

## PERTEMUAN 3: GRAPHICAL USER INTERFACE (GUI)

### A. TUJUAN PERKULIAHAN:

Pada pertemuan ini akan dijelaskan mengenai pembuatan desain interface menggunakan class objek (komponen) dan pengolahannya. Setelah mempelajari materi perkuliahan ini, mahasiswa mampu:

- 3.1 Menggunakan class objek untuk desain interface
- 3.2 Menggunakan method dan event objek pada desain interface
- 3.3 Menuliskan source code sesuai event yang digunakan

### B. URAIAN MATERI

<b>Tujuan Pembelajaran 3.1:</b>
Penggunaan class Objek untuk desain interface

- **JLabel, JTextField, JButton**

JLabel merupakan komponen yang digunakan untuk menambah keterangan/teks pendek. JLabel dapat juga digunakan untuk menampilkan image.

JTextField digunakan untuk memasukkan data (input) dan menampilkan hasil (output data) dalam bentuk satu baris.

JButton digunakan merupakan tombol perintah untuk melakukan suatu proses

Masing-masing class objek mempunyai beberapa metode

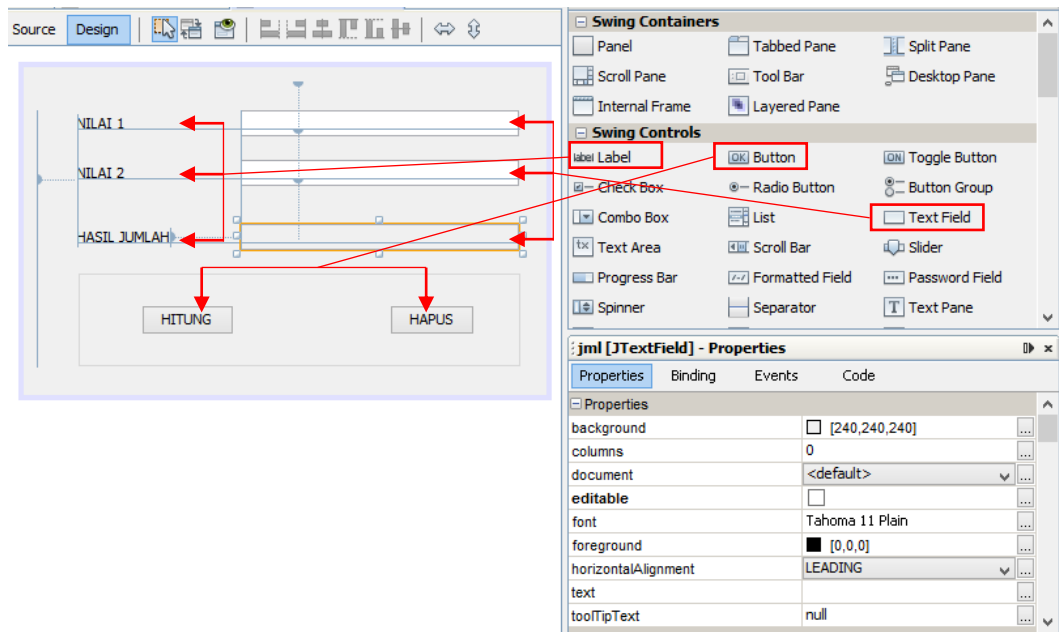
#### **Metode yang akan sering digunakan pada JTextField**

- **getText()** untuk membaca data/nilai pada JTextField
- **setText()** untuk menampilkan/memberi nilai pada JTextField

Bentuk umum penulisan (Sintaks):

<i>namaVarobj.metode</i>
--------------------------

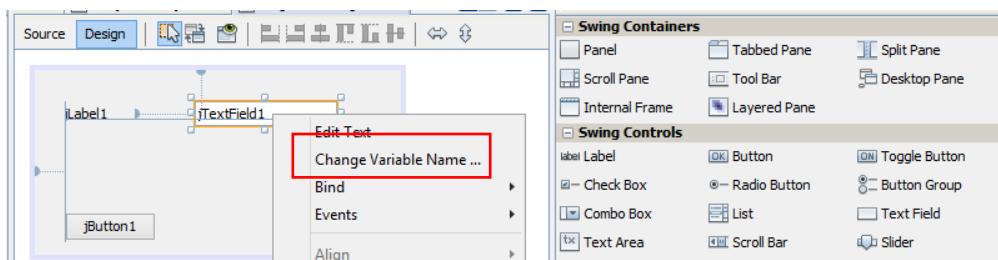
contoh: buat desain di bawah ini



Gb. 3.1 Pallette Desain GUI

langkah-langkah:

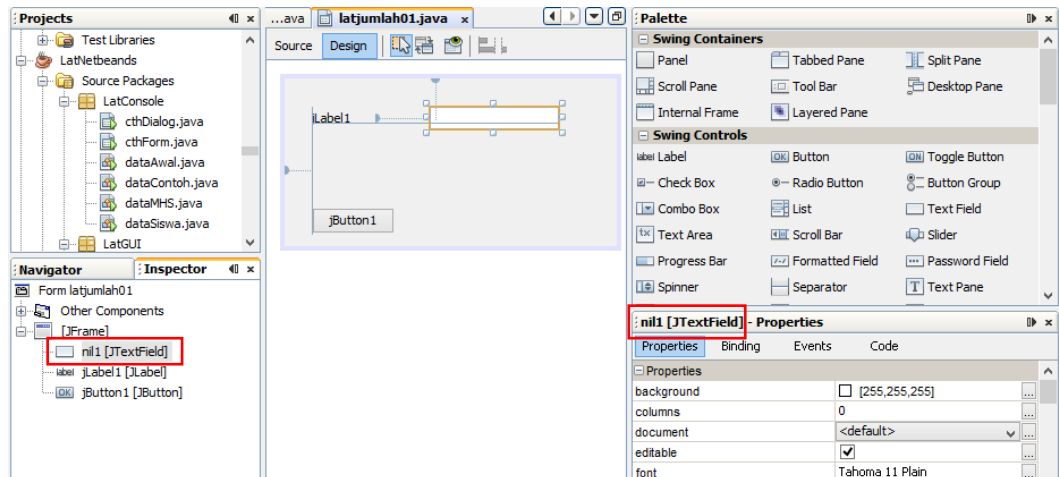
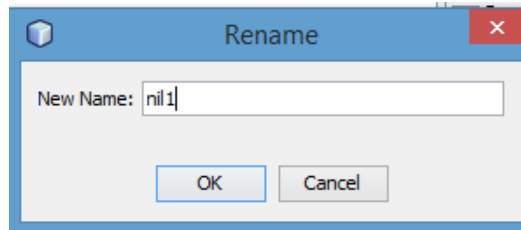
- buat class baru dengan klik kanan pada package > **New > JFrame Form**



**Edit Text** untuk mengatur tampilan nilai awal (jika tidak ada, kosongkan/hapus)



**Change Variable Name** untuk mengatur nama variable objek, nama variable ini akan digunakan dalam penulisan program.



*Catatan:*

*Semua data yang di-input/output pada objek bertipe String*

Ada beberapa metode yang digunakan untuk konversi data

a. Konversi Tipe Data String menjadi **Data Angka**

`Integer.parseInt(data string);` → mengubah dari tipe data string ke integer

`Float.parseFloat(data string);` → mengubah dari tipe data string ke float

`Double.parseDouble(data string);` → mengubah dari tipe data string ke double

Atau

`Integer.valueOf (data_String).intValue()`

`Double.valueOf (data_String).doubleValue()`

`Float.valueOf (data_String).floatValue()`

b. Konversi Tipe Data Angka menjadi **Data String**

Integer.toString(data angka integer); → mengubah dari integer ke String

Float.toString(data angka float); → mengubah dari float ke String

Double.toString(data angka double); → mengubah dari double ke String

Cara untuk menampilkan data ke JTextField

**Sintaks:**

namaVarObj.setText(data string);

I. namaVarObj.setText(Float.toString(namaVarAngka));

II. namaVarObj.setText(" " + namaVarAngka);

III. namaVarObj.setText("nilai");

• **Radio Button, Checkbox**

Metode yang digunakan pada Radio Button dan Checkbox adalah **isSelected()**

Perbedaan Radio Button dengan Checkbox, pada radio button hanya dapat dipilih satu pada satu kelompok pilihan sedangkan Checkbox bisa dipilih lebih dari satu.

**Sintaks:**

namaVarObj.metode

contoh:

```
if(namaVarObj.isSelected())
{
    ..... ;
    ..... ;
}
Else //Checkbox
{
    ..... ;
    ..... ;
}
```

Metode yang digunakan pada TextArea sama dengan TextField yaitu **setText()**  
Sintaks:

```
namaVarObj.metode
```

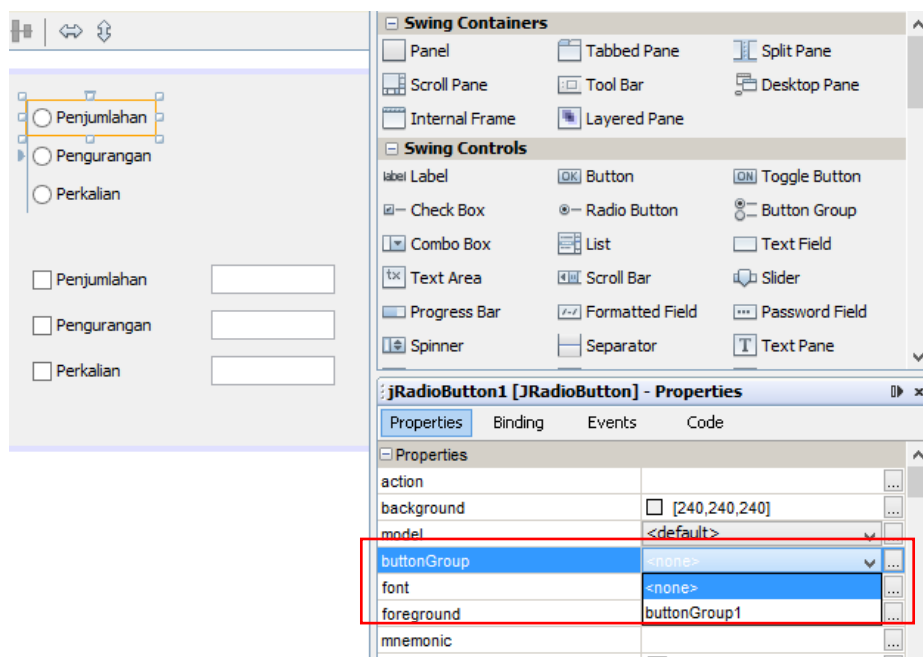
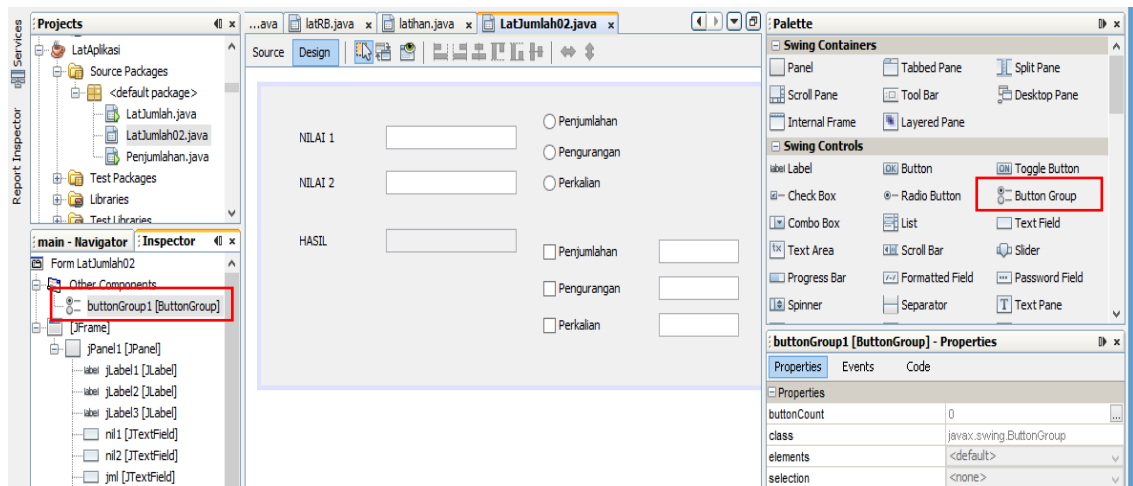
contoh desain interface:

The diagram shows a layout for a simple calculator application. On the left, there are three text input fields labeled 'NILAI 1', 'NILAI 2', and 'HASIL'. To the right of 'NILAI 1' and 'NILAI 2' are three radio buttons for selecting an operation: 'Penjumlahan' (Addition), 'Pengurangan' (Subtraction), and 'Perkalian' (Multiplication). To the right of the 'HASIL' field, there are three checkboxes for the same operations, each followed by a text field to display the result.

Gb. 3.2 Contoh Desain GUI

The screenshot shows the implemented GUI in a window titled 'FORM LATIHAN PENJUMLAHAN'. The layout matches the wireframe. In this state, the radio buttons for 'Penjumlahan', 'Pengurangan', and 'Perkalian' are all selected, which is incorrect for a single-choice operation. A red box highlights these radio buttons, and a callout box with an arrow pointing to them contains the text 'Harusnya hanya 1 pilihan' (Should be only 1 choice). Another red box highlights the checkboxes for the same operations, which are all checked.

Gunakan objek Button Group, objek tidak tampil di window tapi akan terlihat di window inspector. Kemudian atur properties **buttonGroup** untuk masing-masing objek radio button (pilih nama buttongroup yang sama jika dalam satu kelompok)



Gb. 3.3 Jendela Properties

FORM LATIHAN PENJUMLAHAN

NILAI 1

NILAI 2

HASIL

☐ Penjumlahan

☒ Pengurangan

☐ Perkalian

☒ Penjumlahan

☒ Pengurangan

☒ Perkalian

hanya terpilih 1 pilihan saja

- **TextArea**

Metode yang digunakan pada objek TextArea adalah **setText()**

Sintaks

```
namaVarObj.setText(nimField.getText() + namaField.getText()+.....);
```

contoh desain:

Nim

Nama

- **Combobox**

Digunakan untuk membuat pilihan data

Beberapa metode yang digunakan, a.l:

- **getSelectedItem()**

digunakan untuk membaca/mengambil data yang hasilnya berupa teks pilihan

Sintaks:

```
namaVarObj. getSelectedItem().toString()
```

- **getSelectedIndex()**

digunakan untuk membaca/mengambil data yang hasilnya berupa nomor urut pilihan data (no. urut dimulai dari 0)

Sintaks:

```
namaVarObj. getSelectedIndex()
```

**Penggunaan kondisi untuk data pada Combobox**

```
If(namaVarObj. getSelectedItem().toString().equals("data pembanding"))
```

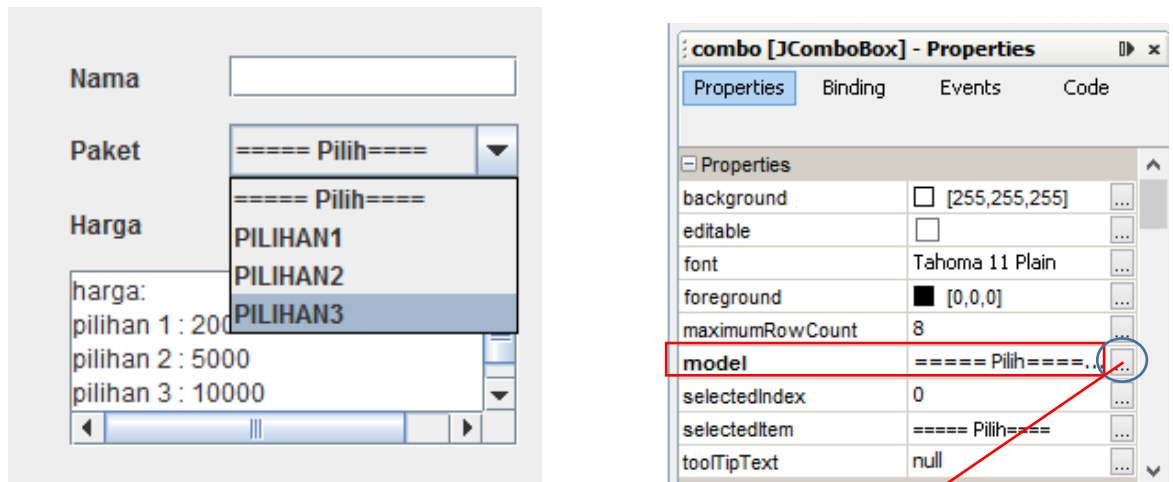
```
If(namaVarObj. getSelectedIndex() == no.urut)
```

Contoh desain:

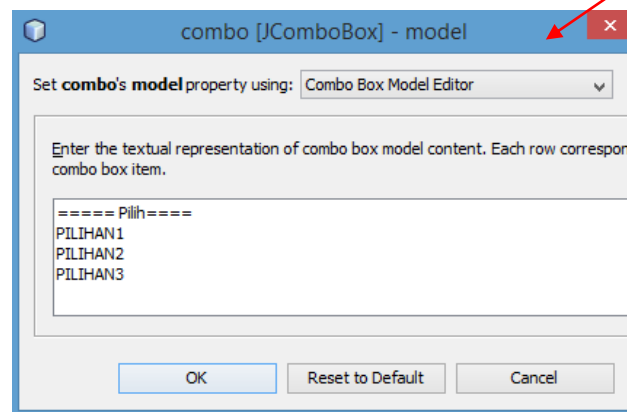
The image shows a graphical user interface (GUI) for a form. It contains three input fields: 'Nama' (a text box), 'Paket' (a combobox with the text '==== Pilih ===='), and 'Harga' (a text box). Below these fields is a text area containing the text: 'harga: pilihan 1 : 2000', 'pilihan 2 : 5000', and 'pilihan 3 : 10000'.



Cara menambahkan item/pilihan pada combobox



Membuat pilihan Combobox melalui jendela properti:



Gb. 3.4 Item Combobox

Membuat pilihan Combobox melalui kode program:

Metode yang digunakan: **addItem()**, **removeAllItems()**

**addItem()** untuk menambah pilihan pada combobox

**removeAllItems()** menghapus seluruh pilihan pada combobox

```
private void formWindowActivated(java.awt.event.WindowEvent evt) {
    // TODO add your handling code here:
    combo.removeAllItems();
    combo.addItem("==== Pilih ====");
    for (int i=1; i<=3; i++){
        combo.addItem("Pilihan" + i);
    }
}
```

### Tujuan Pembelajaran 3.2:

Penggunaan method dan event objek pada desain interface

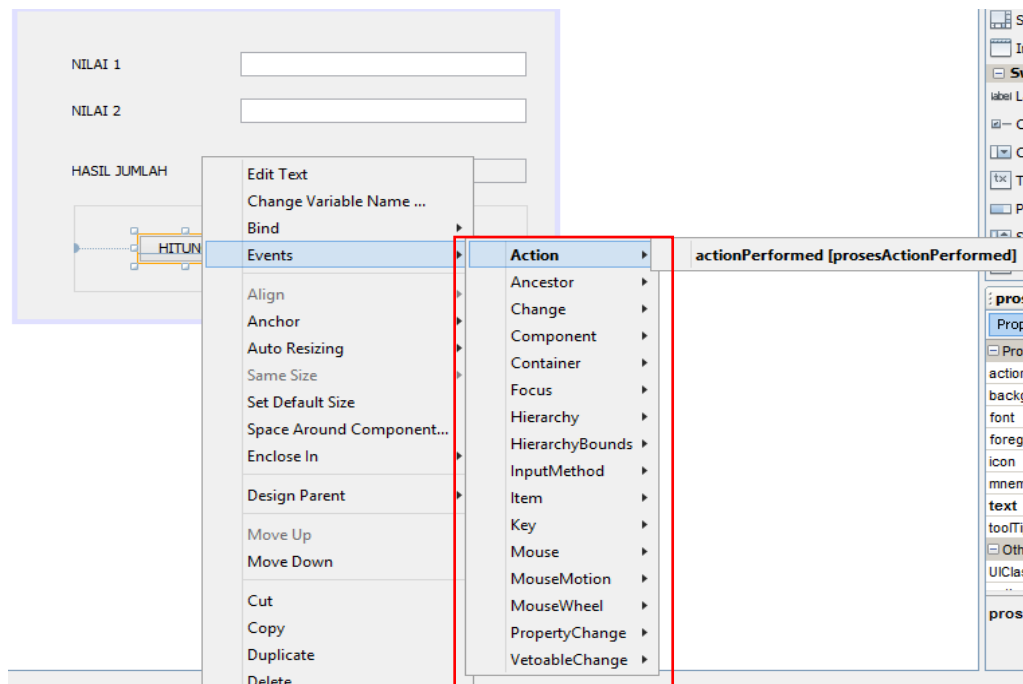
#### Event

Merupakan respon dari program ketika user melakukan tindakan terhadap GUI tertentu dalam aplikasi.

Beberapa Event, yaitu:

1. Event objek merupakan objek yang mendeskripsikan/menggambarkan sebuah event yang dipicu/dibangkitkan (di *-trigger*) oleh event source
2. Event Handler merupakan method yang menerima event objek dan melakukan respon sesuai dengan event objek tersebut
3. Event Source merupakan pembangkit/*trigger* sebuah event objek yang ditangkap oleh Event Listener. Event source dibedakan dari penamaan komponen objek class (nama Variabel Objek)
4. Event Listener merupakan interface yang akan meng-handle event yang terjadi. Event Listener digunakan untuk menangkap event yang terjadi pada komponen/Event Source. Listener harus diimplementasikan oleh class yang akan meng-handle event.

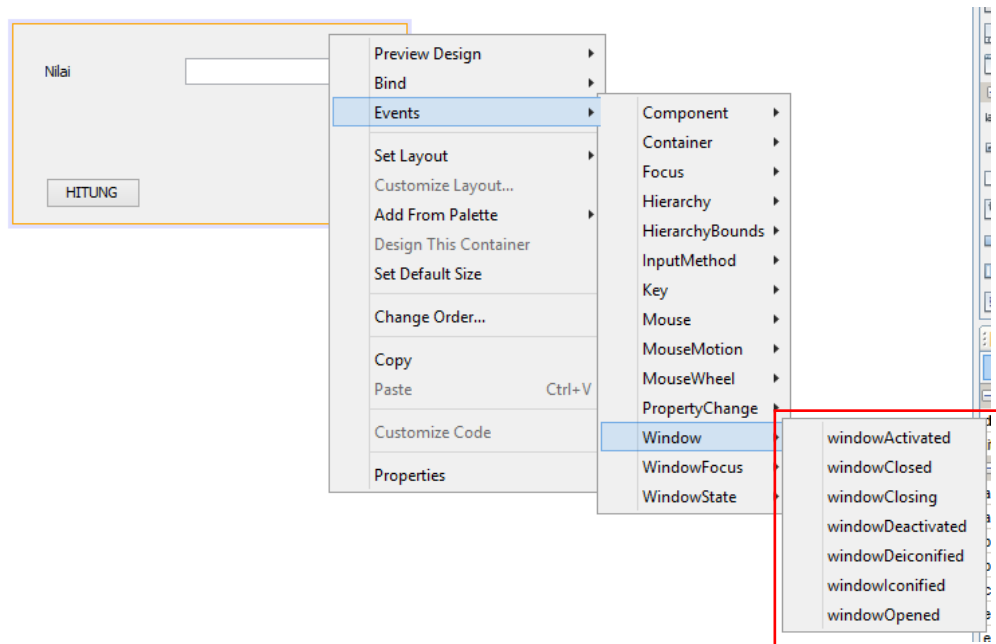
### Beberapa metode pada Listener



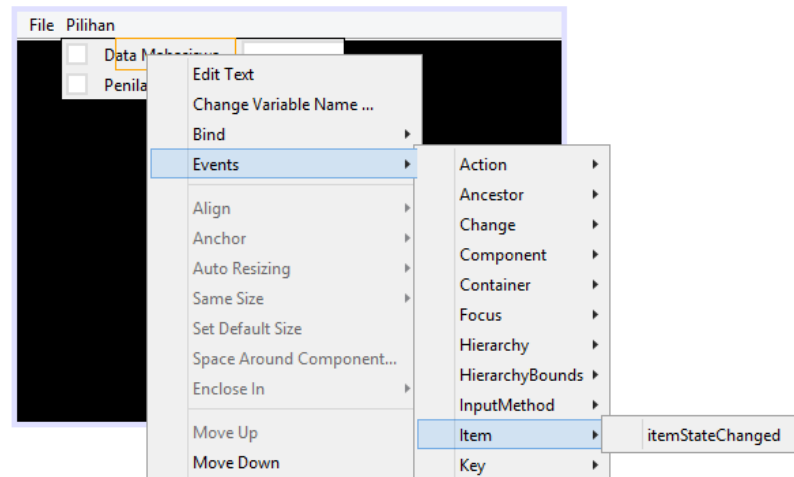
Gb. 3.5 Metode Listener

- ✓ `actionPerformed (Action Event)`  
merupakan event/kejadian ketika objek diklik
- ✓ `adjustmentValueChanged(AdjustmentEvent)`  
merupakan event/kejadian ketika nilai berubah
- ✓ `componenHidden(ComponentEvent)`  
merupakan event/kejadian ketika komponen/objek disembunyikan
- ✓ `componenShown(ComponentEvent)`  
merupakan event/kejadian ketika komponen/objek ditampilkan
- ✓ `componenMoved(ComponentEvent)`  
merupakan event/kejadian ketika komponen/objek berpindah tempat
- ✓ `componenResized(ComponentEvent)`  
merupakan event/kejadian ketika ukuran komponen/objek berubah
- ✓ `componenAdded(ContainerEvent)`  
merupakan event/kejadian ketika komponen/objek ditambahkan
- ✓ `componenRemoved(ContainerEvent)`  
merupakan event/kejadian ketika komponen/objek dihapus

- ✓ `focusGained(FocusEvent)`  
merupakan event/kejadian ketika komponen/objek menjadi focus
- ✓ `focusLost(FocusEvent)`  
merupakan event/kejadian ketika focus berpindah dari komponen/objek
- ✓ `keyPressed(KeyEvent)`  
merupakan event/kejadian ketika menekan tombol
- ✓ `keyReleased(KeyEvent)`  
merupakan event/kejadian ketika penekanan terhadap tombol dilepas
- ✓ `keyTyped(KeyEvent)`  
merupakan event/kejadian ketika dilakukan pengetikan
- ✓ `mouseClicked(MouseEvent)`  
merupakan event/kejadian ketika mouse kiri diklik
- ✓ `mouseEntered(MouseEvent)`  
merupakan event/kejadian ketika penunjuk mouse pertama kali menyorot ke komponen/objek
- ✓ `mouseExited(MouseEvent)`  
merupakan event/kejadian ketika penunjuk mouse meninggalkan komponen/objek
- ✓ `mousePressed(MouseEvent)`  
merupakan event/kejadian ketika mouse kiri ditekan
- ✓ `mouseReleased(MouseEvent)`  
merupakan event/kejadian ketika penekanan mouse berakhir/dilepas
- ✓ `mouseDragged(MouseEvent)`  
merupakan event/kejadian ketika mouse diklik, ditekan sambil ditarik/diseret
- ✓ `mouseMoved(MouseEvent)`  
merupakan event/kejadian ketika penunjuk mouse berpindah posisi



- ✓ `windowOpened(WindowEvent)`  
merupakan event/kejadian ketika jendela/window dibuka
- ✓ `windowClosing(WindowEvent)`  
merupakan event/kejadian ketika jendela/window sedang ditutup
- ✓ `windowClosed(WindowEvent)`  
merupakan event/kejadian ketika jendela/window ditutup
- ✓ `windowActivated(WindowEvent)`  
merupakan event/kejadian ketika jendela/window diaktifkan
- ✓ `windowDeactivated(WindowEvent)`  
merupakan event/kejadian ketika jendela/window tidak aktif
- ✓ `windowIconified(WindowEvent)`  
merupakan event/kejadian ketika jendela/window dalam kondisi ukuran minimal
- ✓ `windowDeiconified(WindowEvent)`  
merupakan event/kejadian ketika jendela/window berubah dari kondisi ukuran minimal



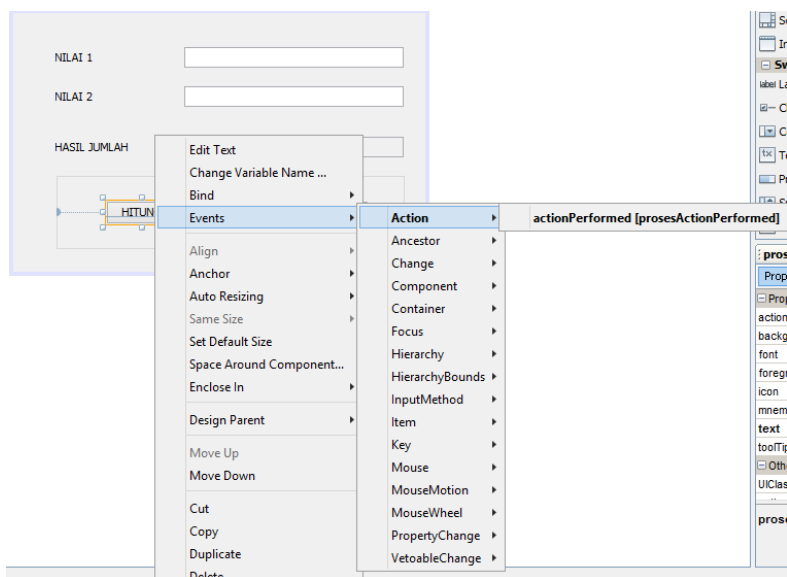
- ✓ `itemStateChanged(ItemEvent)`  
merupakan event/kejadian ketika status item/menu berubah/dipilih

### Tujuan Pembelajaran 3.3:

Penulisan source code sesuai event yang digunakan

program ditulis pada objek class yang yang diberi tindakan/event.

Contoh event:



Setelah membuat desain interface dan menentukan event, maka saatnya untuk penulisan program.

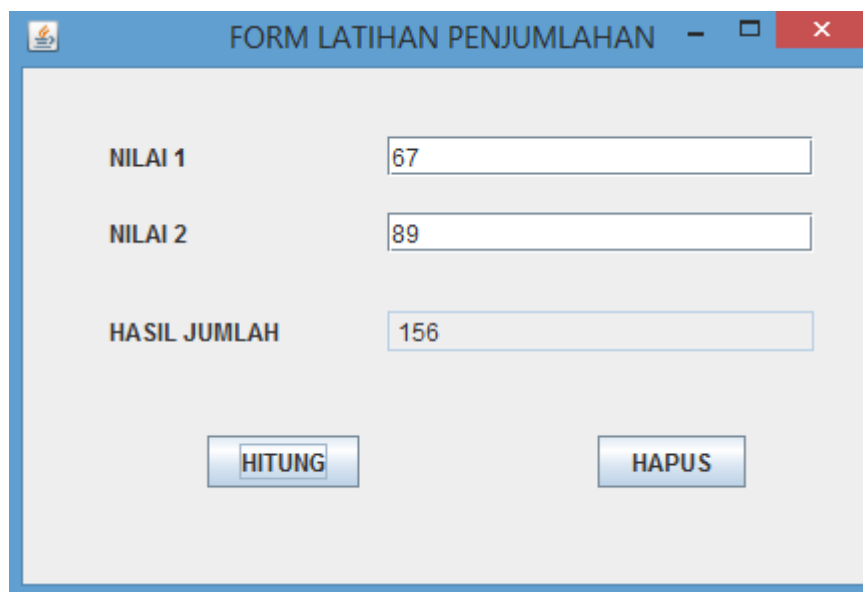
#### Contoh 1:

source untuk desain penjumlahan

Hasil penjumlahan akan tampil saat **tombol Hitung** diklik.

```
private void prosesActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    int n1,n2,hs;  
    n1=Integer.parseInt(nil1.getText());  
    n2=Integer.parseInt(nil2.getText());  
    hs=n1+n2;  
    jml.setText(" " + hs);  
}
```

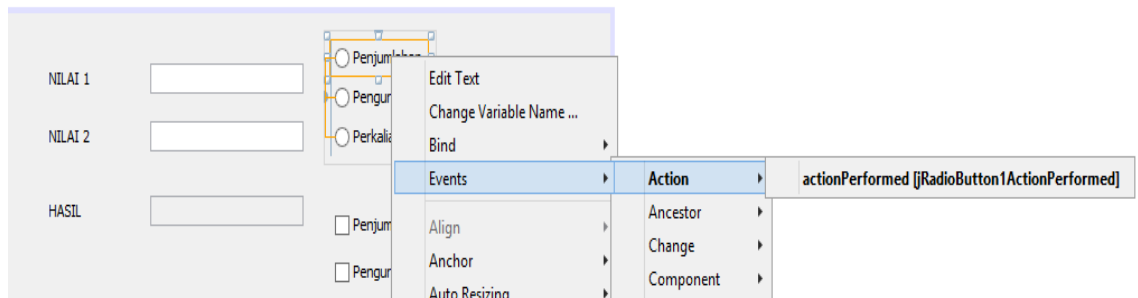
Hasil:



The screenshot shows a Java Swing window titled "FORM LATIHAN PENJUMLAHAN". Inside the window, there are three text input fields. The first field is labeled "NILAI 1" and contains the number "67". The second field is labeled "NILAI 2" and contains the number "89". The third field is labeled "HASIL JUMLAH" and contains the number "156". Below these fields, there are two buttons: "HITUNG" and "HAPUS". The "HITUNG" button is on the left and the "HAPUS" button is on the right. The window has a standard Windows-style title bar with a blue background and a red close button.

#### Contoh 2:

source untuk desain penjumlahan, pengurangan, dan perkalian menggunakan pilihan pada Radio Button (pilih salah satu). Saat radio button dipilih hasil akan tampil sesuai pilihan operasi matematika



```
private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int n1,n2,hs;
    n1=Integer.parseInt(nil1.getText());
    n2=Integer.parseInt(nil2.getText());
    hs=n1+n2;
    jml.setText(" " + hs);
}
```

Hasil:

### Contoh 3:

Source untuk desain penjumlahan, pengurangan, dan perkalian menggunakan pilihan pada checkbox (bisa pilih lebih dari 1), hasil akan tampil sesuai pilihan



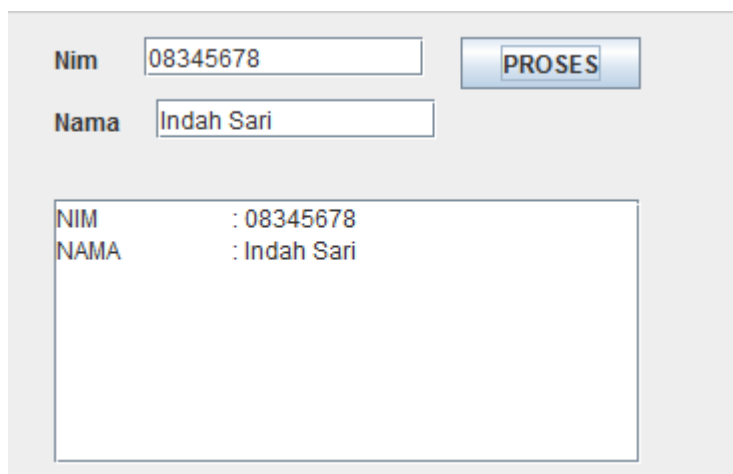
operasi matematika jika pilihan dihilangkan (centang hilang) hasil akan dikosongkan lagi.

```
private void cb01ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int n1, n2, hs;
    n1 = Integer.parseInt(nil1.getText());
    n2 = Integer.parseInt(nil2.getText());
    hs = n1 + n2;
    if (cb01.isSelected())
        cb1.setText(" " + hs);
    else
        cb1.setText(" ");
}
```

## TextArea

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    hasilText.setText("NIM \t: "+nimField.getText()+"\nNAMA\t: "+namaField.getText());  
}
```

Hasil:

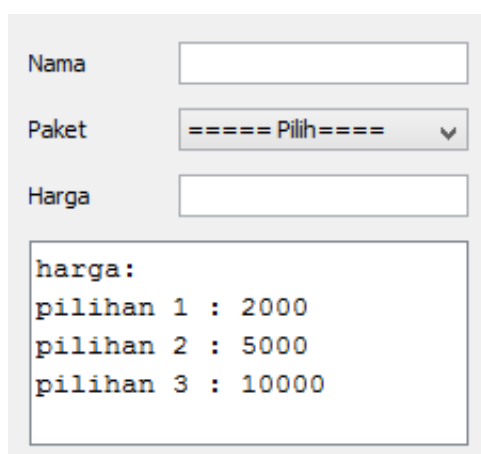


Nim 08345678 PROSES

Nama Indah Sari

NIM : 08345678  
NAMA : Indah Sari

## Combobox




Nama

Paket ===== Pilih =====

Harga

harga:  
pilihan 1 : 2000  
pilihan 2 : 5000  
pilihan 3 : 10000



Nama

Paket PILIHAN2

Harga 5000

harga:  
pilihan 1 : 2000  
pilihan 2 : 5000  
pilihan 3 : 10000

```
private void comboActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (combo.getSelectedIndex() == 0)
        hrg.setText("");
    else if (combo.getSelectedIndex() == 1)
        hrg.setText("2000");
    else if (combo.getSelectedIndex() == 2)
        hrg.setText("5000");
    else
        hrg.setText("10000");
}
```

### C. LATIHAN SOAL

Buat desain dan source codenya untuk tampilan di bawah ini!

**Nilai Pemrograman II** = 10% x hadir + 20% x tugas + 30% x uts + 40% x uas

**Biaya/bln:**

Teknik Informatika	: 300000
Ekonomi	: 200000
MIPA	: 225000

**Grade:**

Nilai pemrograman kurang dari 50, grade E

Min nilai pemrograman 50, grade D

60, grade C

70, grade B

80, grade A

**Biaya Reg,** 100000

**Biaya UTS,** 300000

**Biaya UAS,** 300000

**Total** = biaya Reg + biaya UTS + biaya UAS

Saat tombol **HITUNG** di klik tampilkan di textarea:

NIM, nama, jenis kelamin, jurusan, semester, nilai pemrograman II, grade, total

**D. DAFTAR PUSTAKA**

Budiharto, W. (2004). *Pemrograman Web Mneggunakan J2EE*. Jakarta:

Elexmedia Komputindo.

JENI, T. P. (2007). JENI 1-6.

Wijono, S. H., Suharto, B. H., & Wijono, M. S. (2006). *Pemrograman Java*

*Servlet dan JSP dengan Netbeans*. Yogyakarta: C.V ANDI OFFSET.