



# GUI JAVA

Dosen : Ibu Nurhayati  
Asdos : Dwindi Faradita

# Komponen Dasar Swing

Komponen Dasar  
Swing Java

Top-level Container

Intermediate Container

Atomic Component

Layout Manager

Event Handling

# Top-level Container

- merupakan container dasar untuk peletakan komponen lainnya.
- Contoh implementasinya berupa class **Jframe**, **Jdialog**, dan **Japplet**
- Contoh kode memanggil class tersebut (sesuai latihan sebelumnya)

– Pada Class FrameModal :

```
extends JFrame
```

– Pada Class DialogPengirim :

```
extends JDialog{
```

# Intermediate Container

- merupakan container perantara untuk peletakan komponen lainnya yang berguna untuk membantu pengaturan tata letak
- Untuk dapat dijalankan harus ditambahkan ke Top Level Container agar tampil.
- Contoh implementasinya adalah class **JPanel**.
- Contoh kode memanggil class tersebut

- Pada Class FrameModal:

```
private JPanel contentPane;
```

```
contentPane = new JPanel();
```

- Pada Class DialogPengirim :

```
private final JPanel contentPanel = new JPanel();
```

# Atomic Component

- merupakan komponen yang memiliki fungsi yang spesifik, dimana umumnya user langsung berinteraksi dengan komponen jenis ini.
- Contoh implementasinya berupa class **Jbutton**, **Jlabel**, **JtextField**, dan **JtextArea**
- Contoh kode memanggil class tersebut (sesuai latihan sebelumnya)
  - Pada **Class DialogPengiriman**:

```
private JTextField txtNama;  
private JTextArea textAlamat;  
private JLabel lblEmail;  
private JTextField txtEmail;  
private JButton btnSimpan;  
private JButton btnBatal;  
private JLabel lblWall;
```

# Layout Manager

- berfungsi untuk mengatur bagaimana tata letak atau posisi komponen yang akan diletakkan, satu sama lain di dalam suatu container.
- Contoh implementasinya berupa class **BorderLayout**, **BoxLayout**, **FlowLayout**, **GridBagLayout**, dan **GridLayout**
- Contoh kode memanggil class tersebut (sesuai latihan sebelumnya)
  - Pada **Class frameModal**:

```
setBounds(100, 100, 779, 581);
```

```
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
```

```
lblWall.setBounds(0, 0, 777, 533);
```

# Event Handling/ Listener

- Event listener menerima berita dari **event-event** dan **proses-proses** interaksi user.
- Contohnya Ketika tombol ditekan, listener akan mengendalikan dengan menampilkan sebuah informasi yang berguna untuk user.
- Contoh kode memanggil class tersebut (sesuai latihan sebelumnya)
  - Pada **Class frameModal**:

```
mntmPengirim.addActionListener(new ActionListener()
{
    @Override
    public void actionPerformed(ActionEvent e)
    {
        dialogPengirim dialP = new dialogPengirim();
        dialP.setVisible(true);
    }
});
```



# ATOMIC COMPONENT



# Objek JLabel

- Objek ini dapat digunakan untuk menampilkan teks-teks pendek yang tidak dapat diedit, image atau keduanya. JLabel berfungsi memberikan keterangan pada komponen Java lain (umumnya obyek dari kelas JTextField).

- Hirarkhi turunan kelas jlabel

```
javax.swing.JComponent  
└─ javax.swing.JLabel
```

- Label mempunyai latar belakang yang transparan sehingga dapat menyesuaikan dengan latar belakang container classes dimana label diletakkan. Penggunaan metoda setBackground tidak memberikan dampak perubahan warna pada latar belakang label.

# Objek JLabel (2)

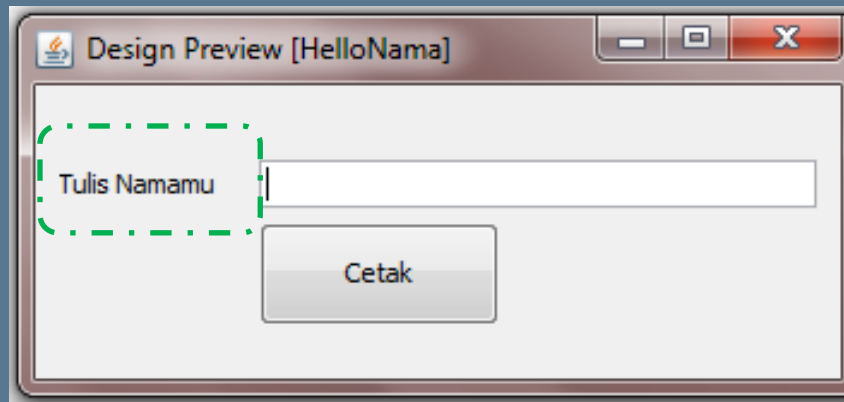
- **Konstruktor Kelas JLabel**

Konstruktor	Keterangan
JLabel()	Mengkonstruksi obyek label tanpa teks dan ikon.
JLabel(String text)	Mengkonstruksi obyek label dengan teks.
JLabel(String teks, int hAlignment)	Mengkonstruksi obyek label dengan teks serta menentukan penjumlahan secara horisontal.
JLabel(Icon icon)	Mengkontruksi obyek label dengan ikon.
JLabel(Icon icon, int hAlignment)	Mengkonstruksi obyek label dengan ikon serta menentukan penjumlahan secara horisontal.
JLabel(String teks, Icon icon, int hAlignment)	Mengkonstruksi obyek label dengan teks dan ikon serta menentukan penjumlahan secara horisontal.

- Kelas JLabel memiliki metoda yang diwarisi dari kelas Jcomponent, yaitu: getText(); setText(); getIcon(); setIcon(); etc

# Objek JLabel (3)

- Contoh:



- Kode:

```
private JLabel lblWall;
```

(manual; latihan sebelumnya)

```
lblWall = new JLabel("");
```

 **Swing Controls**

label Label

(Palette)

# Objek JButton

- Button biasanya digunakan untuk menerima input dari user berupa penekanan button tersebut.
- Hirarki dari objek ini:

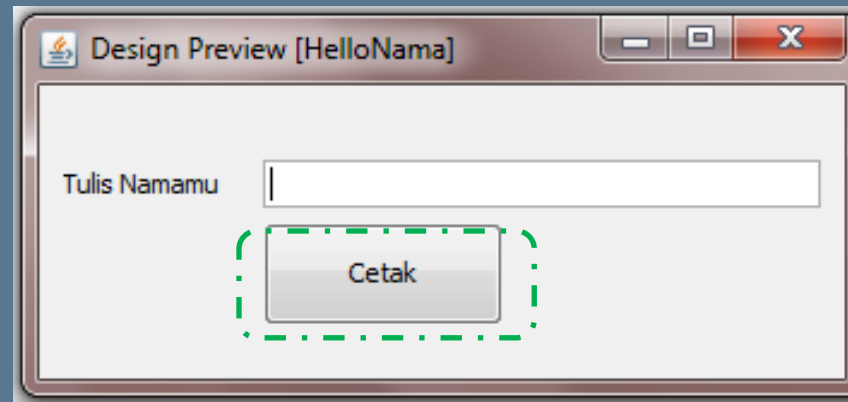
```
javax.swing.JComponent
└─ javax.swing.AbstractButton
    └─ javax.swing.JButton
```

- Konstruktor yang dapat digunakan:

Konstruktor	Keterangan
JButton()	Menciptakan sebuah objek JButton tanpa judul dan ikon
JButton(Icon ikon)	Menciptakan sebuah objek JButton yang dilengkapi dengan ikon
JButton(String teks)	Menciptakan sebuah objek JButton yang dilengkapi dengan judul
JButton(String teks, Icon ikon)	Menciptakan sebuah objek JButton yang dilengkapi dengan judul dan ikon

# Objek JButton (2)

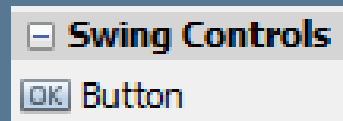
- Contoh:



- Kode:

```
private JButton btnSimpan;  
btnSimpan = new JButton("Simpan");
```

(manual; latihan sebelumnya)



(Palette)

# Objek JTextField

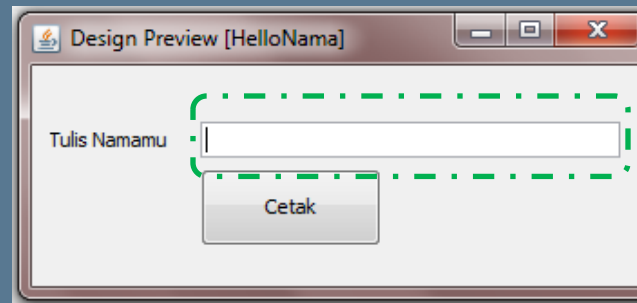
- Objek ini digunakan sebagai salah satu obyek untuk menginput string melalui keyboard atau menampilkan string
- Turunan dari JComponent → JTextComponent → JTextField
- Metoda yg dapat digunakan: getText(), setText(String text) dan setEditable(boolean bValue).
- Konstruktor yang dapat digunakan:

**Konstruktor Kelas JTextField**

Konstruktor	Keterangan
JTextField()	Mengkonstruksi obyek dengan teks kosong.
JTextField(int column)	Mengkonstruksi obyek dengan teks kosong dan jumlah kolom ditentukan oleh parameter columns.
JTextField(String text)	Mengkonstruksi obyek dengan teks ditentukan oleh nilai parameter text.
JTextField(String text, int columns)	Mengkonstruksi obyek dengan teks ditentukan oleh nilai parameter text dan jumlah kolom oleh nilai parameter columns.

# Objek JTextField (2)

- Contoh:



- Kode:

```
private JTextField txtEmail;
```

(manual; latihan sebelumnya)

```
txtEmail = new JTextField();
```

☒ **Swing Controls**

☐ Text Field

(Palette)

# Objek JTextArea

- TextArea ini pada dasarnya sama dengan TextField.
- Namun objek ini mampu menampilkan teks lebih dari satu baris dan fungsi wrapping.
- Sebagaimana kelas JTextField, kelas JTextArea adalah juga turunan dari kelas JTextComponent.

```
javax.swing.JComponent  
└─ javax.swing.JTextComponent  
    └─ javax.swing.JTextArea
```

- Objek ini tidak mendukung scrolling. Untuk dapat mendukung scrolling, Anda perlu meletakkannya di obyek kelas JScrollPane.



# Objek JTextArea (2)

- Tabel Konstruktur JTextArea

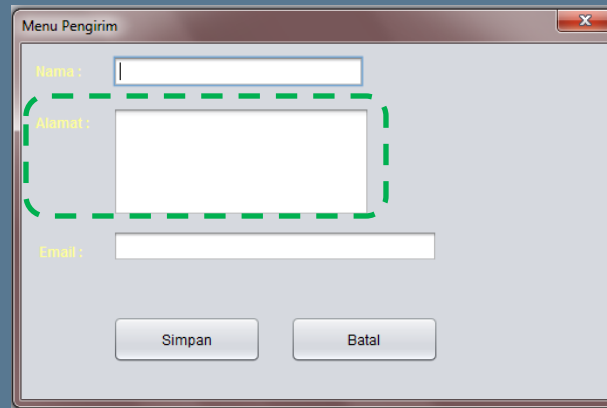
Konstruktor	Keterangan
JTextArea()	Mengkonstruksi obyek text area.
JTextArea(int rows, int cols)	Mengkonstruksi obyek text area baru dengan menentukan jumlah baris dan kolom
JTextArea(String text)	Mengkonstruksi obyek text area baru sekaligus menampilkan teks yang telah ditentukan.
JTextArea(String text, int rows, int cols)	Mengatur ulang jumlah kolom untuk obyek text area ini.

- Tabel Metoda JTextArea

Metoda	Keterangan
insert(String text, int pos)	Menyisipkan teks di posisi yang ditentukan oleh nilai parameter pos.
setWrapStyleWord(boolean word)	Menentukan model pengaturan penulisan kata apabila baris teks telah melampaui area teks.
setLineWrap(boolean wrap)	Menentukan apakah teks akan berpindah ke baris berikutnya apabila telah mencapai area teks.
setRows(int rows)	Mengatur ulang jumlah baris untuk obyek text area ini.
setColumns(int cols)	Mengatur ulang jumlah kolom untuk obyek text area ini.
replaceRange(String str, int start, int end)	Mengganti bagian teks tertentu dengan string str yang dimulai dari int start sampai end.
getRows()	Mengembalikan jumlah baris dari text area.
append(String text)	Menambahkan teks di akhir dokumen dari obyek text area.

# Objek JTextArea (3)

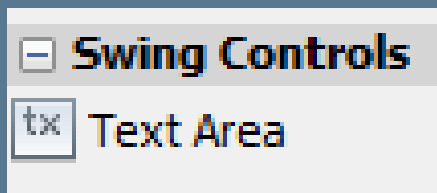
- Contoh:



- Kode:

```
private JTextArea textAlamat;  
  
textAlamat = new JTextArea();
```

(manual; dari latihan sebelumnya)



(Palette)

# Objek JPasswordField

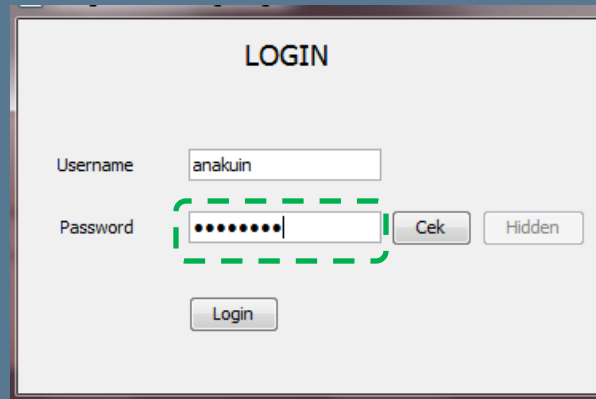
- Field Password Field diimplementasikan berupa class **JPasswordField**.
- Class ini sendiri merupakan turunan dari class JTextField sehingga cara penggunaannya sama.

```
java.lang.Object
  java.awt.Component
    java.awt.Container
      javax.swing.JComponent
        javax.swing.text.JTextComponent
          javax.swing.JTextField
            javax.swing.JPasswordField
```

- Pembedanya adalah teks yang ditampilkannya bukan berupa teks yang diinput oleh user melainkan teks yang berupa **karakter** tertentu, hal ini digunakan agar orang lain tidak tahu apa yang sedang diketikkan oleh user.

# Objek JPasswordField (2)

- Contoh:



- Kode:

```
private JPasswordField passwordField; (manual)  
passwordField = new JPasswordField(10);
```



Latihan  
1

# Objek JCheckBox

- Objek ini digunakan untuk menampilkan pilihan kepada user, dan user bebas menentukan pilihannya ( $\geq 0$  pilihan).
- Objek ini mempunyai dua macam keadaan (state) atau nilai (value) yaitu on/off atau true/false.
- Anda dapat berpindah (toggle) dari satu keadaan (nilai) ke keadaan (nilai) yang lain dengan cara mengklik untuk memberi tanda centang (selected) atau menghilangkan tanda centang (deselected) di check box
- hirarki turunan JCheckBox

```
javax.swing.JComponent  
└─ javax.swing.AbstractButton  
    └─ javax.swing.JToggleButton  
        └─ javax.swing.JCheckBox
```

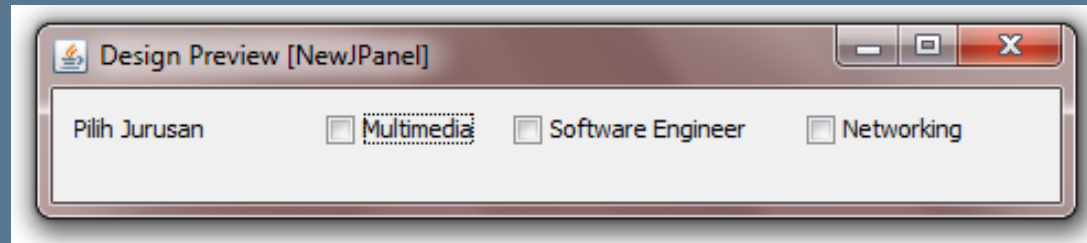
# Objek JCheckBox (2)

- Tabel kontruktor kelas JCheckBox :

Konstruktor	Keterangan
JCheckBox()	Mengkonstruksi obyek check box tanpa teks dan icon.
JCheckBox(Icon icon)	Mengkonstruksi obyek check box dengan ikon.
JCheckBox(Icon icon, boolean nilai)	Mengkonstruksi obyek check box dengan ikon serta menentukan apakah check box dalam kondisi dipilih atau tidak.
JCheckBox(String text)	Mengkonstruksi obyek check box dengan text.
JCheckBox(String text, boolean nilai)	Mengkonstruksi obyek check box dengan text serta menentukan apakah check box dalam kondisi dipilih atau tidak.
JCheckBox(String text, Icon icon)	Mengkonstruksi obyek check box dengan text dan ikon.
JCheckBox(String text, Icon icon, boolean nilai)	Mengkonstruksi obyek check box dengan text dan ikon serta menentukan apakah check box dalam kondisi dipilih atau tidak

# Objek JCheckBox (3)

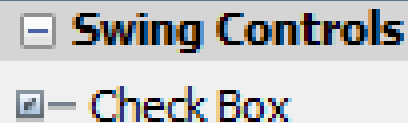
- Contoh:



- Kode:

```
private JCheckBox cekjurusan;  
cekjurusan = new JCheckBox();
```

(manual)



(Palette)

Latihan 2

# Objek JRadioButton

- Objek ini mempunyai dua keadaan atau nilai yaitu on/off atau true/false.
- Kelas JRadioButton adalah turunan dari kelas JToggleButton.

```
javax.swing.JComponent
└─ javax.swing.AbstractButton
    └─ javax.swing.JToggleButton
        └─ javax.swing.JRadioButton
```

- Anda dapat mengelompokkan beberapa radio button menggunakan kelas **ButtonGroup**.
  - Ketika Anda mengklik salah satu radio button (selected), maka radio button yang lain menjadi kosong (unselected).
- Apabila Anda tidak menggunakan ButtonGroup, maka setiap radio button adalah independen (radio button satu tidak akan mempengaruhi radio button lainnya).



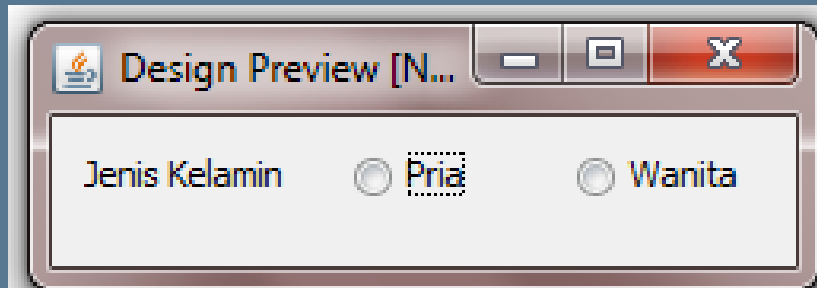
# Objek JRadioButton (2)

- Konstruktor yang dapat digunakan:

Konstruktor	Keterangan
JRadioButton()	Mengkonstruksi obyek radio button tanpa teks dan ikon
JRadioButton(Icon icon)	Mengkonstruksi obyek radio button dengan ikon.
JRadioButton(Icon icon, boolean nilai)	Mengkonstruksi obyek radio button dengan ikon serta menentukan apakah radio button dalam kondisi dipilih atau tidak.
JRadioButton(String text)	Mengkonstruksi obyek radio button dengan text.
JRadioButton(String text, boolean nilai)	Mengkonstruksi obyek radio button dengan text serta menentukan apakah radio button awalnya dipilih atau tidak.
JRadioButton(String text, Icon icon)	Mengkonstruksi obyek radio button dengan text dan ikon.
JRadioButton(String text, Icon icon, boolean nilai)	Mengkonstruksi obyek radio button dengan text dan ikon serta menentukan apakah radio button dalam kondisi dipilih atau tidak

# Objek JRadioButton (3)

- Contoh:

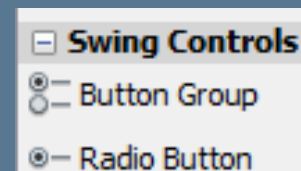


- Kode:

```
JRadioButton birdButton = new JRadioButton(birdString);
```

(manual)

Latihan 3



(Palette)

# Objek JComboBox

- Kelas JComboBox merupakan turunan dari kelas JComponent.
- Combo box juga disebut dengan daftar pilihan (choice list) yang berisi daftar elemen (item) yang dapat dipilih oleh pengguna program (user).
- Objek ini akan menghasilkan dua jenis event yaitu **ItemEvent** dan **ActionEvent**. Di mana, ItemEvent akan dipicu pertama kali, baru kemudian ActionEvent.
- Tabel Konstruktor

Konstruktor	Keterangan
JComboBox()	Mengkonstruksi obyek combo box dengan model data default.
JComboBox(Object[] arrayItem)	Mengkonstruksi obyek combo box dengan item atau elemennya diperoleh dari larik obyek.
JComboBox(ComboBoxModel model)	Mengkonstruksi obyek combo box dengan item atau elemennya diperoleh dari ComboBoxModel.

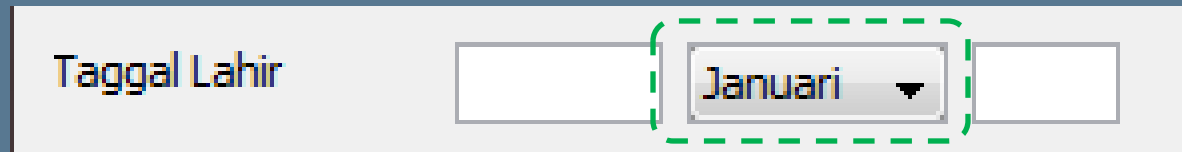
# Objek JComboBox (2)

- Tabel Metoda

Metoda	Keterangan
<code>addItem(Object item)</code>	Menambahkan item atau elemen di daftar item dari combo box.
<code>getItemAt(int index)</code>	Mengembalikan item atau elemen dari indek tertentu.
<code>getItemCount()</code>	Mengembalikan jumlah (int) item atau elemen dari daftar combo box.
<code>getSelectedIndex()</code>	Mengembalikan indek dari item atau teks yang dipilih.
<code>getSelectedItem()</code>	Mengembalikan item atau elemen yang dipilih.
<code>removeItem(Object item)</code>	Menghapus sebuah item atau elemen dari daftar combo box.
<code>removeItemAt(int index)</code>	Menghapus sebuah item atau elemen pada indek tertentu dari daftar combo box.
<code>setMaximumRowCount(int count)</code>	Menampilkan jumlah baris maksimum yang dapat ditampilkan oleh combo box
<code>insertItemAt(Object item, int index)</code>	Menambahkan item di daftar item combo box pada indek yang telah ditentukan.
<code>removeAllItem()</code>	Menghapus semua item atau elemen dari daftar.

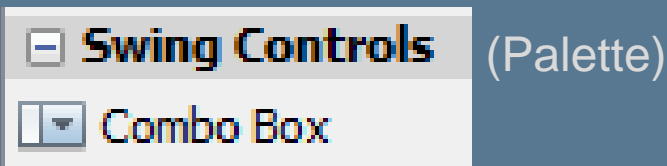
# Objek JComboBox (3)

- Contoh:



- Kode:

```
private JComboBox comboBox; (manual)  
comboBox = new JComboBox(rgb);
```



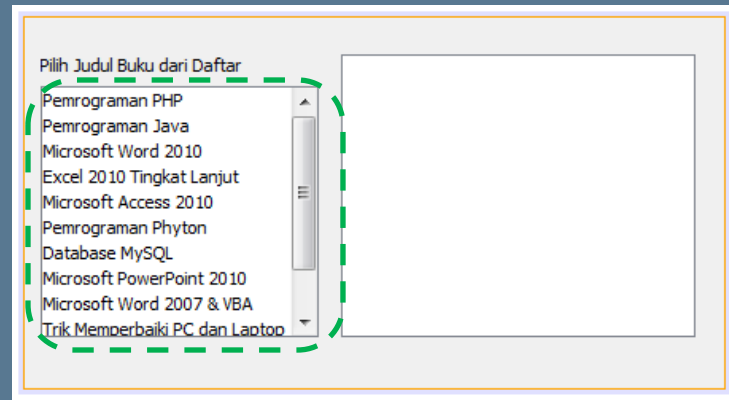
# Objek JList

- List Box juga digunakan untuk menampilkan sekumpulan daftar pilihan yang dapat dipilih. User dapat memilih  $\geq 1$  nilai string yang terdapat di JList.
- Berbeda dengan combo box, list box selalu menampilkan daftar pilihannya.
- Metoda kelas Jlist, di antaranya: `getSelectedIndex()`, `getSelectedValue()`, `setSelectedValues()`.
- tabel daftar konstruktor

Konstruktor	Keterangan
<code>JList()</code>	Mengkonstruksi obyek list tanpa memiliki daftar obyek (elemen).
<code>JList(Object[] listData)</code>	Mengkonstruksi obyek list yang berisi elemen dari obyek array (larik).

# Objek JList (2)

- Contoh:

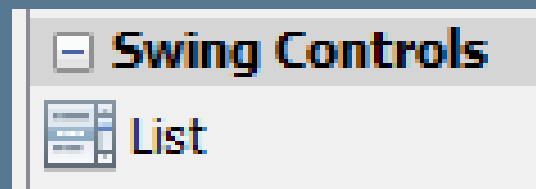


- Kode:

```
private JList list;
```


(manual)

```
list = new JList(judulBuku);
```



(Palette)

Latihan  
4



# Perbedaan JComboBox dan JList

- Obyek combo box hanya memperbolehkan **pemilihan satu elemen/item**,
- Sedangkan obyek list memperbolehkan **pemilihan lebih dari satu elemen/item**.

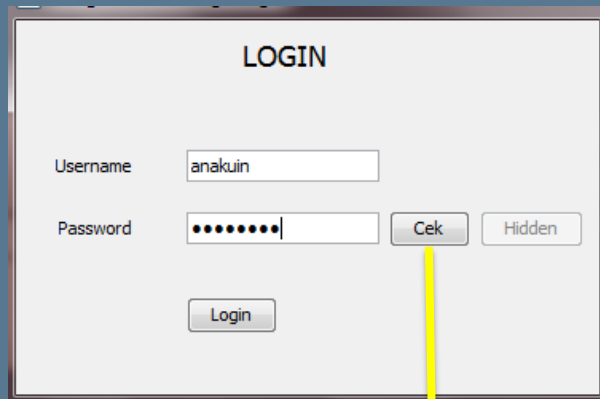




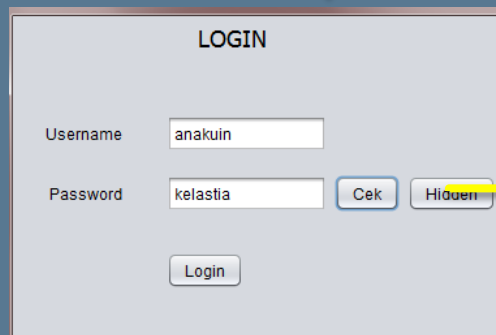
# LATIHAN

# Latihan 1

Membuat Halaman Login (Username: anakuin;  
password: kelastia)

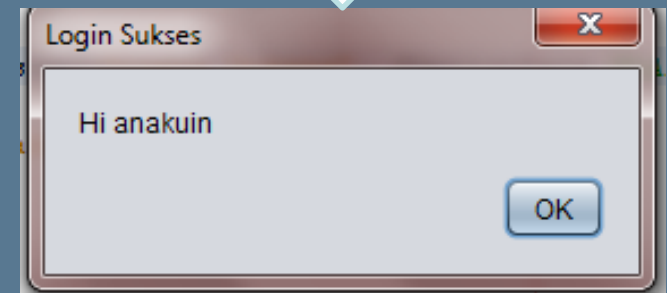


A screenshot of a web login form titled "LOGIN". It contains two input fields: "Username" with the value "anakuin" and "Password" with masked characters "••••••". To the right of the password field are two buttons: "Cek" and "Hidden". Below the password field is a "Login" button. A yellow arrow points from the "Cek" button to the second form below.



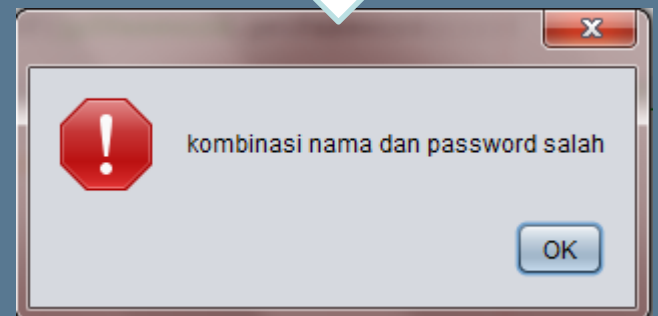
A screenshot of the same login form, but the password field now contains the text "kelastia". The "Cek" button is highlighted with a blue border. A yellow arrow points from the "Cek" button of this form to the "Login Sukses" dialog box on the right.

Username & Password sesuai



A dialog box titled "Login Sukses" with a close button (X) in the top right corner. The main text area displays "Hi anakuin". At the bottom right is an "OK" button.

Username & Password tidak sesuai



A dialog box with a red octagonal warning icon containing an exclamation mark. The text inside says "kombinasi nama dan password salah". There is an "OK" button at the bottom right.

# Jawaban Latihan 1

## Halaman Login

1) Ubah nama variabel

Label	Variabel
(jTextField1	jtfUsername
jPasswordField1	jpfPassword
jButton1	btnCek
jButton2	btnLogin

2) Action pada Button Cek

Aksi yang dilakukan oleh button cek adalah membuat password tampak dan button Hidden jadi aktif/ dapat diklik.

```
private void btnCekActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(btnCek.isEnabled()){  
        jpfPassword.setEchoChar((char)0);  
        btnHidden.setEnabled(true);  
    }  
}
```

# Jawaban Latihan 1

## 3) Action pada Button Hidden

Aksi yang dilakukan oleh button hidden adalah membuat password tidak tampak dan button Hidden jadi non-aktif/ tidak dapat diklik.

```
private void btnHiddenActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(btnHidden.isEnabled()) {  
        jpfPassword.setEchoChar('*');  
        btnHidden.setEnabled(false);  
    }  
}
```

# Jawaban Latihan 1

## 4) Action pada Button Login

melakukan pengecekan terhadap data terinput pada field username dan password dengan data yang tersedia.

Jika sesuai akan berhasil login, jika tidak maka akan ada peringatan

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
  
    //data  
    String user = "anakuin";  
    String pass = "kelastia";  
  
    if(user.equalsIgnoreCase(jtfUsername.getText())  
        && pass.equalsIgnoreCase(String.valueOf(jpfPassword.getPassword()))){  
        this.setVisible(false);  
        JOptionPane.showMessageDialog(null, "Hi "+user, "Login Sukses", JOptionPane.PLAIN_MESSAGE);  
    }else{  
        JOptionPane.showMessageDialog(null, "kombinasi nama dan password salah",  
            "", JOptionPane.ERROR_MESSAGE);  
  
        //reset  
        jtfUsername.setText("");  
        jtfUsername.requestFocus();  
        jpfPassword.setText("");  
    }  
}
```

Kembali

# Notes

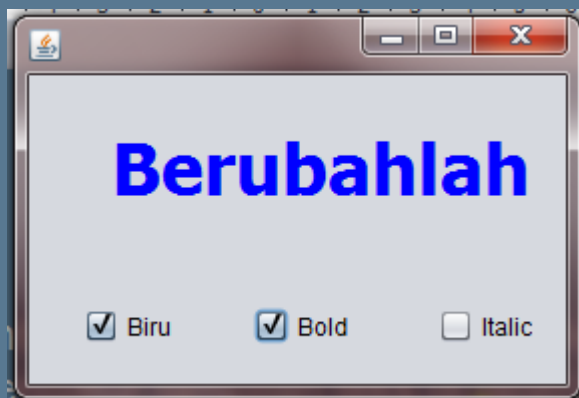
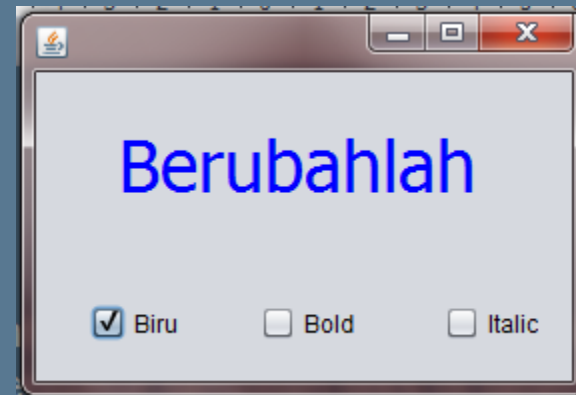
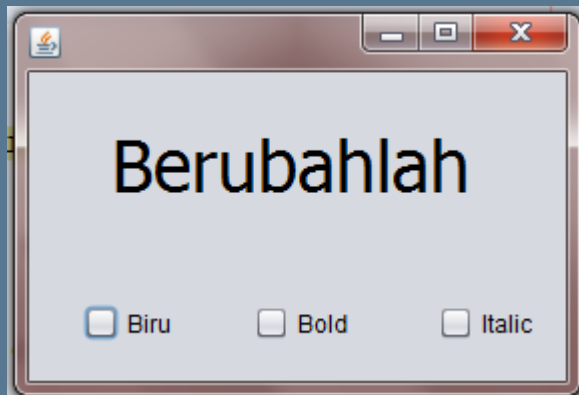
Kode untuk melakukan enter secara otomatis ke field/button/etc selanjutnya adalah dengan klik kanan > event > key > KeyPressed

Contoh:

```
private void  
PasswordKeyPressed(java.awt.event.KeyEvent evt) {  
    if(evt.getKeyCode() == evt.VK_ENTER)  
        Login.doClick();  
}
```

# Latihan 2

Mengatur Teks (JCheckBox digunakan untuk mengatur tampilan teks di label).



# Jawaban Latihan 2

## Halaman Pengaturan Teks

- 1) Buat UI yang sesuai kebutuhan dan Ubah nama variabel
- 2) Action pada setiap CheckBox

Jika user memilih box warna biru, teks akan berubah menjadi warna biru.

Jika user memilih box Bold, maka teks akan dibentuk menjadi Bold Teks

Jika user memilih box Italic, maka teks akan dibentuk menjadi Italic Teks.

Kondisi hanya dilakukan, jika box dalam keadaan terceklist.



# Jawaban Latihan 2

## Halaman Pengaturan Teks

- 3) Kode yang dibutuhkan disetiap checkbox sama.
- 4) Klik kanan pada setiap checkbox > event > action > ActionPerformed

```
Color warna= Color.BLACK;
int nilaiBold = Font.PLAIN;
int nilaiItalic = Font.PLAIN;

if (jcbWarna.isSelected())
    warna = jcbWarna.isSelected() ? Color.BLUE : Color.BLACK;

if (jcbBold.isSelected())
    nilaiBold = jcbBold.isSelected() ? Font.BOLD : Font.PLAIN;

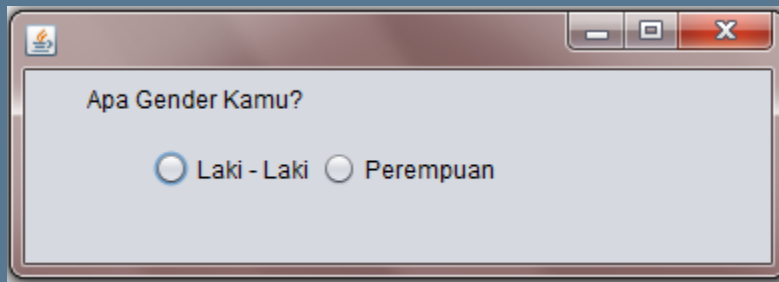
if (jcbItalic.isSelected())
    nilaiItalic = jcbItalic.isSelected() ? Font.ITALIC : Font.PLAIN;

j1Hasil.setFont(new Font("Tahoma", nilaiBold + nilaiItalic, 36));
j1Hasil.setForeground(warna);
```

Kembali

# Latihan 3

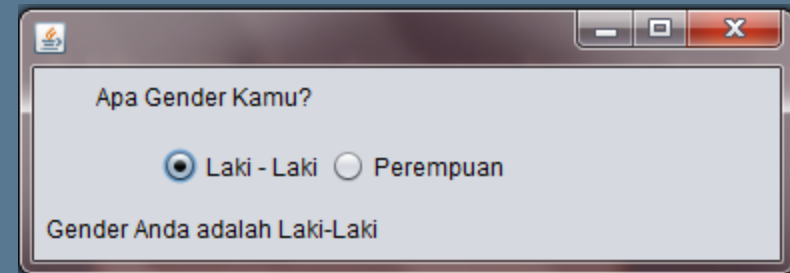
Memilih Gender (user **hanya** dapat memilih satu gender)



Apa Gender Kamu?

☐ Laki - Laki ☐ Perempuan

This screenshot shows a standard Windows-style dialog box with a title bar containing a small icon and standard minimize, maximize, and close buttons. The main area contains the text "Apa Gender Kamu?" followed by two radio button options: "Laki - Laki" and "Perempuan". Both radio buttons are currently unselected.

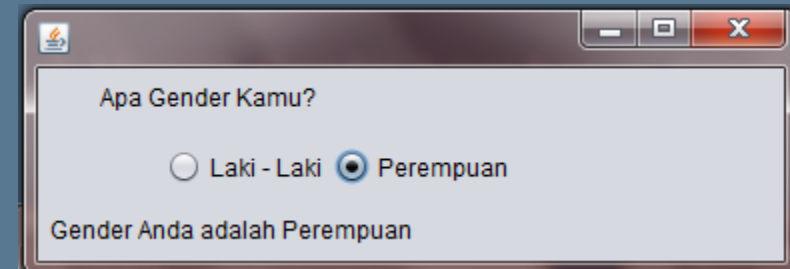


Apa Gender Kamu?

☒ Laki - Laki ☐ Perempuan

Gender Anda adalah Laki-Laki

This screenshot shows the same dialog box as the first one, but with the "Laki - Laki" radio button selected (indicated by a blue dot). Below the options, the text "Gender Anda adalah Laki-Laki" is displayed.



Apa Gender Kamu?

☐ Laki - Laki ☒ Perempuan

Gender Anda adalah Perempuan

This screenshot shows the same dialog box with the "Perempuan" radio button selected. The feedback text at the bottom now reads "Gender Anda adalah Perempuan".

# Jawaban Latihan 3

## Halaman Pengaturan Teks

- 1) Buat UI yang sesuai kebutuhan dan Ubah nama variabel (1 buah label, 2 buah radio button, dan 1 buah radio button group)
- 2) Action pada Radio Button  
User hanya dapat memilih satu gender saja, laki-laki atau perempuan.  
Jika user memilih laki-laki, maka akan muncul teks keterangan pilihan user. begitu pula saat memilih pilihan perempuan.

# Jawaban Latihan 3

## Halaman Pengaturan Teks

- 3) Kebutuhannya, saat klik pilihan, akan muncul teks berisi keterangan dari hasil pilihan user tersebut. Bisa menggunakan MouseClicked atau ActionPerformed
- 4) Kode:

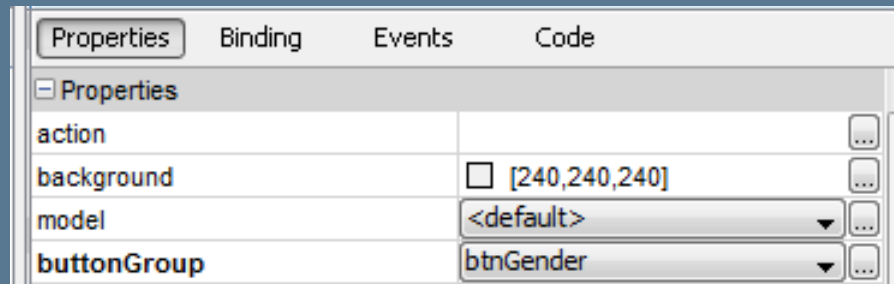
```
private void jrbLakiLakiMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    jlHasil.setText("Gender Anda adalah Laki-Laki");  
}
```

```
private void jrbPerempuanActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if(jrbPerempuan.isSelected())  
        jlHasil.setText("Gender Anda adalah Perempuan");  
}
```

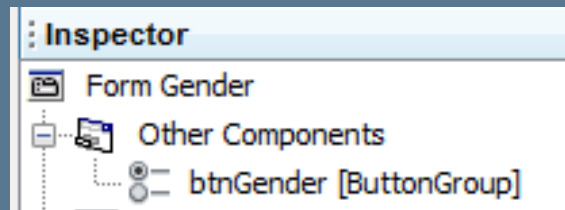
# Jawaban Latihan 3

## Halaman Pengaturan Teks

- 5) Untuk membatasi user **hanya** dapat memilih satu pilihan, gunakan Radio Button Group.  
Klik Radio Button > Properties > buttonGroup > Pilih Button Groupnya

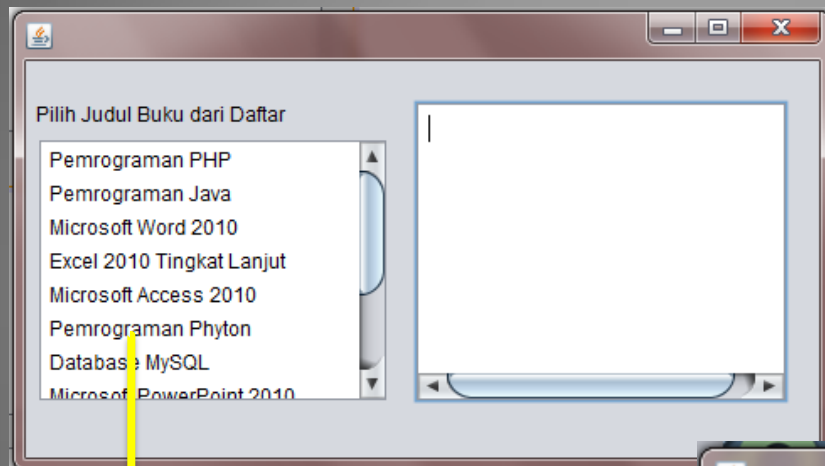


- 6) Untuk mengganti nama Button Group: klik Inspector (biasanya disebelah kiri bawah) > Pilih Button Group > Klik Kanan > Change Variabel Name

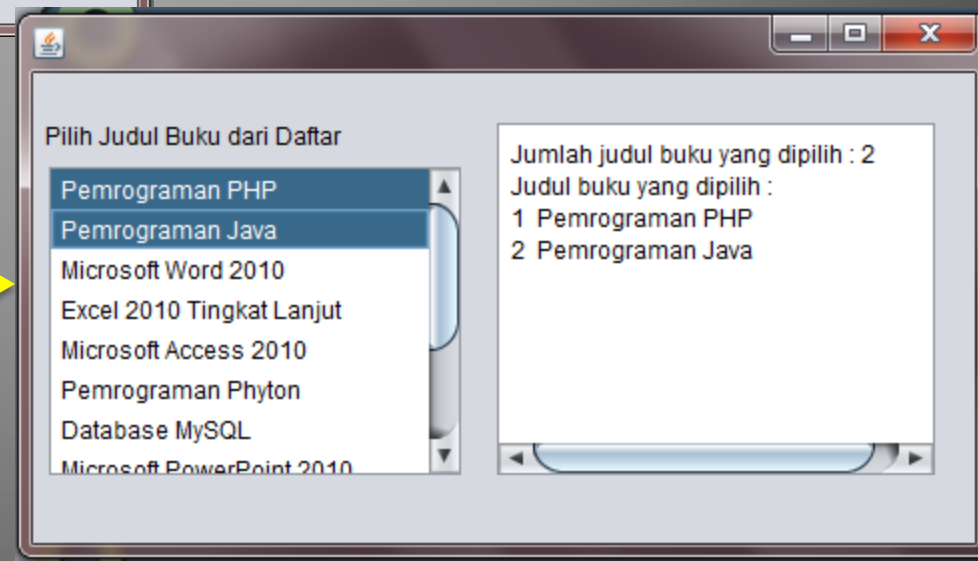


# TUGAS 1

## Membuat Halaman Pemilihan Judul Buku



Saat memilih/klik judul buku, maka data buku akan tampil sesuai judul buku yang dipilih



# TUGAS 2

Buatlah Aplikasi Segitiga Pascal, jika obyek Button “Proses” di klik, maka selanjutnya hasil bilangan Prima akan ditampilkan dan dimana user dapat menentukan bilangan awal dan jumlah barisnya.

