

# Desktop Programming



Hello Java



# Overview

- Java AWT/Swing
  - Brief history, introduction to the main packages
- Fundamentals of Swing
  - Containers
  - Components
  - Layouts
  - Event-driven programming



# Hello Java!

```
12  public class Main {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.println("Hello");
20      }
21
22  }
```



# Java AWT

- Abstract Windowing Toolkit
- Original Java GUI API
- Very limited in capability
  - Few components
  - API not well structured, particularly event handling for user actions
  - Not entirely portable (used native widgets)



- Java Foundation Classes (or “Swing”)
  - Replacement for AWT (although does share some classes)
  - Also provide basis for developing new GUI features (which are being continually added)
- What does Swing include?
  - 100% Java
  - Swing components (more, and more sophisticated)
  - Pluggable Look and Feel Support
  - Accessibility API
  - Better graphics support (Java 2D)
  - Drag and Drop



- Disadvantages
  - Can be slow (resource hungry)
  - Large complex API (big learning curve)
  - Many features best suited for GUI builders, IDEs
- Aim of the next few lectures is to introduce the basic concepts
  - Provide you with background so can continue studies yourself
- Important to use Swing and not AWT
  - Swing is the recommended way to build Java GUIs



# Introduction to GUI Programming

- What are the stages in building a GUI application?
- Design the user interface
  - Organising pre-built GUI components to build windows, dialogs
  - E.g buttons, tables, menus, etc
- Writing the application logic
  - What does the application do?
- Writing event-handling code to tie the GUI components to the application logic
  - More on event-handling in next lesson...



# Introduction to GUI Programming

- Essentially, JFC/Swing provides a framework which consists of:
  - A number of GUI components that can be used to build a user interface (`javax.swing`)
  - An event-handling framework for tying user actions to application code (`javax.swing.event`)
- Occasionally use classes from the AWT equivalents (`java.awt`, `java.awt.event`)
  - Some Swing classes inherit from originals
  - Distinguish Swing versions from AWT versions with “J” prefix.





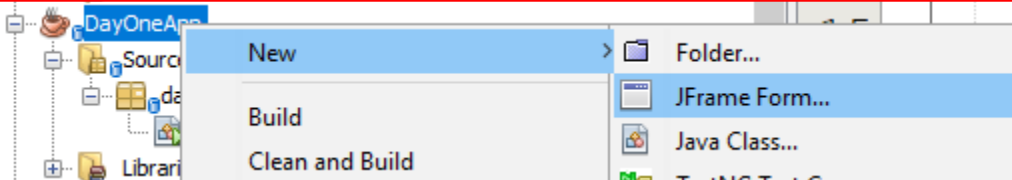
# Building a GUI

- A GUI is built in layers.
- Bottom most layer is the window (`Container`)
  - Contains all other components
  - Can provide basic features like maximise/minimise buttons, title bar, menu bar, etc
- On top of this are layered (`Component`)
  - Components, e.g. buttons, text fields
  - or intermediate containers, e.g. panels
- Arrangement of components in a contained is handled by a *layout manager*
  - Its job is to instruct components on how to arrange themselves so the GUI is drawn correctly.

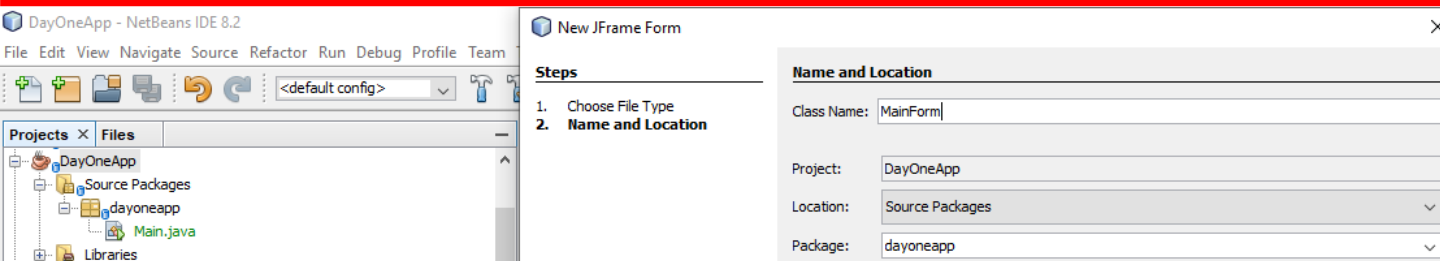


# Building a GUI

Klik kanan project, pilih New > JFrame Form



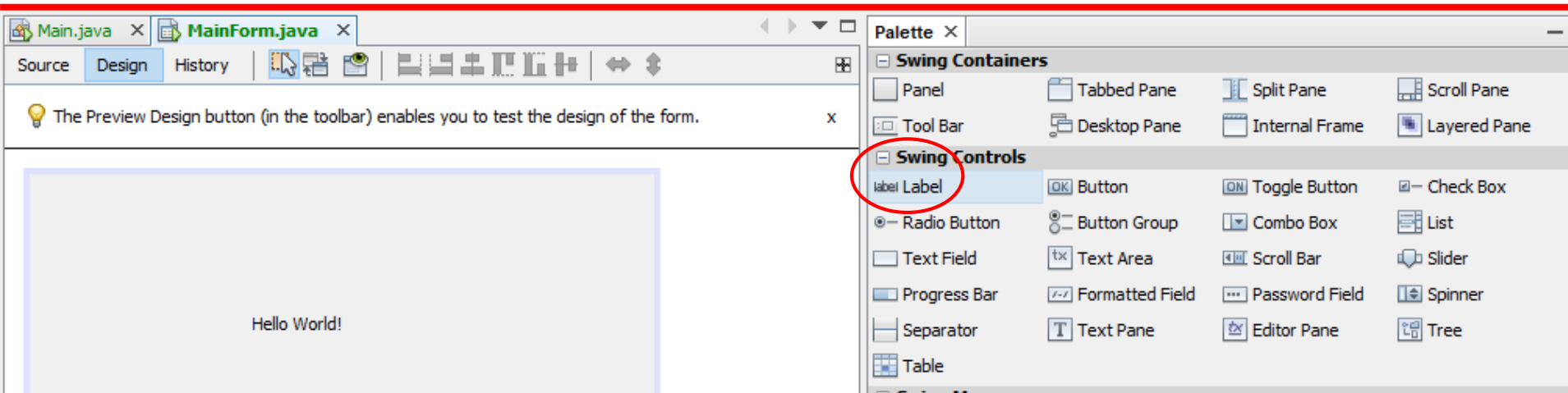
Isi “Class Name”, “Location” dan “Package”





# Building a GUI: Component

Drag komponen “Label” pada pallette ke form.  
Klik dua kali untuk mengganti teks.



Klik kanan > “Run File”.





# Building a GUI: Properties

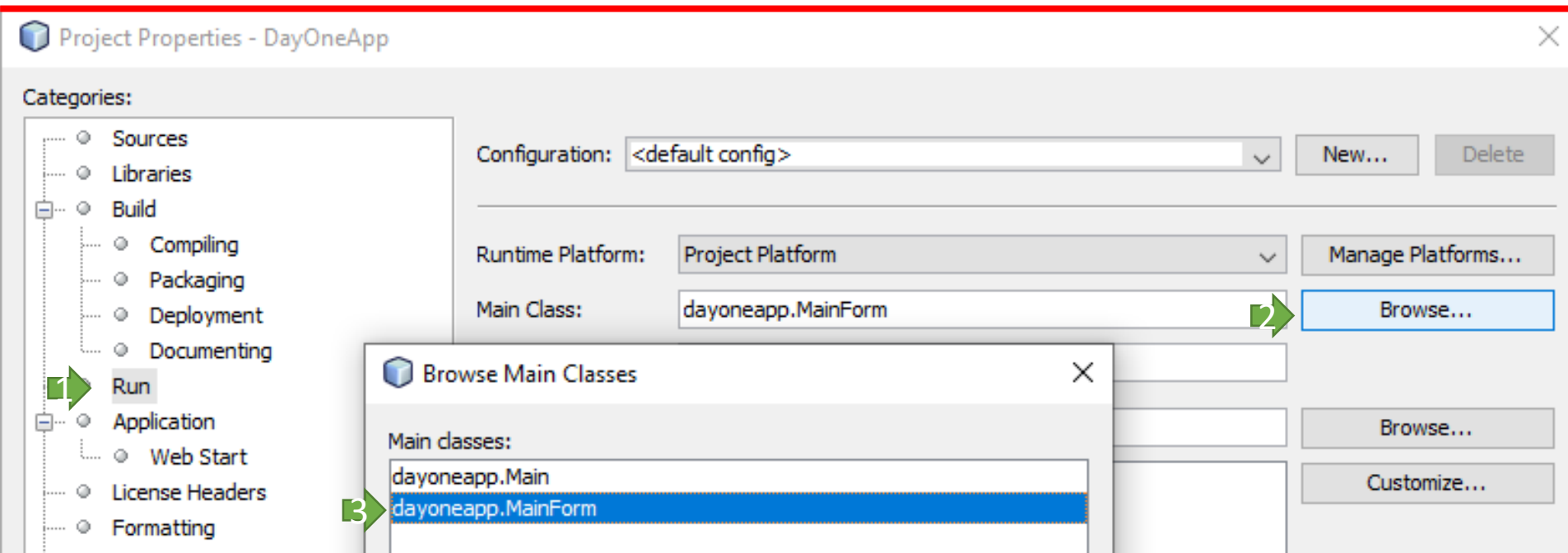
Cobalah mengganti title program melalui jendela properti





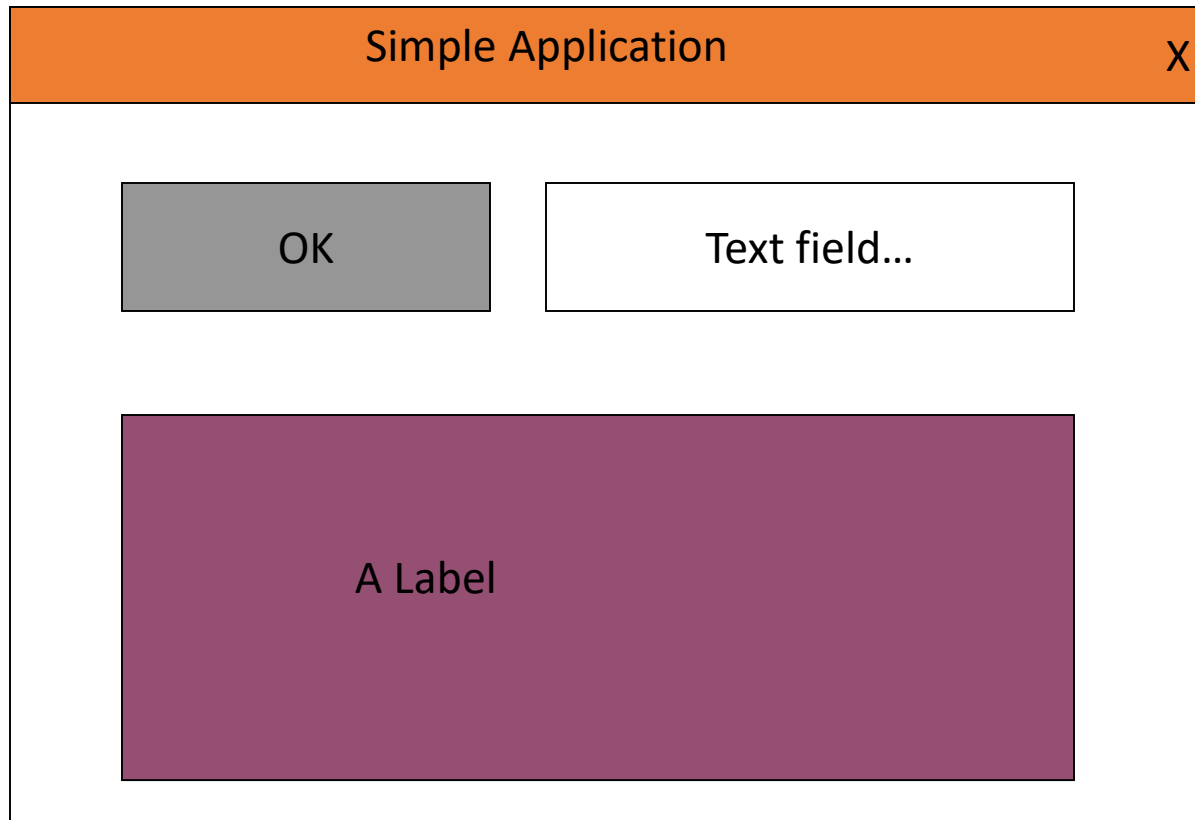
# How to Set Your Main Form as Main Class

1. Klik kanan pada project > “properties”
2. Pilih menu “Run” pada “categories”
3. Klik “Browse” pada “Main Class” > Pilih





# Building a GUI



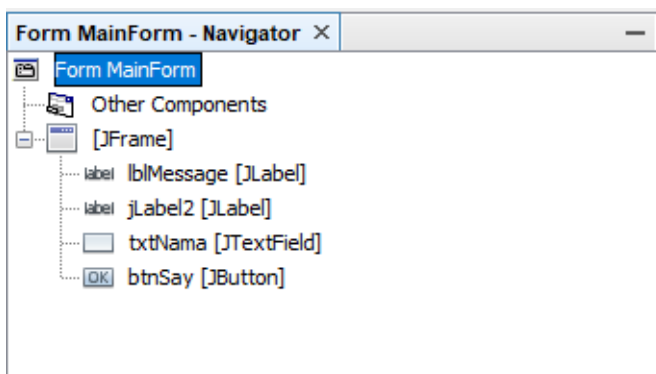
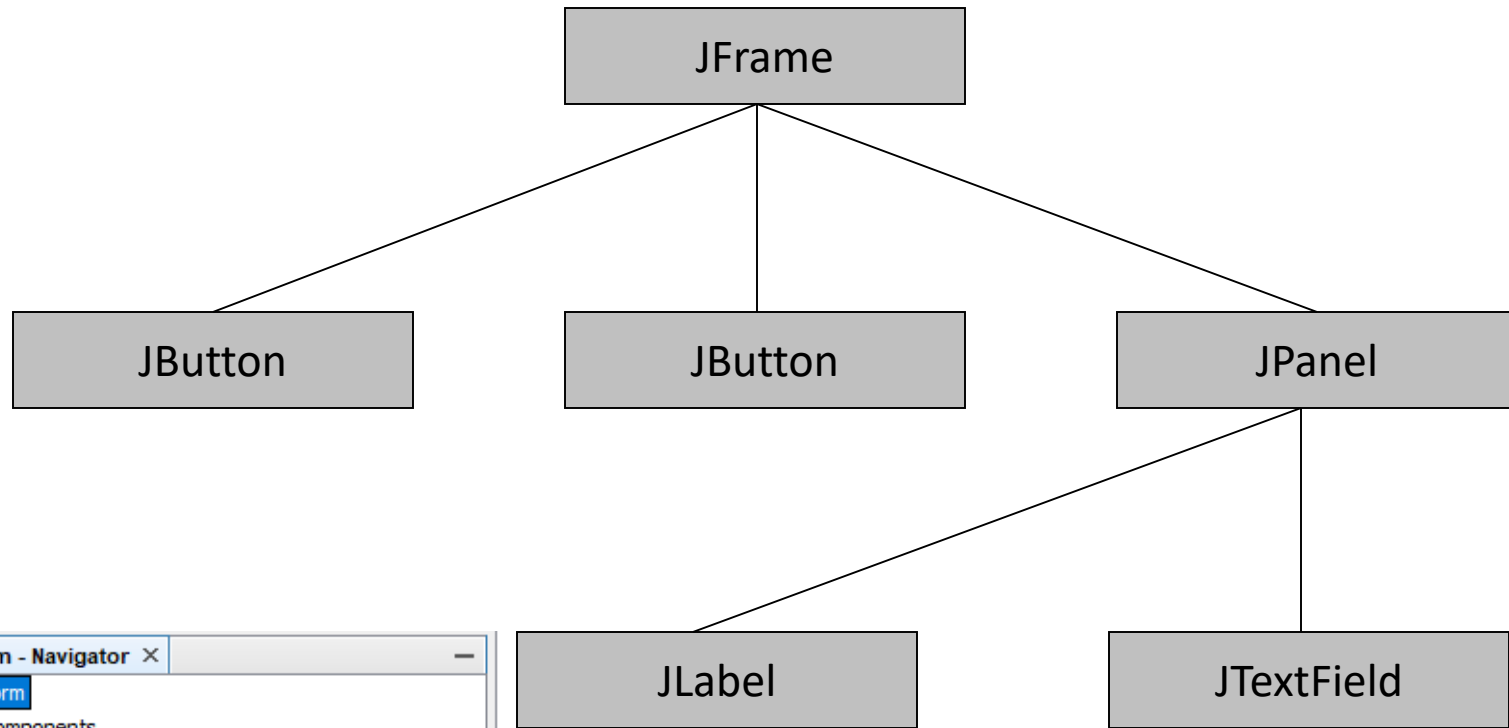


# The containment hierarchy

- This layered GUI can be viewed as a hierarchy of components
  - *NOT* an inheritance hierarchy,
  - It just describes how components are nested one within another



# The containment hierarchy







# Swing Top level containers

- JWindow
  - Basic no frills window, just a square on the screen
- JFrame
  - The basic Swing window. Offers basic window controls, resizable
- JDialog
  - For building dialog boxes, e.g. File open/save
- JApplet
  - For building applets, embedded into a web page



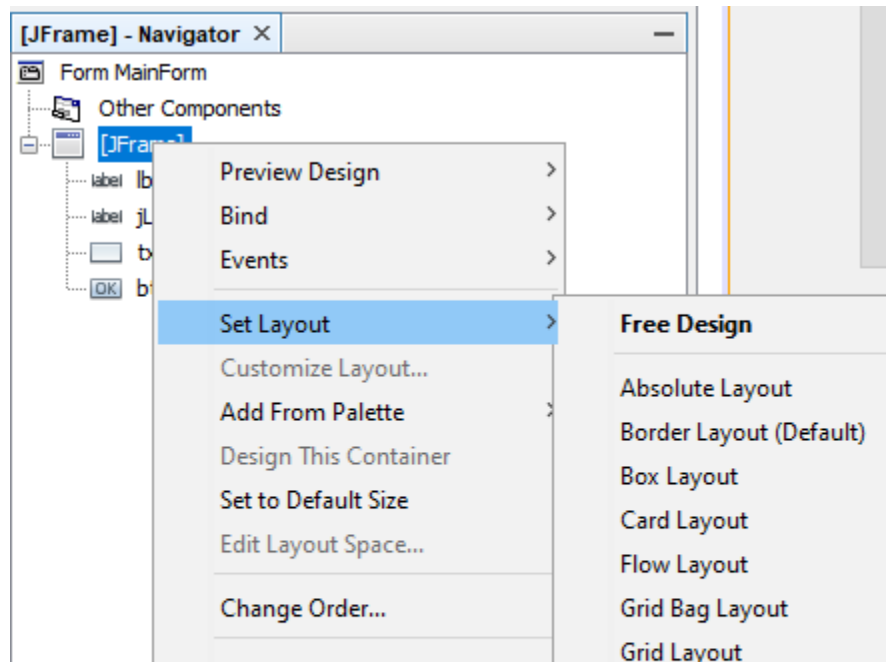
# Working with JFrames

- Many different possibilities, but the basics include:
  - Setting window title
  - Setting location on screen
  - Setting size of window
  - Restricting resizes
  - Set close operation (exit the program), as by default it does nothing.



# Layout Managers

- Responsible for layout out (arranging) components in a Container
- Several different types with different uses
- None of them provide for precise x-y alignment, unlike VB forms



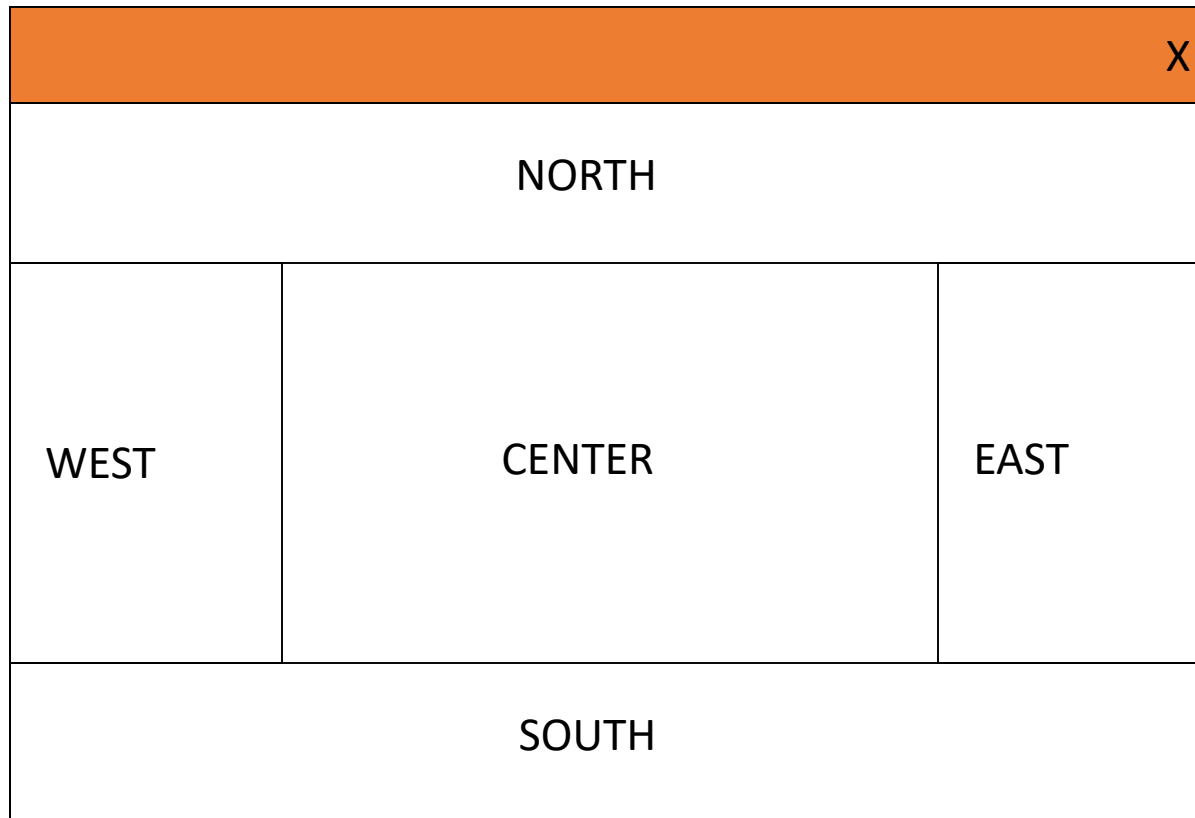


## Border Layout

- This is the default layout for JFrame
- Divides the content pane into 5 areas (north, south, east, west, center)
- Areas are expanded/contracted as needed, along with their contents.
  - Therefore ignores preferred size of the components.
- Center is the default if not specified.
- Adding two components to the same zone means they get added *one on top of the other*
  - Instead add the components to a JPanel, and then add that instead.



# Border Layout





## Grid Layout

- Divides the container into a rectangular grid
  - Configurable number rows/columns
- Each grid location is of equal size, one component assigned to each.
- Automatically assigns components to next available location



## Other layout managers

- Flow Layout (default for `JPanel`)
  - Arranges components left-to-right
  - Used to arrange buttons on a panel
- Card Layout
  - Arranges components like a deck of cards
  - Only one card visible at a time
- Box Layout, Grid Bag Layout
  - Very sophisticated managers, used by GUI builders for very precise GUI designs.
  - Not recommended for hand use!



# Menus

- A JFrame can have only a single menu bar
  - Instance of the JMenu object
- A menu bar can have several menus on it
  - Instances of the JMenu object
- A menu can have several items on it
  - Instances of the JMenuItem object
- Example





# Event Handler

1. Buatlah UI seperti dibawah ini
2. Sesuaikan “variable name” masing-masing komponen

Nama

Say!

Hello World!

txtNama

btnSay

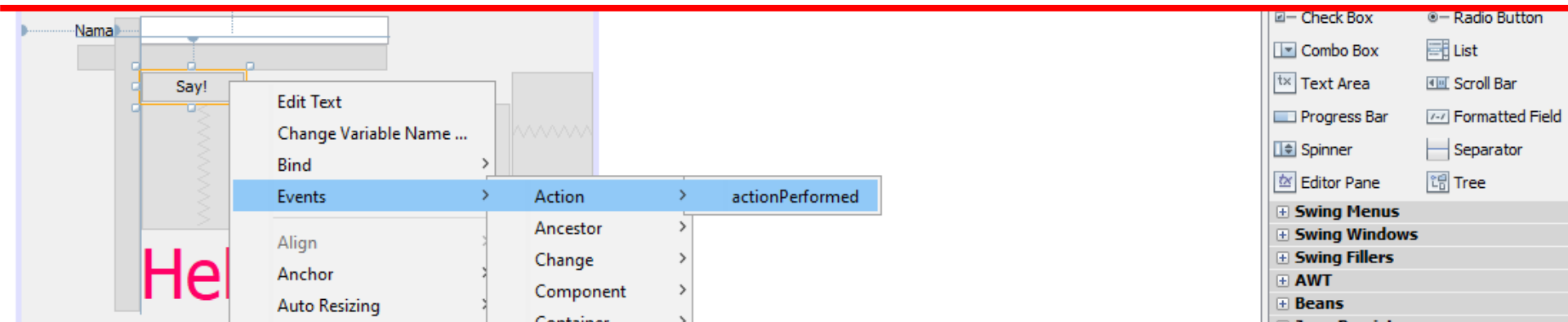
lblMessage



# Event Handler

Klik kanan pada komponen button

Pilih “Events” > “Action” > “actionPerformed”



Tuliskan kode berikut

```
83 private void btnSayActionPerformed(java.awt.event.ActionEvent evt) {  
84     // TODO add your handling code here:  
85     lblMessage.setText("Hello " + txtNama.getText());  
86 }
```



# Change Look and Feel

Berikut daftar “Look and Feel” yang dapat dipilih

1. Metal
2. Nimbus
3. Windows
4. CDE/Motif
5. Windows Classic

```
91 public static void main(String args[]) {  
92     try {  
93         for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.  
94             if ("Nimbus".equals(info.getName())) {  
95                 javax.swing.UIManager.setLookAndFeel(info.getClassName());  
96                 break;  
97             }  
98     }  
}
```



## Deploy

- Klik kanan project
- Pilih “Build” atau “Clean and Build”
- Lokasi file binary di <project\_dir>/dist/
- Berikut ini perintah untuk menjalankan program melalui terminal
  - `java -jar <jar_name>.jar`



# Latihan

Task 2 App

Nama: Royko

Hobi:

- ☒ Makan
- ☒ Nonton
- ☐ Shopping

Jenis Kelamin:

- ☒ Laki-laki
- ☐ Perempuan

Jurusan:

- Teknik Informatika
- Manajemen Informatika
- Sistem Informasi
- Desain Komunikasi Visual

Submit!

Message

Nama kamu Royko  
Kamu adalah seorang Lelaki  
Hobi kamu Makan Nonton  
Saat ini kamu sedang kuliah di jurusan Desain Komunikasi Visual

OK



# Latihan

Buatlah GUI seperti dibawah ini

The screenshot shows a GUI form with the following elements:

- Nama:** A text input field.
- Hobi:** Three checkboxes labeled "Makan", "Nonton", and "Shopping".
- Jenis Kelamin:** Two radio buttons labeled "Laki-laki" and "Perempuan".
- Jurusan:** A list box containing the following items: "Teknik Informatika", "Manajemen Informatika", "Sistem Informasi", and "Desain Komunikasi Visual".
- Submit!:** A button at the bottom right.

Sesuaikan properti variable name dan text

## Komponen

CheckBox1  
CheckBox2  
CheckBox3  
Button 1  
TextBox1  
RadioButton1  
RadioButton2  
ListBox1  
ButtonGroup

## Variable Name

cbHob1  
cbHob2  
cbHob3  
btnSubmit  
txtNama  
rdLaki  
rdPerempuan  
lstJurusan  
btgGroupJK

## Text

Makan  
Nonton  
Shopping  
Submit!  
(kosong)  
Laki-Laki  
Perempuan  
(tidak diubah)  
(tidak ada)



# Latihan

```
147 private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {  
148     // TODO add your handling code here:  
149     String nama = "";  
150     String hobi = "";  
151     String jk = "";  
152     String jurusan = "";  
153     String info = "";  
154  
155     nama = txtNama.getText();  
156     if(cbHobi1.isSelected())  
157         hobi+= " Makan";  
158     if(cbHobi2.isSelected())  
159         hobi+= " Nonton";  
160     if(cbHobi3.isSelected())  
161         hobi+= " Shopping";  
162  
163     if(rdLaki.isSelected()){  
164         jk += "Lelaki";  
165     } else {  
166         jk += "Perempuan";  
167     }  
168  
169     jurusan = lstJurusan.getSelectedValue();  
170  
171     info = "Nama kamu " + nama;  
172     info += "\nKamu adalah seorang " + jk;  
173     info += "\nHobi kamu " + hobi;  
174     info += "\nSaat ini kamu sedang kuliah di jurusan " + jurusan;  
175  
176     JOptionPane.showMessageDialog(null, info);  
177 }
```



# Latihan

Buatlah program konversi suhu

Input

From

☐ Celcius

☐ Fahrenheit

☐ Calvin

To

☐ Celcius

☐ Fahrenheit

☐ Calvin

Calculate!

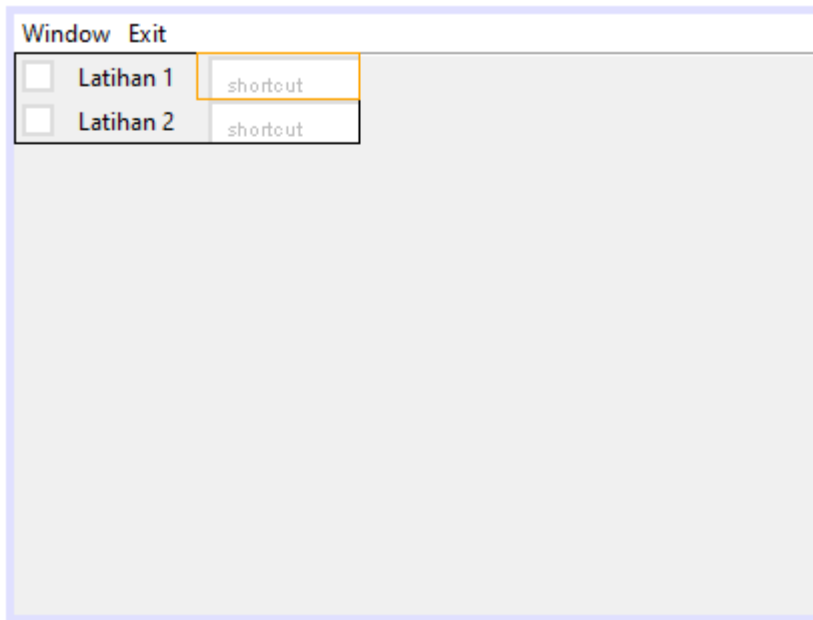
Result





# Multi Window

Menampilkan frame lain dari frame utama.



```
new Suhu().setVisible(true);
```

Set defaultCloseOperation to dispose