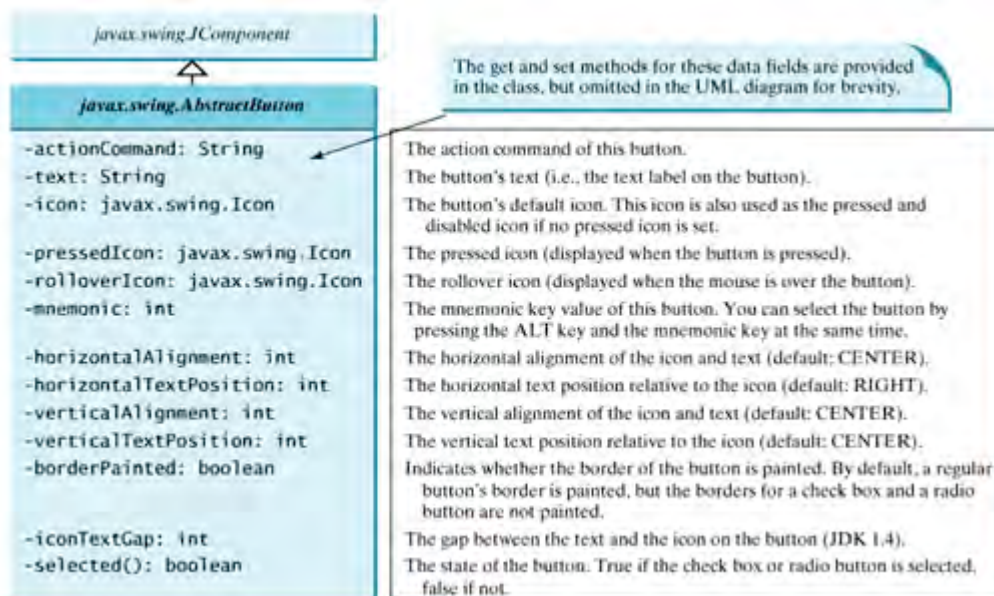
A. Kompetensi

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu menggunakan class JButton dengan berbagai Event Handling.

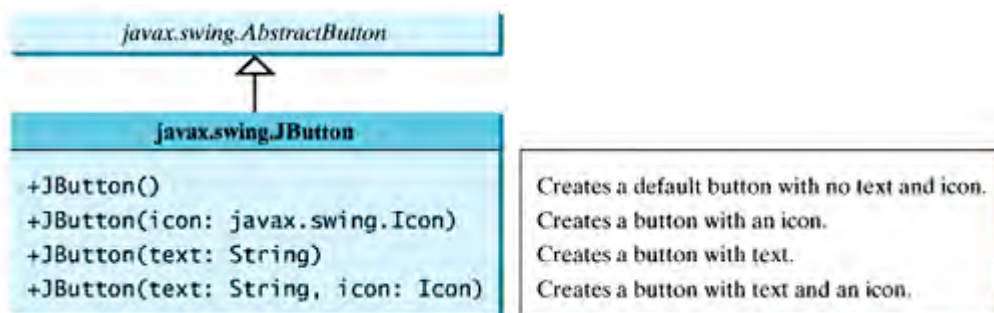
B. Dasar Teori

1. Button

Button adalah sebuah komponen yang akan *trigger* sebuah *action event* ketika di klik. Swing menyediakan banyak jenis button. Semua kelas button adalah *child* dari kelas *AbstractButton*



Pada praktikum kali ini kita akan menggunakan kelas **JButton**



a. Icon, Pressed Icon, dan Rollover Icon

Sebuah button memiliki default icon, pressed icon, dan rollover icon. Pada umumnya kita menggunakan default icon, icon yang lain kita gunakan sebagai *special effect*. Pressed Icon akan ditampilkan saat button ditekan. Rollover Icon akan ditampilkan saat pointer mouse berada di atas button. Pada listing di



bawah ini default iconnya adalah bendera amerika, pressed iconnya menggunakan bendera kanda, dan rollover iconnya menggunakan bendera Inggris.

```
1 import javax.swing.*;
2
3 public class TestButtonIcons extends JFrame {
4     public static void main(String[] args) {
5         // Create a frame and set its properties
6         JFrame frame = new TestButtonIcons();
7         frame.setTitle("ButtonIcons");
8         frame.setSize(200, 100);
9         frame.setLocationRelativeTo(null); // Center the frame
10        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11        frame.setVisible(true);
12    }
13
14    public TestButtonIcons() {
15        ImageIcon usIcon = new ImageIcon("image/usIcon.gif");
16        ImageIcon caIcon = new ImageIcon("image/caIcon.gif");
17        ImageIcon ukIcon = new ImageIcon("image/ukIcon.gif");
18
19        JButton jbt = new JButton("Click it", usIcon);
20        jbt.setPressedIcon(caIcon);
21        jbt.setRolloverIcon(ukIcon);
22
23        add(jbt);
24    }
25 }
```



(a) Default icon



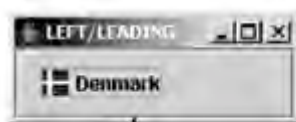
(b) Pressed icon



(c) Rollover icon

b. Alignment

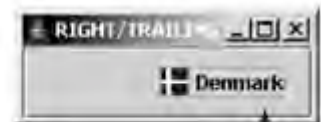
Horizontal alignment menentukan posisi text dan icon secara horizontal pada sebuah button. Method untuk mengeset *horizontal alignment* adalah **setHorizontalAlignment(int)** dengan parameternya merupakan salah satu dari lima variable konstan **LEADING, LEFT, CENTER, RIGHT, TRAILING**.



Horizontally left



Horizontally center



Horizontally right

Vertical alignment menentukan posisi text dan icon secara vertical pada sebuah button. Method untuk mengeset vertical alignment adalah **setVerticalAlignment(int)** dengan parameternya merupakan salah satu dari tiga variable konstan **TOP, CENTER, BOTTOM**.

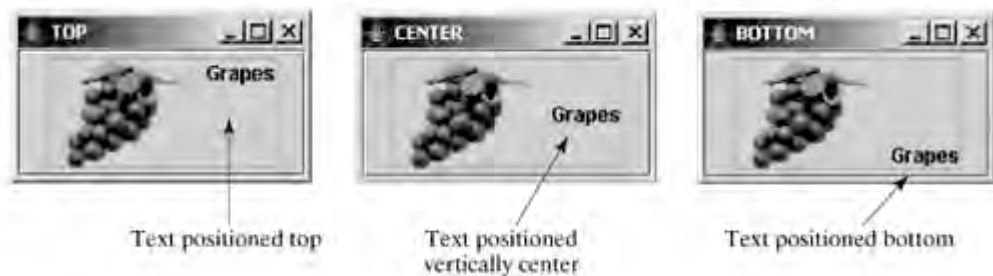


c. Text Position

Horizontal text position menentukan posisi text relative terhadap posisi icon secara horizontal. Method untuk mengeset horizontal text position adalah **setHorizontalTextPosition(int)** dengan parameternya merupakan salah satu dari lima variable konstan **LEADING, LEFT, CENTER, RIGHT, TRAILING**.



Vertical text alignment menentukan posisi text relative terhadap posisi icon secara vertical pada sebuah button. Method untuk mengeset vertical text alignment adalah **setVerticalTextAlignment(int)** dengan parameternya merupakan salah satu dari tiga variable konstan **TOP, CENTER, BOTTOM**.



2. Layout Manager

JFrame adalah sebuah *container* yang bisa memuat komponen-komponen lain. Kita bisa meletakkan button ke dalam sebuah frame. Setiap *container* menggunakan objek *layout manager* untuk menyusun tata letak komponen di dalam *container*.





FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA

LAB SHEET PEMROGRAMAN 2

Semester 2	Button dan Event Handling	4 x 50 mnt
No. LST/EKA/PTI208/07	Revisi : 02	April 2010
		Hal 4 dari 9

- (a) adalah sebuah frame yang memuat sebuah button dengan default layout. Jika kita tidak menspesifikasikan sebuah layout manager, maka default layout manager akan digunakan

```
1 import javax.swing.*;
2
3 public class HoldComponents {
4     public static void main(String[] args) {
5         JFrame frame = new JFrame();
6
7         // Add a button to frame
8         JButton jbtOK = new JButton("OK");
9         frame.add(jbtOK);
10
11        frame.setTitle("Window 1");
12        frame.setSize(200, 150);
13        frame.setLocation(200, 100);
14        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
15        frame.setVisible(true);
16    }
17 }
```

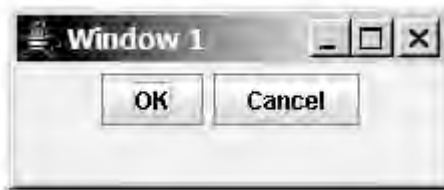
- (b) sebuah frame yang memuat sebuah button menggunakan flow layout manager.

```
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class UseFlowLayout {
5     public static void main(String[] args) {
6         JFrame frame = new JFrame();
7
8         // Set FlowLayout for the frame
9         FlowLayout layout = new FlowLayout();
10        frame.setLayout(layout);
11
12        // Add a button to frame
13        JButton jbtOK = new JButton("OK");
14        frame.add(jbtOK);
15
16        [Page 291]
17        frame.setTitle("Window 1");
18        frame.setSize(200, 150);
19        frame.setLocation(200, 100);
20        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21        frame.setVisible(true);
22    }
23 }
```

Contoh dua buah button menggunakan flow layout manager



```
1 import javax.swing.*;
2 import java.awt.*;
3
4 public class TwoButtons {
5     public static void main(String[] args) {
6         JFrame frame = new JFrame();
7
8         // Set FlowLayout for the frame
9         FlowLayout layout = new FlowLayout();
10        frame.setLayout(layout);
11
12        // Add two buttons to frame
13        JButton jbtOK = new JButton("OK");
14        JButton jbtCancel = new JButton("Cancel");
15        frame.add(jbtOK);
16        frame.add(jbtCancel);
17
18        frame.setTitle("Window 1");
19        frame.setSize(200, 150);
20        frame.setLocation(200, 100);
21        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22        frame.setVisible(true);
23    }
24 }
```



3. Delegation Event Model

Delegasi event model menguraikan bagaimana program Anda dapat merespon interaksi dari user. Untuk memahami model, pertama-tama mari kita pelajari melalui tiga komponen utamanya.

1. Event Source

Event source mengacu pada komponen GUI yang meng-generate event. Sebagai contoh, jika user menekan tombol, event source dalam hal ini adalah tombol.

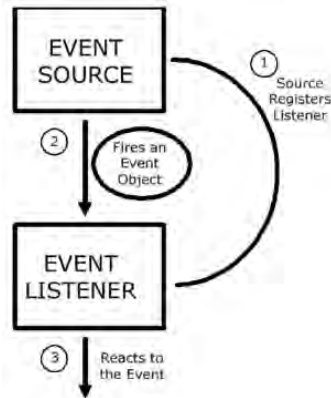
2. Event Listener/Handler

Event listener menerima berita dari event-event dan proses-proses interaksi user. Ketika tombol ditekan, listener akan mengendalikan dengan menampilkan sebuah informasi yang berguna untuk user.

3. Event Object

Ketika sebuah event terjadi (misal, ketika user berinteraksi dengan komponen GUI), sebuah object event diciptakan. Object berisi semua informasi yang perlu tentang event yang telah terjadi. Informasi meliputi tipe dari event yang telah terjadi, seperti ketika mouse telah di-klik. Ada beberapa class event untuk kategori yang berbeda dari user action. Sebuah event object mempunyai tipe data mengenai salah satu dari class ini.

Di bawah ini adalah *delegation event model*.



Berikut ini adalah tabel tipe-tipe event, listener yang bersangkutan dan method dari listener tersebut.

Events, Event Listeners, and Listener Methods

Event Class (Handlers)	Listener Interface	Listener Methods
<code>ActionEvent</code>	<code>ActionListener</code>	<code>actionPerformed(ActionEvent)</code>
<code>ItemEvent</code>	<code>ItemListener</code>	<code>itemStateChanged(ItemEvent)</code>
<code>MouseEvent</code>	<code>MouseListener</code>	<code>mousePressed(MouseEvent)</code> <code>mouseReleased(MouseEvent)</code> <code>mouseEntered(MouseEvent)</code> <code>mouseExited(MouseEvent)</code> <code>mouseClicked(MouseEvent)</code>
	<code>MouseMotionListener</code>	<code>mouseDragged(MouseEvent)</code> <code>mouseMoved(MouseEvent)</code>
<code>KeyEvent</code>	<code>KeyListener</code>	<code>keyPressed(KeyEvent)</code> <code>keyReleased(KeyEvent)</code> <code>keyTyped(KeyEvent)</code>
<code>WindowEvent</code>	<code>WindowListener</code>	<code>windowClosing(WindowEvent)</code> <code>windowOpened(WindowEvent)</code> <code>windowIconified(WindowEvent)</code> <code>windowDeiconified(WindowEvent)</code> <code>windowClosed(WindowEvent)</code> <code>windowActivated(WindowEvent)</code> <code>windowDeactivated(WindowEvent)</code>
<code>ContainerEvent</code>	<code>ContainerListener</code>	<code>componentAdded(ContainerEvent)</code> <code>componentRemoved(ContainerEvent)</code>

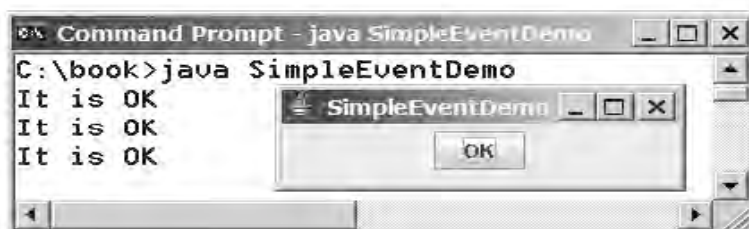


Events, Event Listeners, and Listener Methods

Event Class (Handlers)	Listener Interface	Listener Methods
<code>ComponentEvent</code>	<code>ComponentListener</code>	<code>componentMoved(ComponentEvent)</code> <code>componentHidden(ComponentEvent)</code> <code>componentResized(ComponentEvent)</code> <code>componentShown(ComponentEvent)</code>
<code>FocusEvent</code>	<code>FocusListener</code>	<code>focusGained(FocusEvent)</code> <code>focusLost(FocusEvent)</code>
<code>AdjustmentEvent</code>	<code>AdjustmentListener</code>	<code>adjustmentValueChanged(AdjustmentEvent)</code>

Contoh Program

```
1 import javax.swing.*;
2 import java.awt.event.*;
3 import java.awt.*;
4
5 public class SimpleEventDemo extends JFrame {
6     public SimpleEventDemo() {
7         JButton jbtOK = new JButton("OK");
8         setLayout(new FlowLayout());
9         add(jbtOK);
10
11         ActionListener listener = new OKListener();
12         jbtOK.addActionListener(listener);
13     }
14
15     /** Main method */
16     public static void main(String[] args) {
17         JFrame frame = new SimpleEventDemo();
18         frame.setTitle("SimpleEventDemo");
19         frame.setLocationRelativeTo(null); // Center the frame
20         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         frame.setSize(220, 80);
22         frame.setVisible(true);
23     }
24 }
25
26 class OKListener implements ActionListener {
27     public void actionPerformed(ActionEvent e) {
28         System.out.println("It is OK");
29     }
30 }
```



Pesan akan ditampilkan saat button ditekan

C. Alat/ Bahan

1. Lab. Sheet Pemrograman no 7



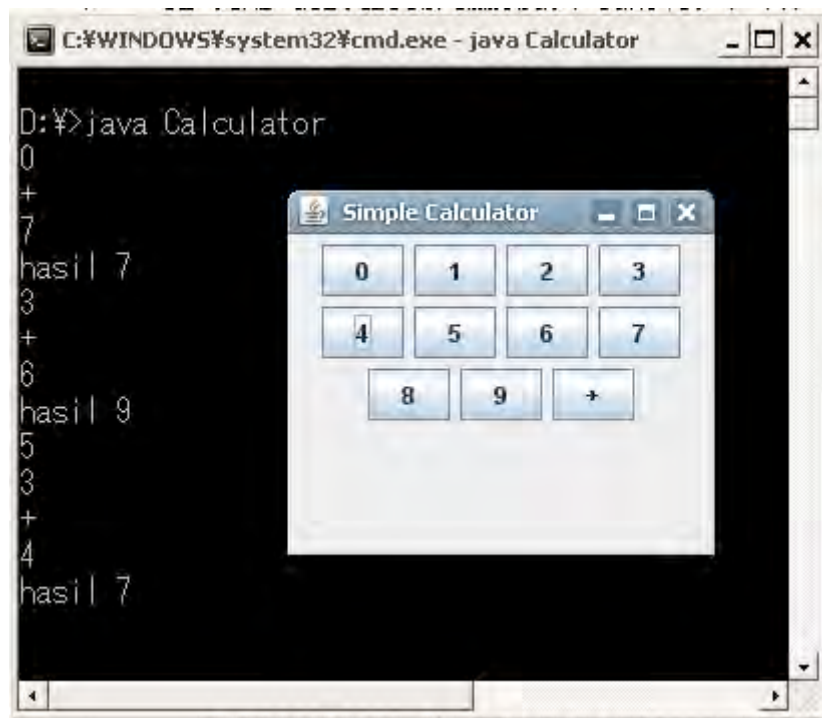
2. PC / Laptop with OS installed
3. JDK 1.5 or latest
4. J-Creator or text editor

D. Langkah Kerja

1. Baca dan pahami dasar teori di atas.
2. Lakukan kompilasi dan eksekusi terhadap contoh-contoh source code atau program yang ada di dasar teori dan LAMPIRAN.
3. Kerjakan tugas individu di bawah.


E. Tugas Individu

Buatlah sebuah calculator sederhana, calculator ini melakukan operasi penjumlahan satu digit. Angka yang dijumlahkan adalah angka sebelum tombol “+” ditekan dengan angka setelah tombol “+” ditekan. Hasil penjumlahan akan tampil setelah tombol angka penjumlah ditekan.



F. Lampiran

- **CH 57 & CH 59** (Introduction to Computer Science using Java, Java 5.0 version, January 2006, Bradley Kjell, Central Connecticut State University <http://chortle.ccsu.edu/CS151/cs151java.html>)

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA			
	LAB SHEET PEMROGRAMAN 2			
	Semester 2	Button dan Event Handling		4 x 50 mnt
	No. LST/EKA/PT1208/07	Revisi : 02	April 2010	Hal 9 dari 9

- **Java™ Swing, 2nd Edition**, Brian Cole, Robert Eckstein, James Elliott, Marc Loy, David Wood

Dibuat oleh : Herman DS	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Fakultas Teknik Universitas Negeri Yogyakarta	Diperiksa oleh :
----------------------------	---	------------------