Object-based Programming

Week 9 – Graphical User Interface with Java Swing

Java Library for UI

Abstract Window Toolkit (AWT)

Swing

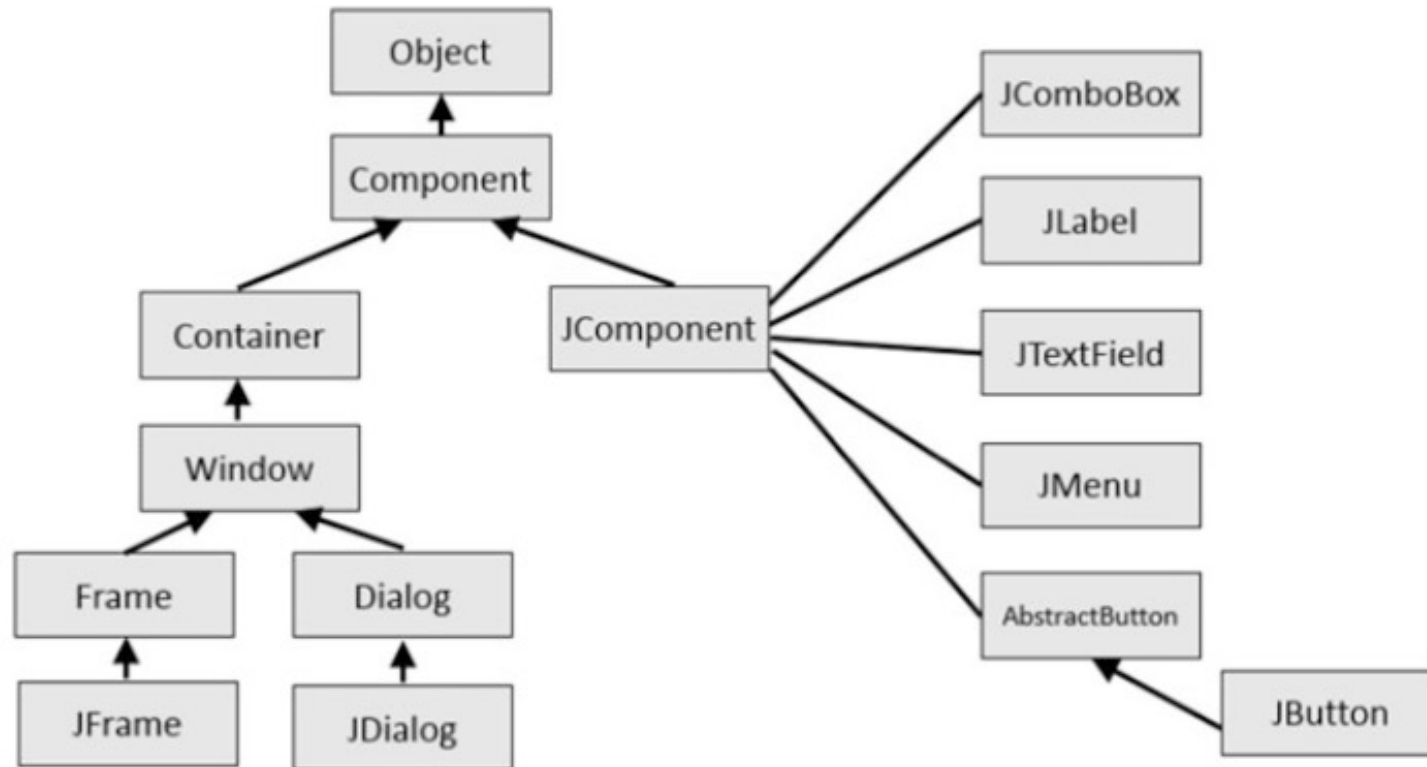JavaFX

Partial Class Hierarchy (Swing)
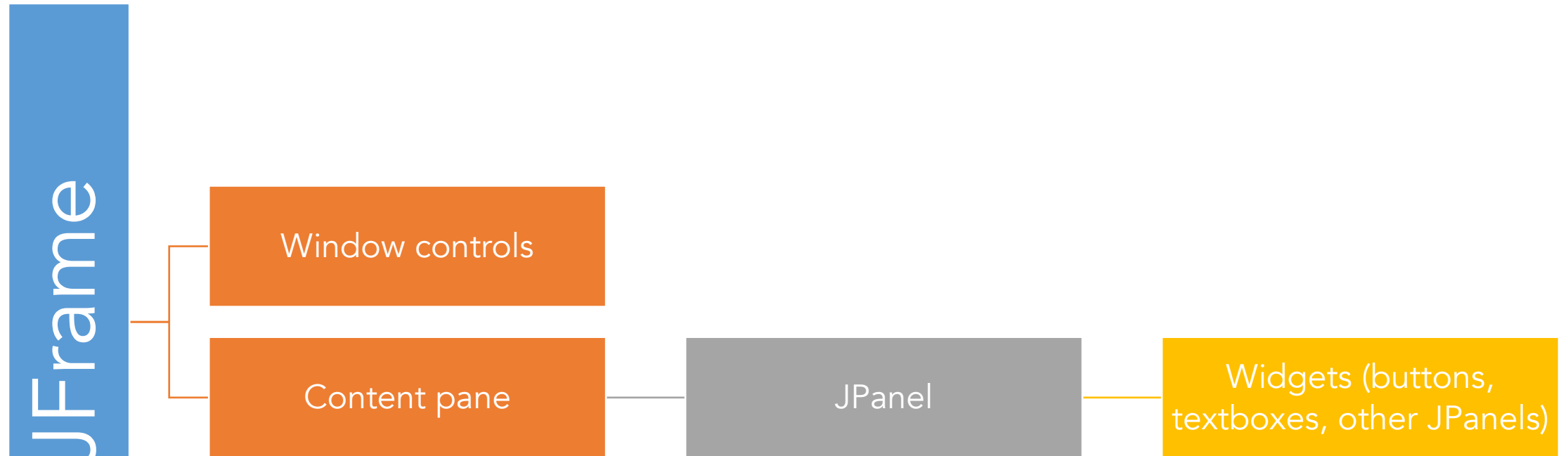
Taxonomy of a GUI

First example

```java
import javax.swing.*;

public class Example extends JFrame {
    public Example() {
        setTitle("Example App");

        setSize(600, 400);

        setVisible(true);
    }

    public static void main(String[] args) {
        Example app = new Example();
    }
}
```


Example App

Default Close Operation

```java
import javax.swing.*;

public class Example extends JFrame {
    public Example() {
        setTitle("Example App");
        setSize(600, 400);

        setDefaultCloseOperation(EXIT_ON_CLOSE);

        setVisible(true);
    }

    public static void main(String[] args) {
        Example app = new Example();
    }
}
```

Opsi:

- EXIT_ON_CLOSE

- DISPOSE_ON_CLOSE

- HIDE_ON_CLOSE

- DO_NOTHING_ON_CLOSE

Adding Components

```java
import javax.swing.*;

public class Example extends JFrame {
    public Example() {
        setTitle("Example App");
        setSize(600, 400);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        JLabel label = new JLabel("Hello World!");
        JButton button = new JButton("Help");

        JPanel panel = new JPanel();
        panel.add(label);
        panel.add(button);

        add(panel);

        setVisible(true);
    }
}
```
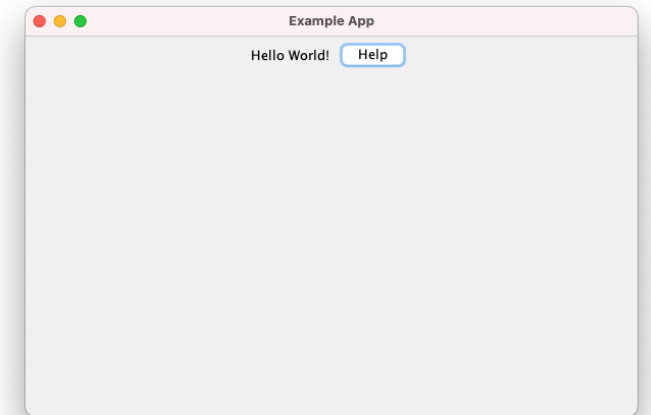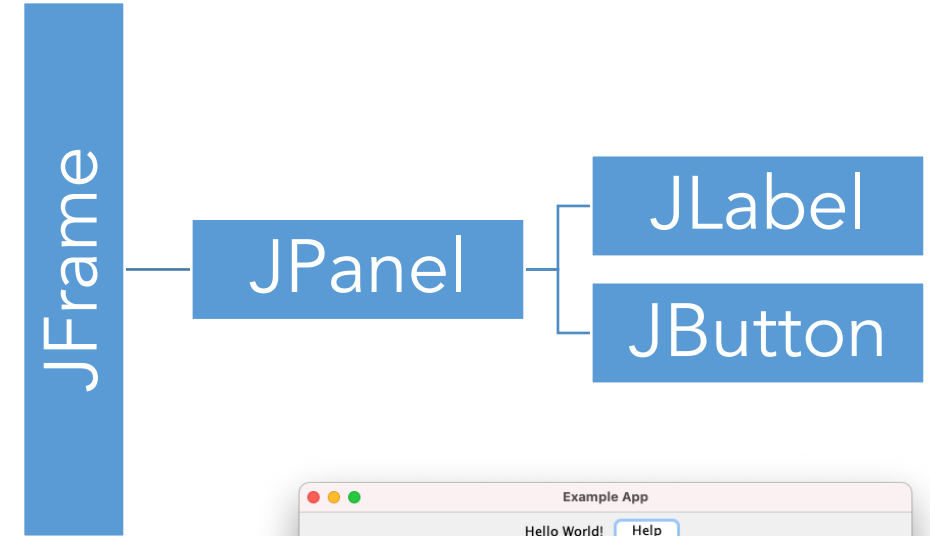
JFrame — JPanel — JLabel / JButton

Example App

Hello World! Help

... Button clicked
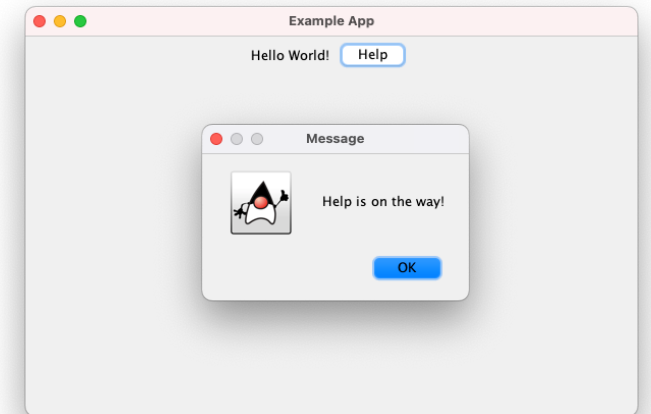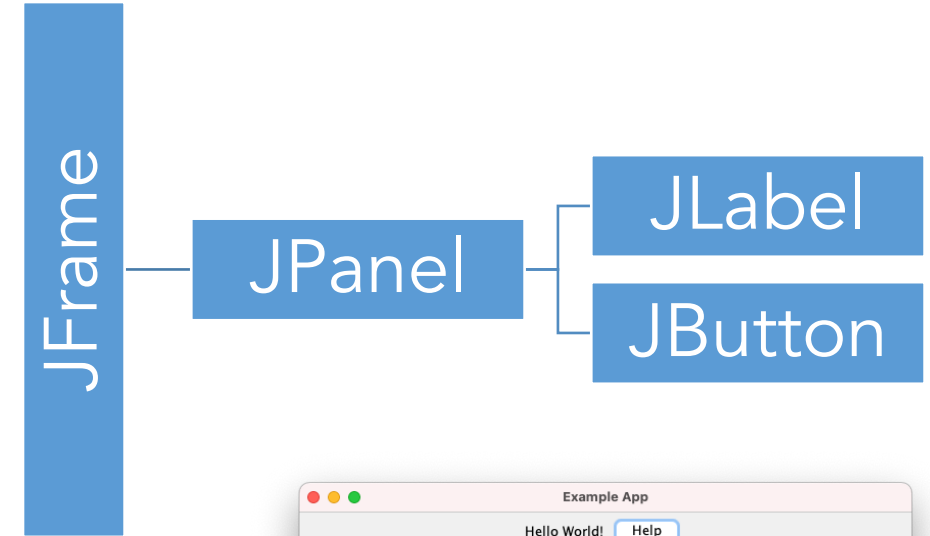
```java
JLabel label = new JLabel("Hello World!");
JButton button = new JButton("Help");

button.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null,
                                "Help is on the way!");
    }
});

...
```

```java
import javax.swing.*;
public class Example extends JFrame implements ActionListener {
    public Example() {
        ...
        JButton button = new JButton("Help");
        button.addActionListener(this);
        ...
    }


    @Override
    public void actionPerformed(ActionEvent e) {
        JOptionPane.showMessageDialog(null,
                                "Help is on the way!");
    }
}
```
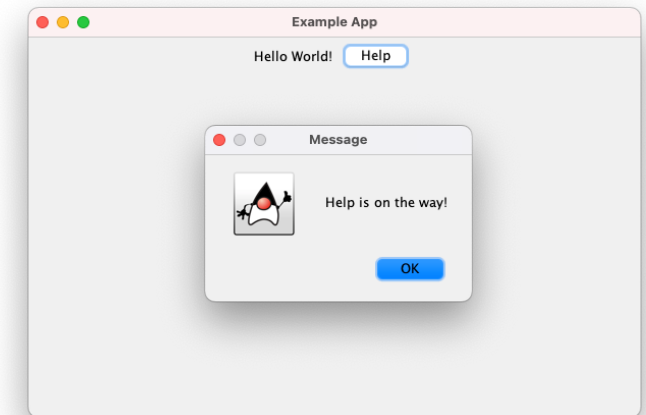
Model-View-Controller (MVC)

Design Pattern

Design Pattern

A template for solving a recurring problem

Neither a code solution, nor tied to any particular language

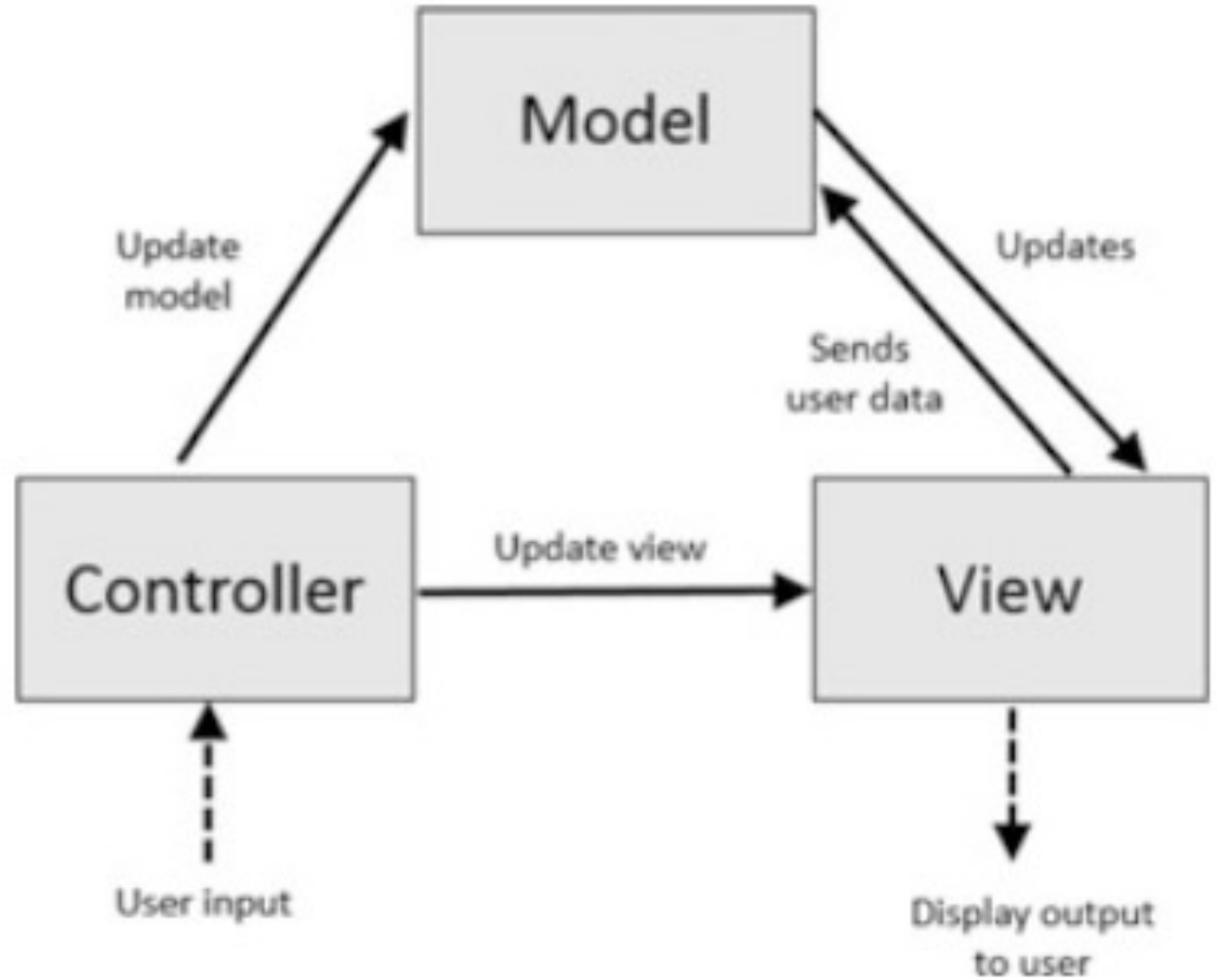Represents an abstract strategy or set of principles

MVC Pattern

MVC Pattern

- Model
  - Underlying application or business logic that is concerned solely with how the application operatis

- View
  - The physical presentation, i.e. the appearance of the user interface

- Controller
  - Concerned with user interaction

Simple Example

- Counter app
- Two buttons => (1) increase; (2) decrease
- A number is displayed on the GUI, starting with an initial value of 0

```java
public class Counter {
    private int value;

    public Counter() {
        value = 0;
    }

    public String toString() {
        return value + "";
    }

    public void increase() {
        value++;
    }

    public void decrease() {
        value--;
    }

    public void reset() {
        value = 0;
    }
}
```

```java
import javax.swing.*;
import java.awt.event.*;

public class Example extends JFrame {
    private Counter counter;

    public Example() {
        counter = new Counter();

        setTitle("Counter App");
        setSize(220, 100);
        setDefaultCloseOperation(EXIT_ON_CLOSE);

        // create swing components

        // add action listeners on the buttons

        // create a panel and add to JFrame

        setVisible(true);
    }

    public static void main(String[] args) {
        new Example();
    }
}
```
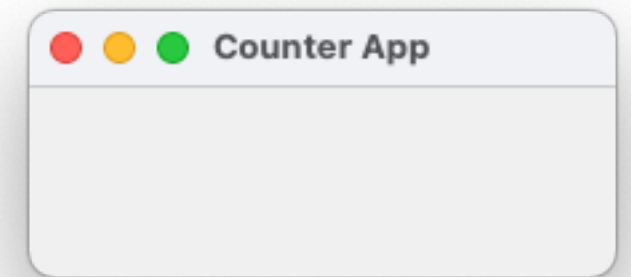
```java
// create swing components
JButton up = new JButton("+");

JButton down = new JButton("-");

JLabel label = new JLabel(counter.toString());


// add action listeners on the buttons


// create a panel and add to JFrame
JPanel panel = new JPanel();

panel.add(down);

panel.add(label);

panel.add(up);


add(panel);
```
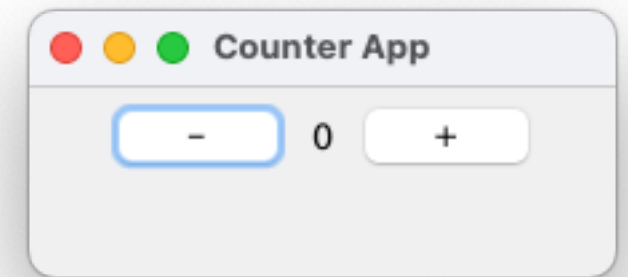
```java
// add action listeners on the buttons
up.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        counter.increase();

        label.setText(counter.toString());

    }

});


down.addActionListener(new ActionListener() {

    @Override

    public void actionPerformed(ActionEvent e) {

        counter.decrease();

        label.setText(counter.toString());

    }

});
```
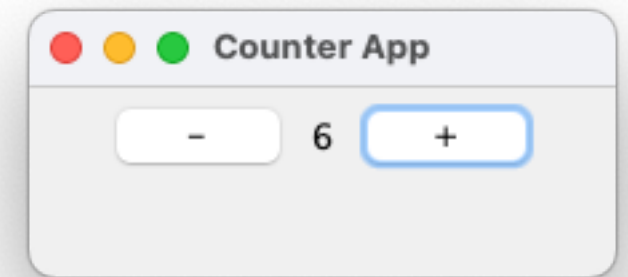
Komponen Swing

- JTextField
- JCheckBox
- JRadioButton
- JToggleButton dan ButtonGroup

...

Latihan

Buat program dengan tampilan ini
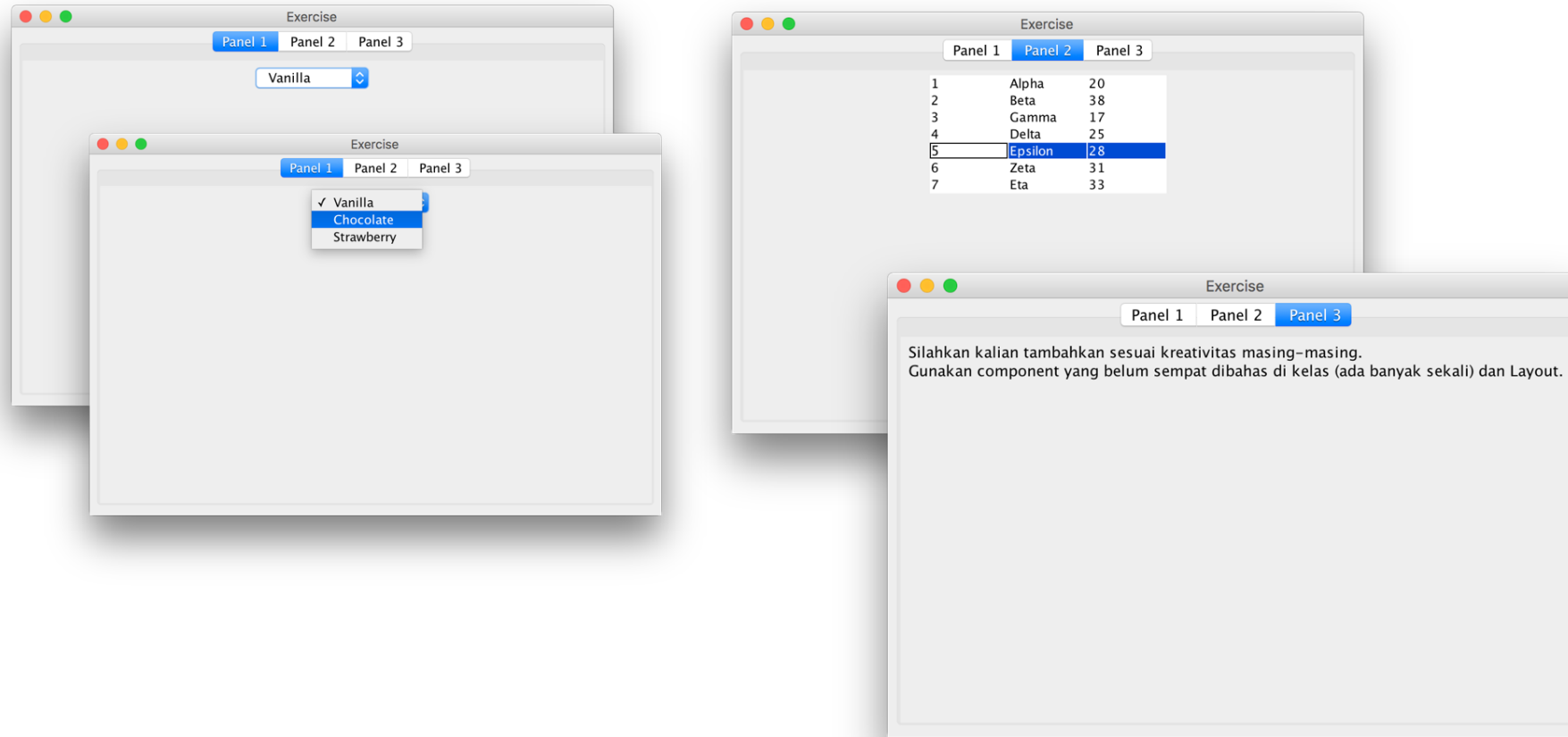
Komponen yang digunakan:
- JTabbedPane (3 tabs)
- JComboBox
- JTable
- dan lain-lain...

Exercise — Panel 1 | Panel 2 | Panel 3
Vanilla

Exercise — Panel 1 | Panel 2 | Panel 3
✓ Vanilla
Chocolate
Strawberry

Exercise — Panel 1 | Panel 2 | Panel 3

1	Alpha	20
2	Beta	38
3	Gamma	17
4	Delta	25
5	Epsilon	28
6	Zeta	31
7	Eta	33

Exercise — Panel 1 | Panel 2 | Panel 3

Silahkan kalian tambahkan sesuai kreativitas masing-masing.
Gunakan component yang belum sempat dibahas di kelas (ada banyak sekali) dan Layout.