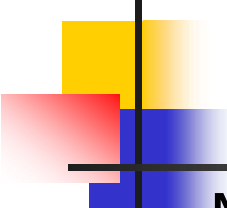


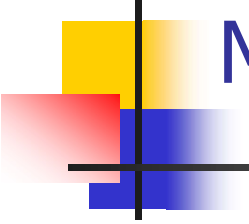
GUI

Oleh:
Mike Yuliana
PENS-ITS



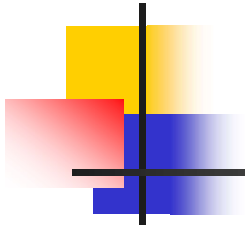
Topik

- Memahami komponen dasar swing
- Memahami implementasi swing dalam program
- Mengimplementasikan button, check box, dan radio button dalam program
- Mengimplementasikan label, Text Field dan Password Field dalam program
- Memahami tentang event/kejadian
- Memahami tentang cara menangani event/kejadian



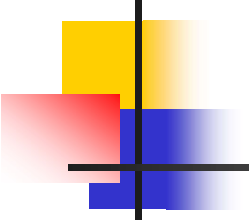
Membuat User Interface dengan Swing

- Program yang dibuat sebelumnya selalu berbasis console
- User lebih senang berinteraksi dengan dengan GUI



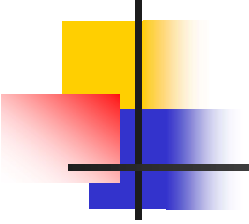
Pengenalan Swing

- Komponen swing yang akan digunakan sebagian besar terdapat dalam package javax.swing, javax.swing.event, java.awt dan java.awt.event.
- Komponen yang ada dalam javax.swing memiliki nama yang dimulai dengan J, contoh JButton, JList, dsb
- Sebagian besar komponen swing merupakan turunan dari komponen AWT



Komponen Dasar Swing

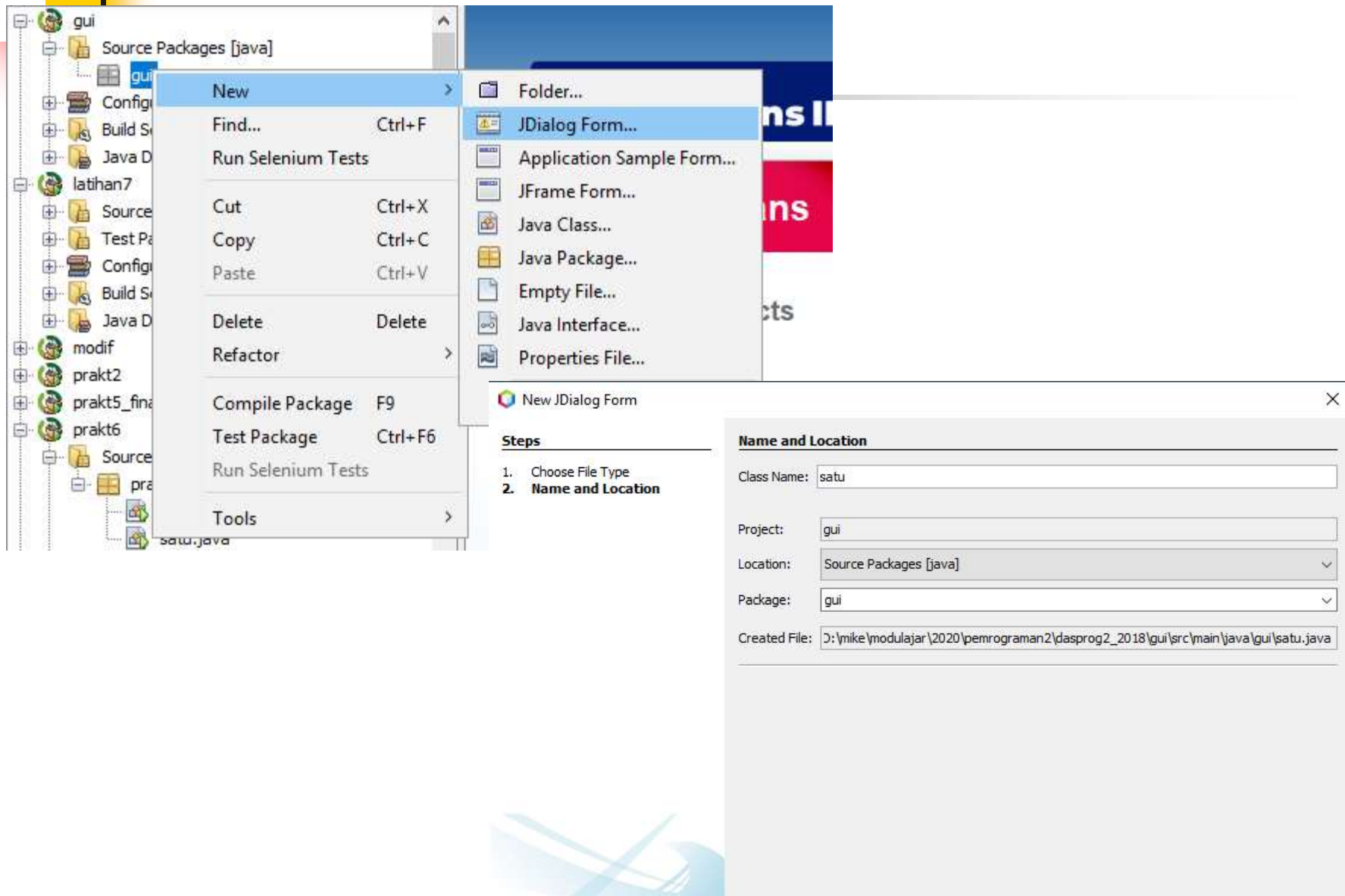
- Top-Level Container, merupakan container dasar dimana komponen lainnya akan diletakkan. Contoh JFrame, JDialog
- Intermediate Container, merupakan container perantara dimana komponen lainnya akan diletakkan, contoh JPanel
- Atomic component, merupakan komponen yang memiliki fungsi yang spesifik, dimana umumnya user langsung berinteraksi dengan komponen jenis ini. Contoh JButton, JLabel, JTextField, dan JTextArea



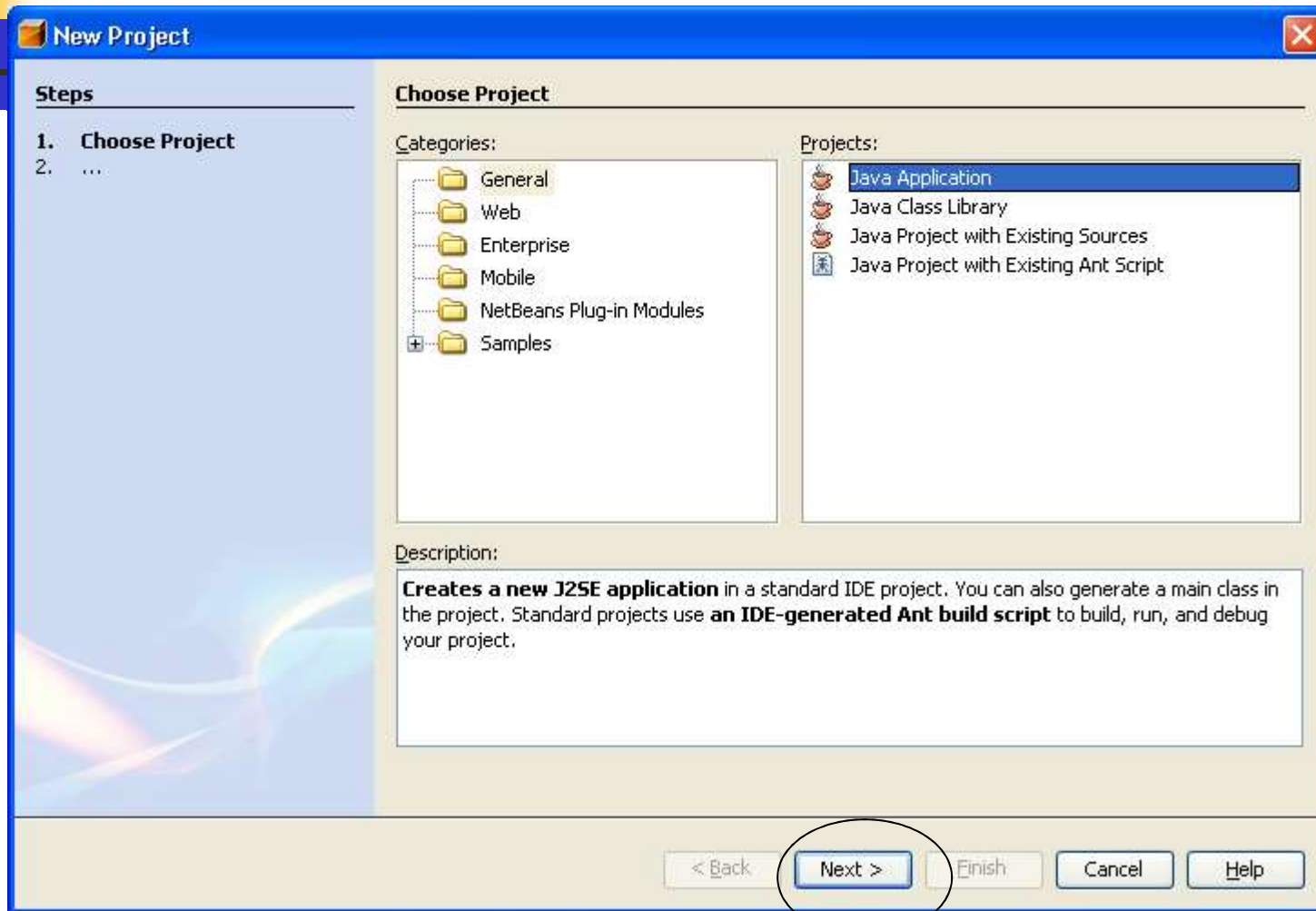
Komponen Dasar Swing

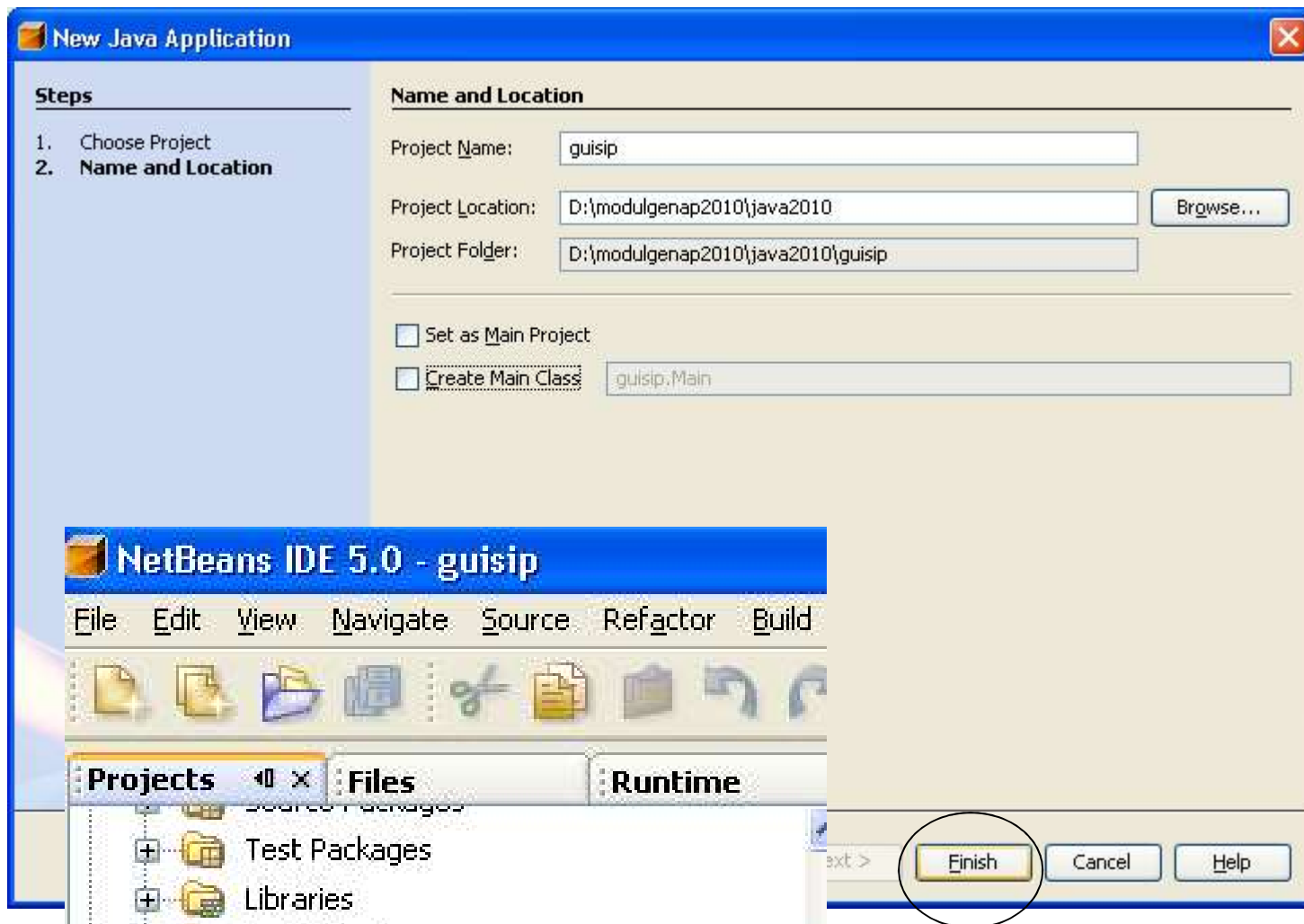
- Layout Manager, berfungsi untuk mengatur bagaimana tata letak atau posisi komponen yang akan diletakkan, satu sama lain di dalam suatu container. Contoh BorderLayout, BoxLayout, FlowLayout.
- Event Handling, untuk menangani event yang dilakukan oleh user seperti menekan tombol, memperbesar atau memperkecil ukuran frame, mengklik mouse dsb

Implementasi Swing dalam program → Netbeans 11

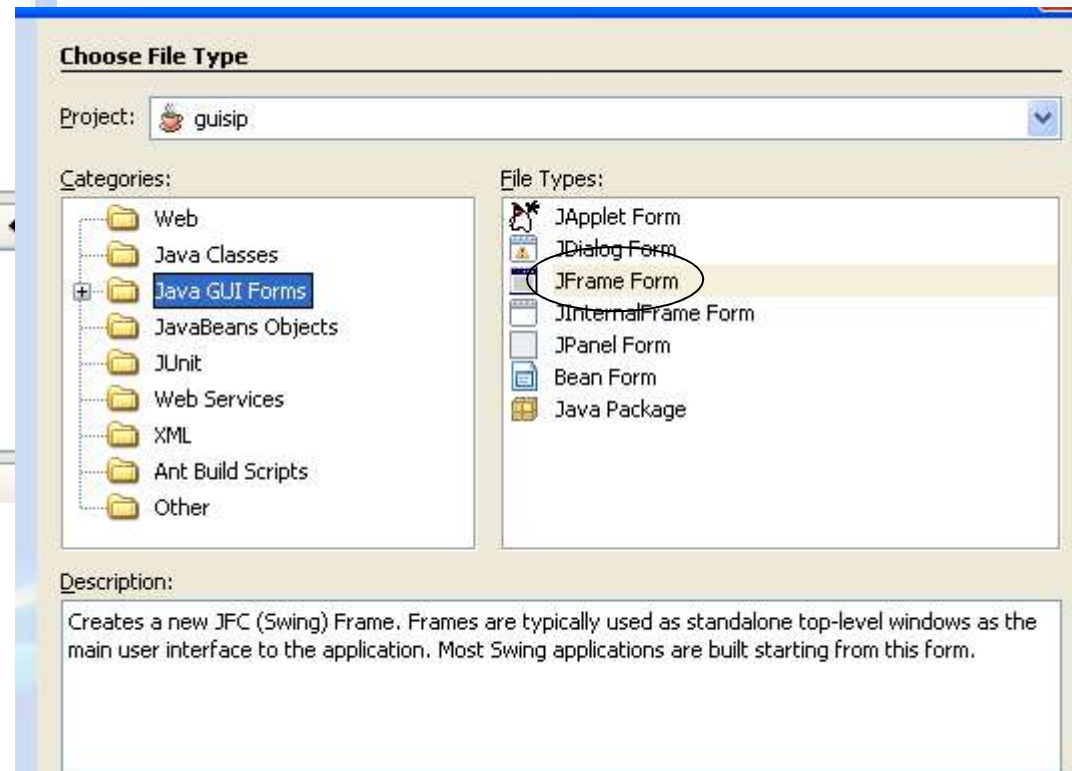
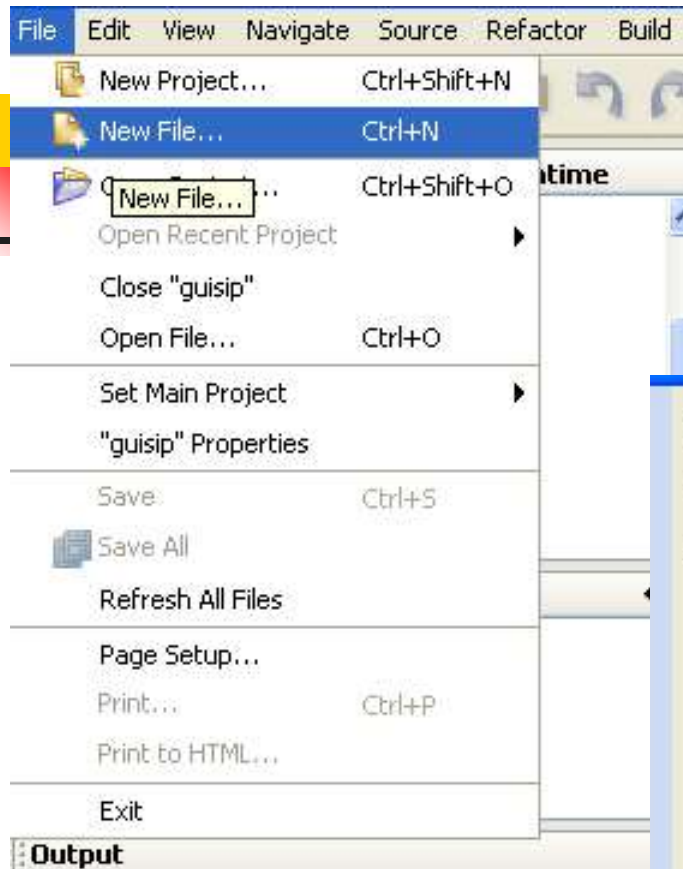


Implementasi Swing dalam program → Netbeans 5 keatas





Tampilan yang dihasilkan





New JFrame Form

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

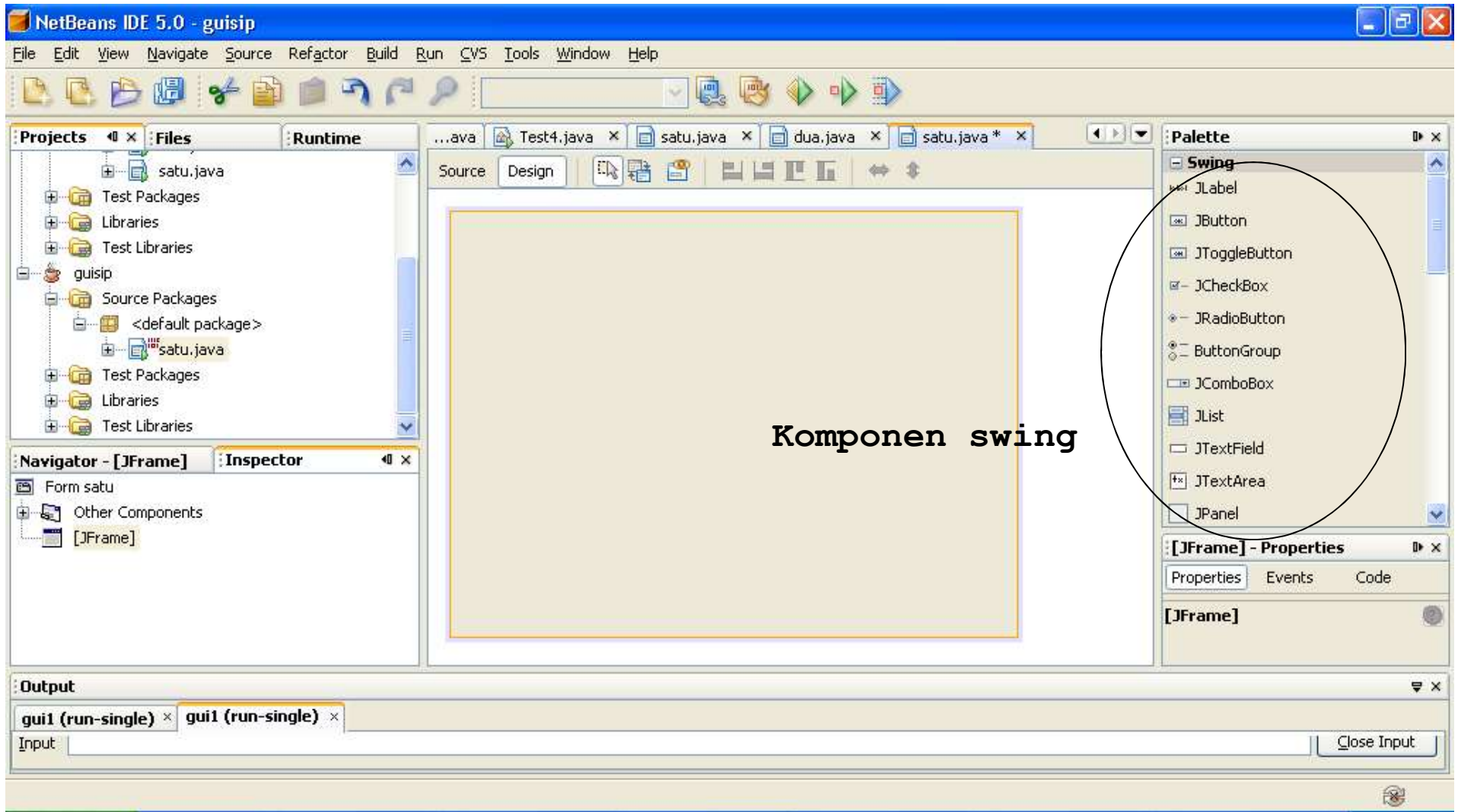
Package:

Created File:

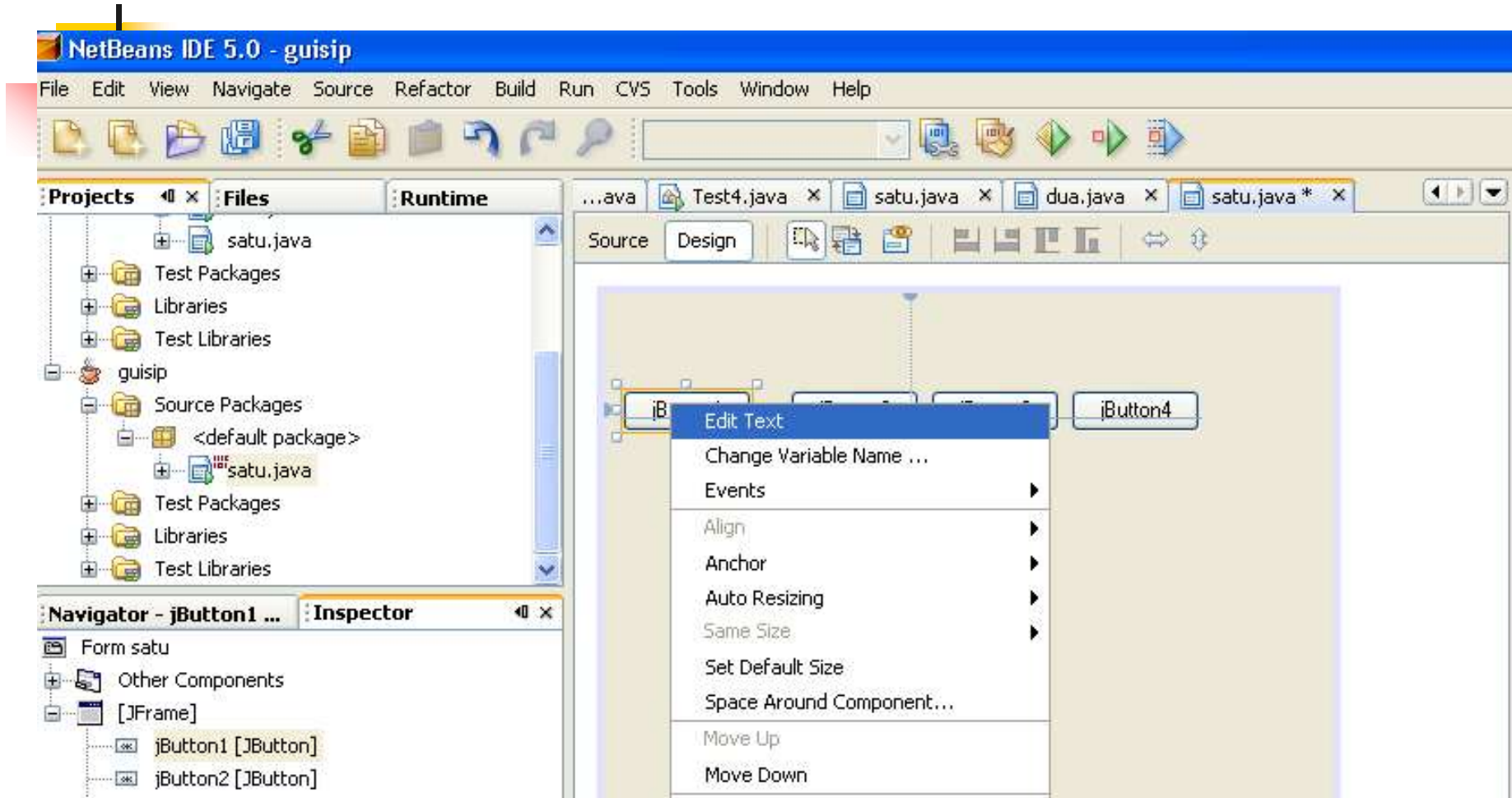
 Warning: As of JDK 1.4, it is highly recommended that you do NOT place Java classes in the defa...

< Back Next > **Finish** Cancel Help

Tampilan yang dihasilkan:



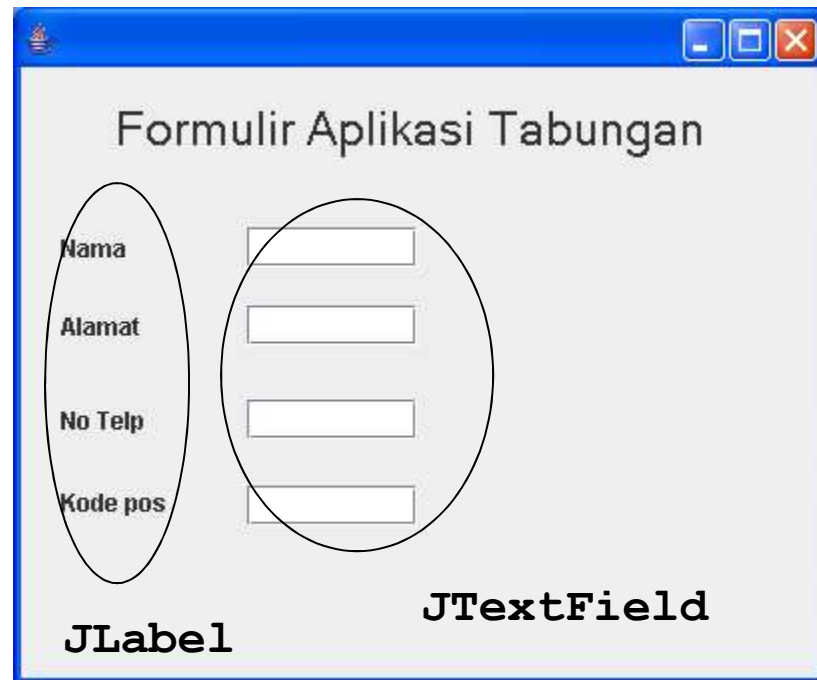
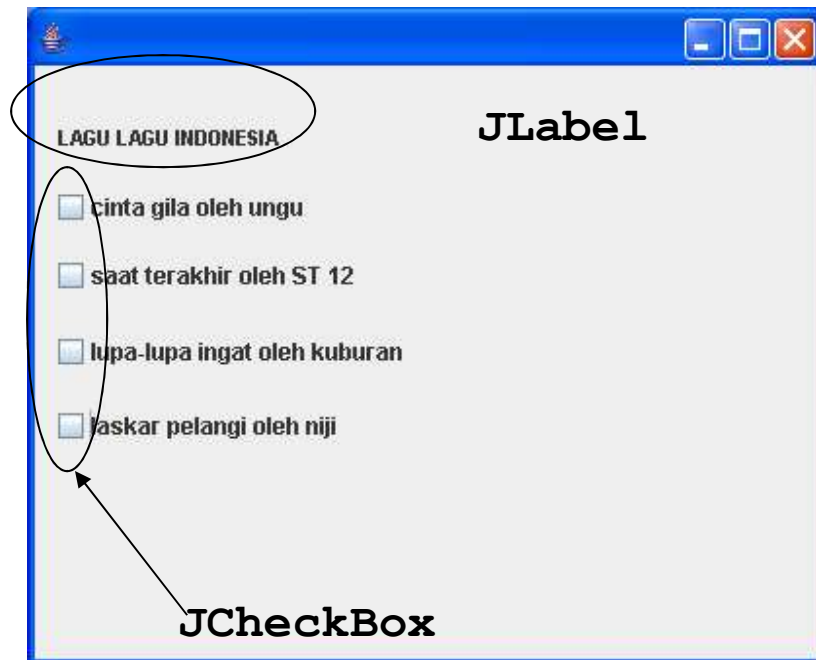
Contoh:

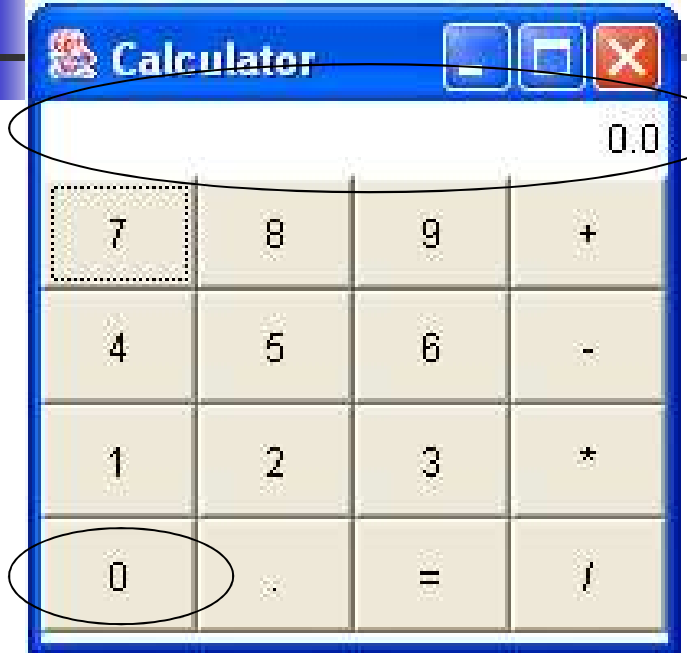
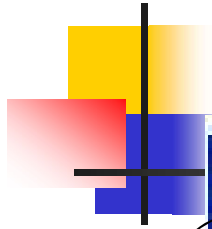


Edit Text digunakan untuk mengedit text dari button, label dll

Tugas!

Buatlah program dengan GUI Netbeans untuk menampilkan gambar-gambar dibawah ini!





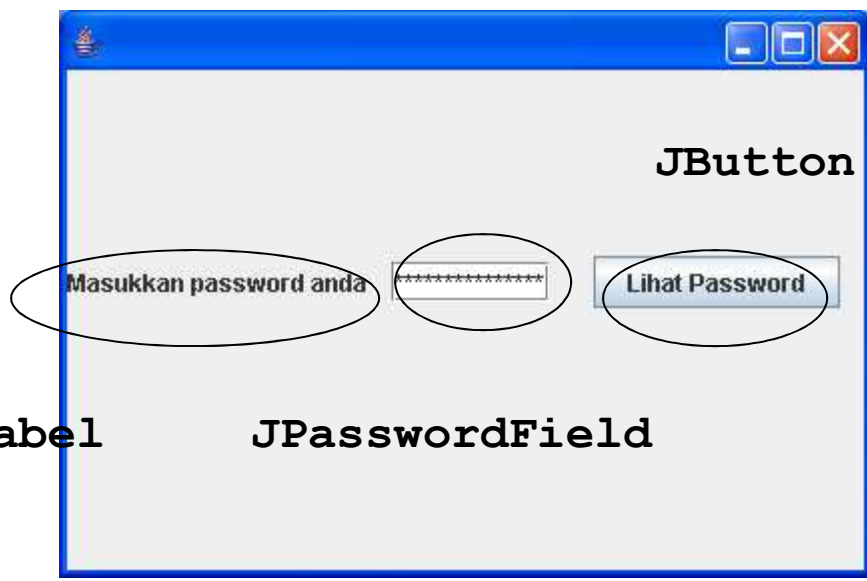
JButton

JTextField

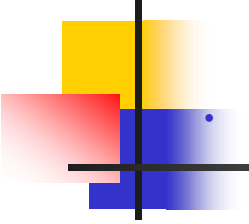
JLabel

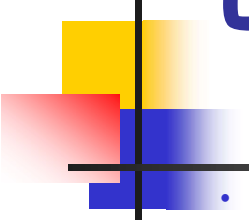
JPasswordField

JButton



What is an Event ?

- 
- Ketika user melakukan aksi terhadap sebuah user interface (misalnya meng-klik mouse atau menekan sebuah tombol), maka tindakan ini akan memunculkan sebuah **event**.
 - **Event** adalah OBJEK yang mendeskripsikan sebuah kejadian (peristiwa yang terjadi)
 - **Event Source** adalah pembangkit sebuah event, misalnya mouse click pada sebuah button akan membangkitkan sebuah `ActionEvent` dgn button sbg Event Source-nya.
 - **Event Handler** adalah sebuah method yang menerima sebuah objek event, menterjemahkan, dan kemudian memproses interaksi user.



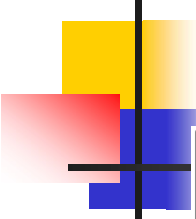
Event Listeners

- Swing menghandle event dengan sekumpulan interface yang disebut dengan **Event Listeners**
- Setiap kategori event, terdapat sebuah interface listener yang bersesuaian.
- Listener tsb harus diimplementasikan oleh class dari objek yang akan menerima event tersebut.
- Listener ini akan menetapkan method mana yang harus didefinisikan dalam sebuah class yang sesuai untuk menerima tipe event tersebut.
- Method-method ini akan dipanggil ketika event ybs terjadi.

Kategori Event, Interface & Methodnya (1)

Kategori	Nama Interface	Method
Action	ActionListener	actionPerformed(ActionEvent)
Item	ItemListener	itemStateChanged(ItemEvent)
Mouse	MouseListener	mousePressed(MouseEvent) mouseReleased (MouseEvent) mouseEntered (MouseEvent) mouseExited (MouseEvent) mouseClicked (MouseEvent)
Mouse Motion	MouseMotionListene r	mouseDragged (MouseEvent) mouseMoved (MouseEvent)
Key	KeyListener	keyPressed(KeyEvent) keyReleased (KeyEvent) keyTyped (KeyEvent)
Focus	FocusListener	focusGained(FocusEvent) focusLost (FocusEvent)

Kategori Event, Interface & Methodnya (2)



Kategori	Nama Interface	Method
Adjustment	AdjustmentListener	adjustmentValueChanged(AdjustmentEvent)
Component	ComponentListener	componentMoved(ComponentEvent) componentHidden(ComponentEvent) componentResized (ComponentEvent) componentShown (ComponentEvent)
Window	WindowListener	windowClosing(WindowEvent) windowOpened (WindowEvent) windowIconified (WindowEvent) windowDeiconified (WindowEvent) windowClosed (WindowEvent) windowActivated (WindowEvent) windowDeactivated (WindowEvent)
Container	ContainerListener	componentAdded(ContainerEvent) componentRemoved (ContainerEvent)
Text	TextListener	textValueChanged(TextEvent)

Example

Act that results in the event

User clicks a button, presses Return while typing in a text field, or chooses a menu item

User closes a frame (main window)

User presses a mouse button while the cursor is over a component

User moves the mouse over a component

Component becomes visible

Component gets the keyboard focus

Table or list selection changes

Listener type

ActionListener

WindowListener

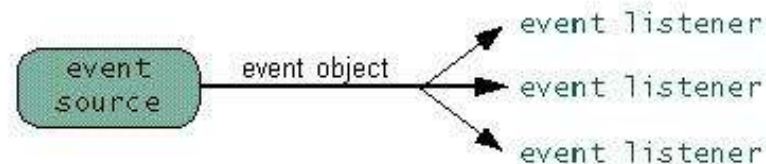
MouseListener

MouseMotionListener

ComponentListener

FocusListener

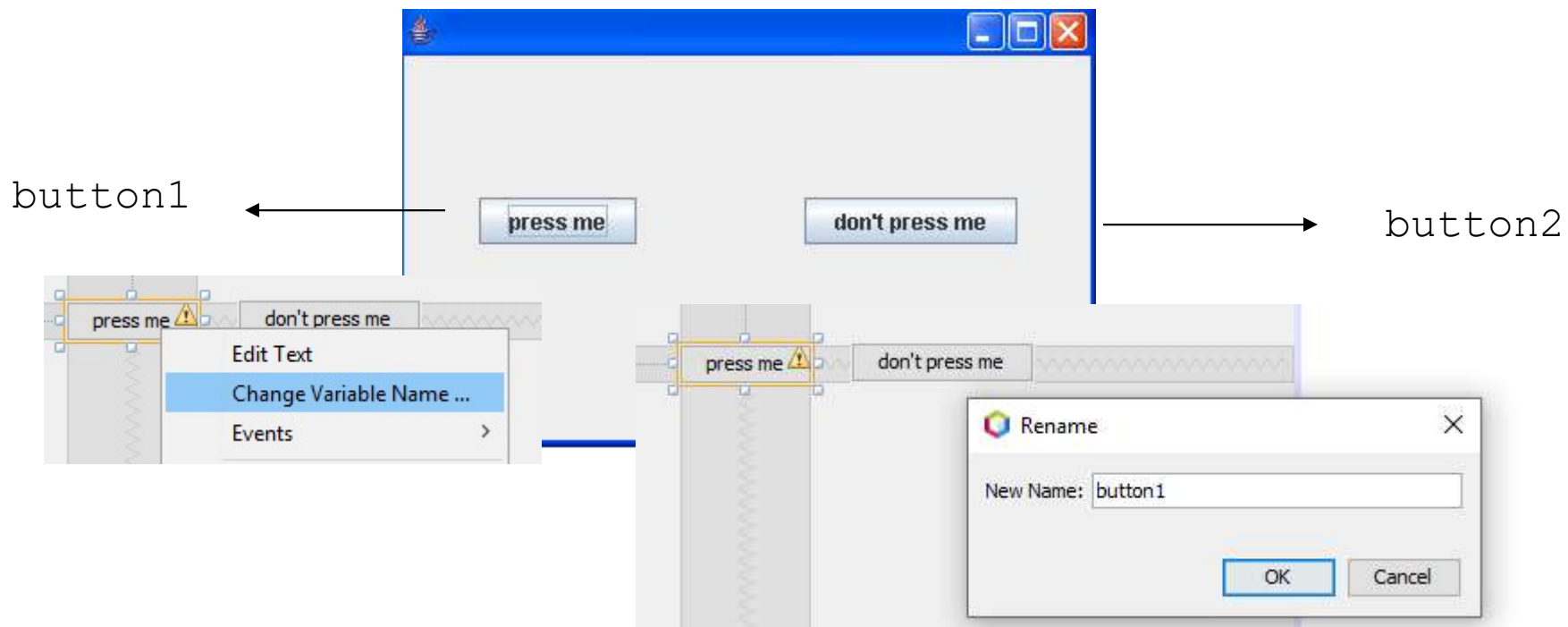
ListSelectionListener

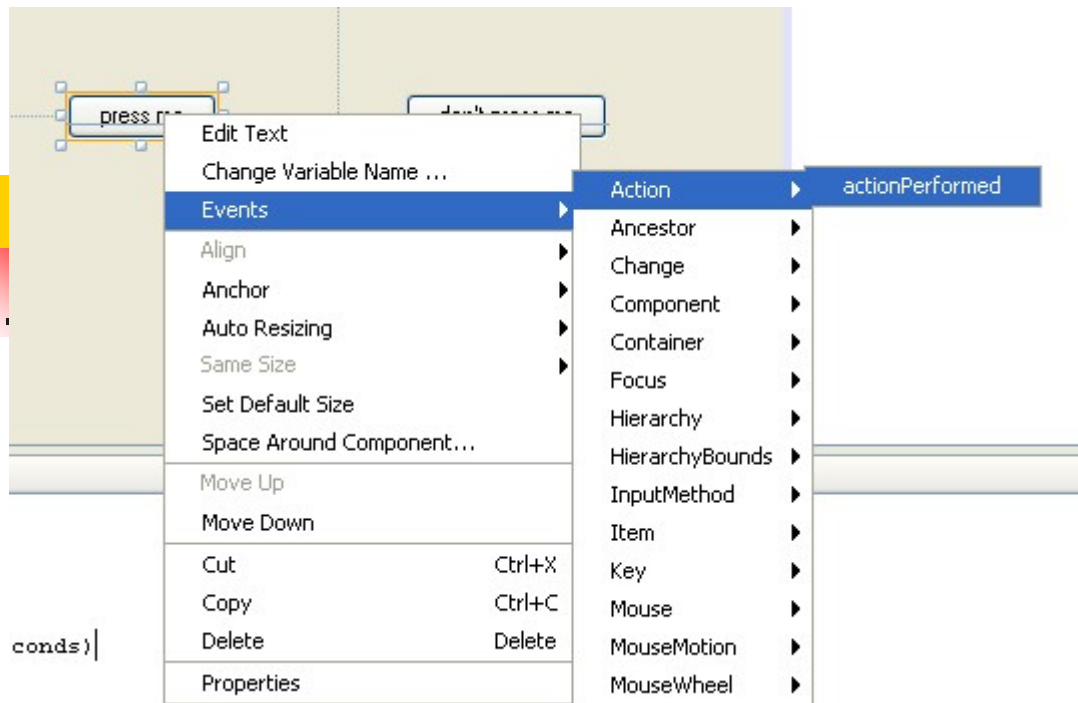


Caption: Multiple listeners can register to be notified of events of a particular type from a particular source.

Contoh 1:

- Program ini digunakan untuk mendeteksi tekanan pada button
- Untuk itu diimplementasikan satu buah interface yaitu ActionListener



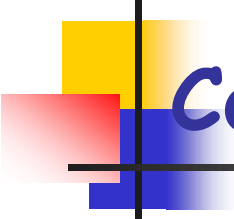


```
Source Design [Icons]
+ Generated Code
- private void button1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.out.println("sedang menekan press me");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new satu().setVisible(true);
        }
    });
}
```

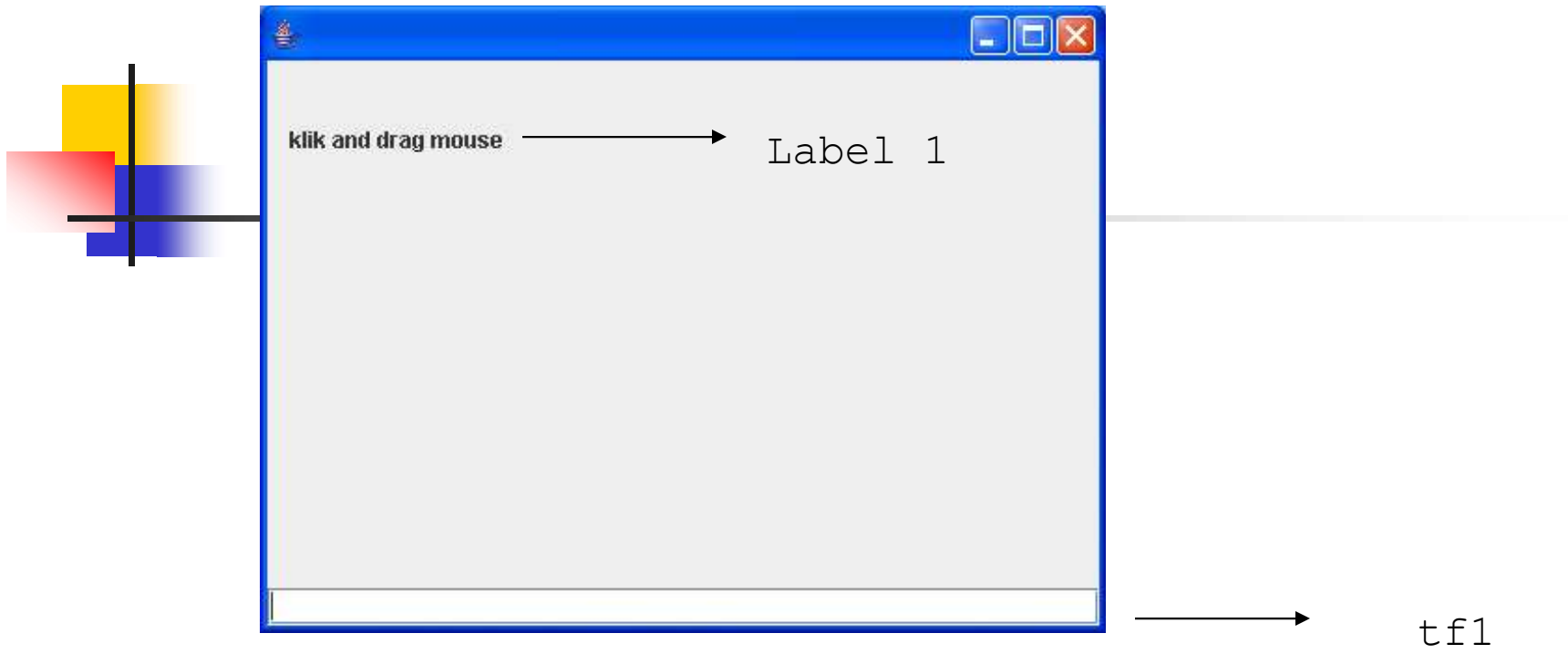
- Tambahkan program pada kasus button tersebut sehingga saat button1 ditekan didapat tampilan sebagai berikut:

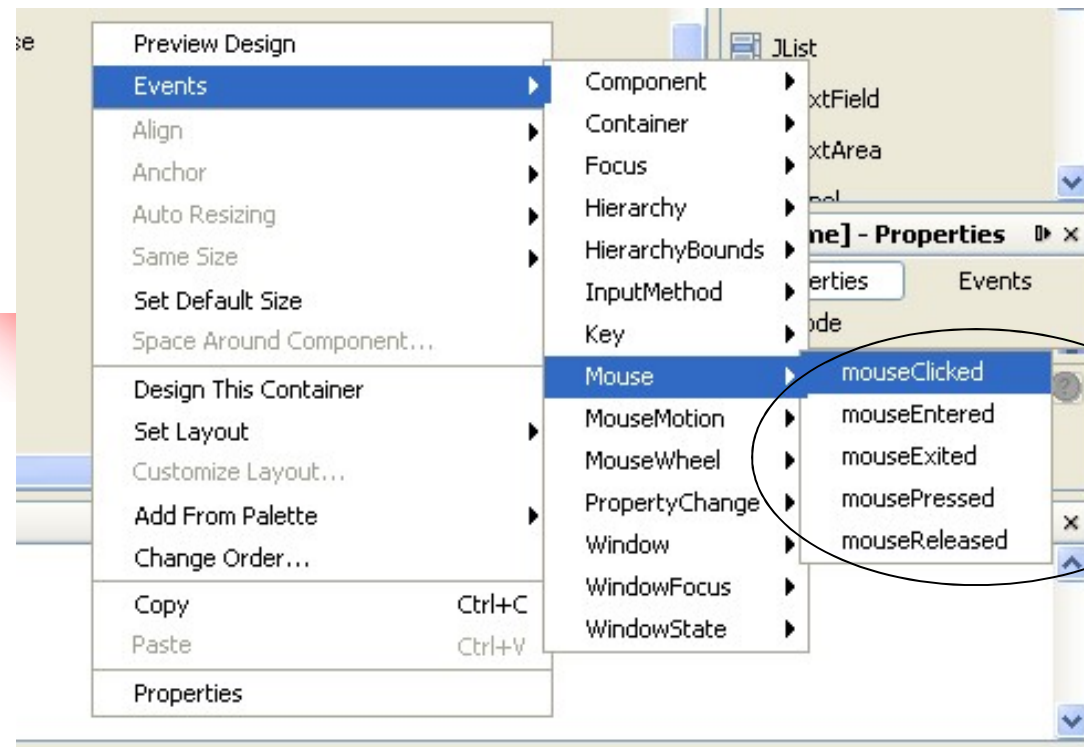




Contoh 2:

- Program ini digunakan untuk mendeteksi gerakan mouse ketika dalam keadaan ditekan (*mouse dragging*) serta gerakannya ketika dalam keadaan tidak ditekan (*mouse moving*).
- Untuk itu diimplementasikan dua buah interface, yaitu `MouseListener` dan `MouseMotionListener`



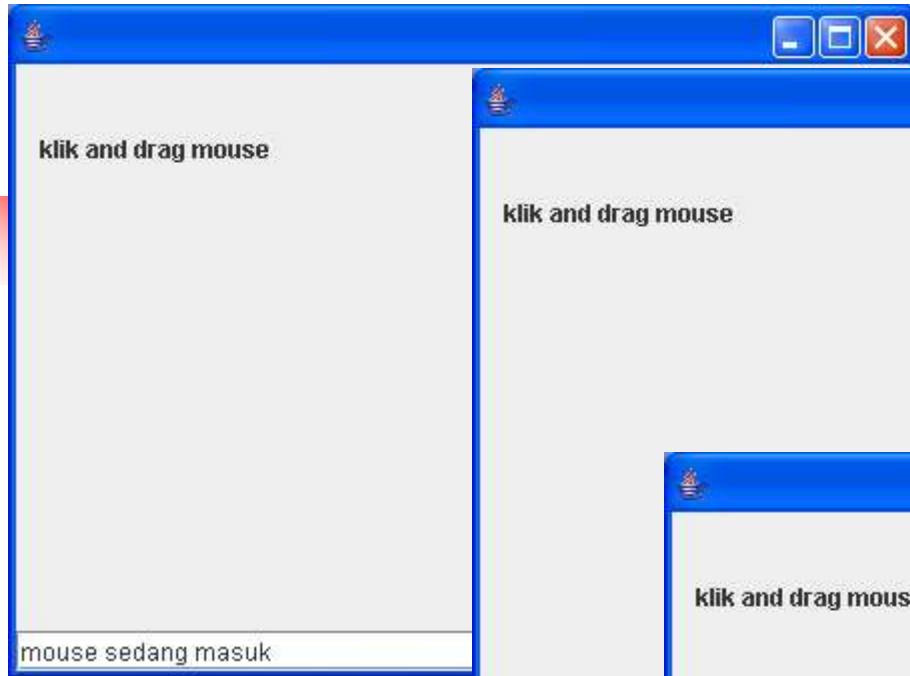


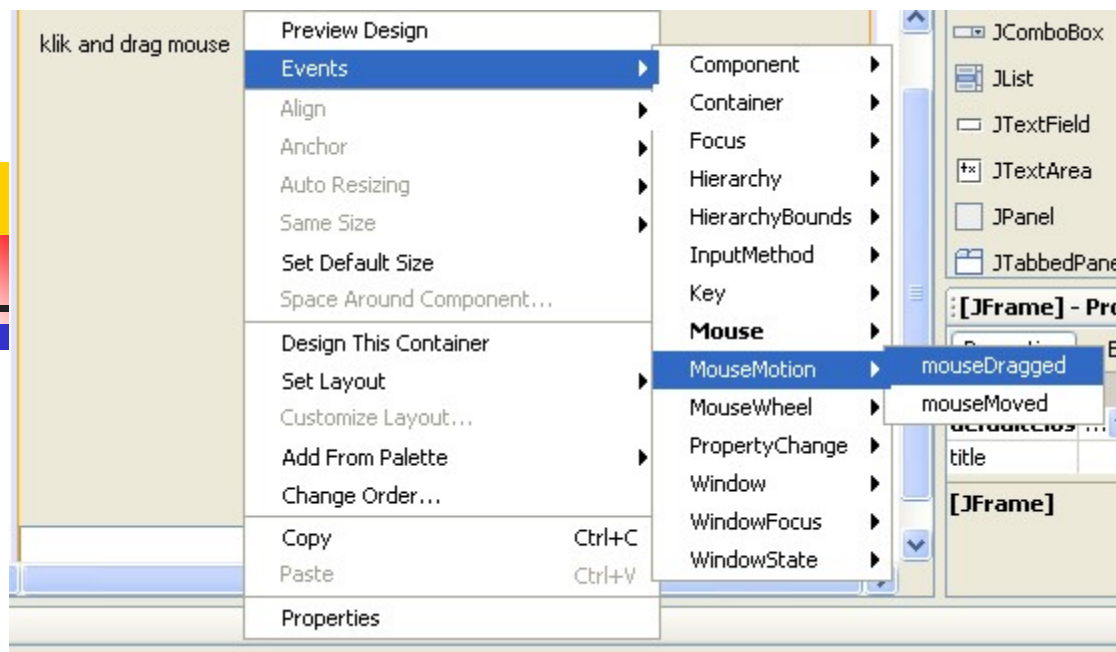
Untuk mendeteksi
Klik mouse, mouse
Masuk dll

```
private void formMouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
    tf1.setText("mouse sedang keluar");
}
```

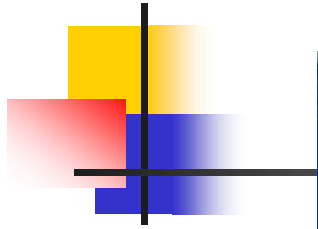
```
private void formMouseEntered(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
    tf1.setText("mouse sedang masuk");
}
```

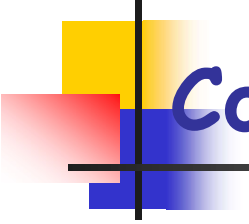
```
private void formMouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
    tf1.setText("mouse sedang diklik");
}
```





```
private void formMouseDragged(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
    tf1.setText("sedang berada pada koordinat sumbu x"+evt.getX()+
               "dan koordinat sumbu y"+evt.getY());
}
```





Contoh 3:

- Program ini digunakan untuk menghitung luas persegi panjang, dimana input yang dimasukkan berupa panjang dan lebar.
- Interface yang digunakan adalah Action Listener

Program untuk menghitung persegi pan...

panjang

lebar

hitung

actionPerformed

dua.java x tiga.java * x

panjang

hitung

Events

Align

Anchor

Auto Resizing

Same Size

Set Default Size

Space Around Component...

Action

Ancestor

Change

Component

Container

Focus

Hierarchy

HierarchyBounds

InputMethod

Item

Key

Mouse

MouseMotion

MouseWheel

PropertyChange

VetoableChange

Prope

Co

Prope

action

backgrc

compa

jButto

Tampilan yang dihasilkan:

Program untuk menghitung persegi pan...

panjang 4 tf1

lebar 3 tf2

hitung

tf3 12

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String p,l;  
    int panjang, lebar, luas;  
    p=tf1.getText();  
    l=tf2.getText();  
    panjang=Integer.parseInt(p);  
    lebar=Integer.parseInt(l);  
    luas=panjang*lebar;  
    tf3.setText(""+luas);  
}
```