Zid’Avwa Al Bari’i

244107020083 / TI – 2I / 26

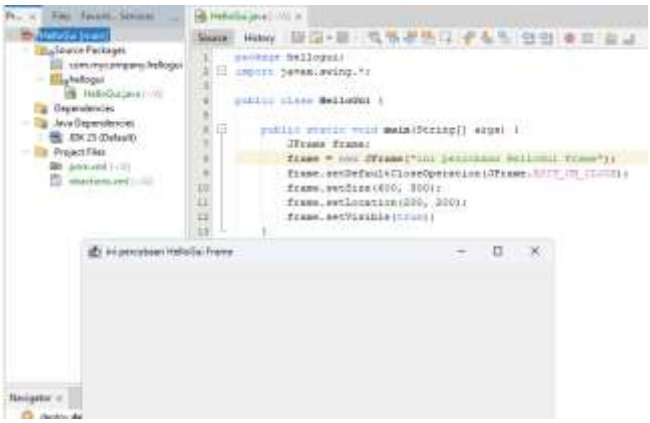
All the code here is on GitHub

https://github.com/ZidAvwa/CollegeStudy/tree/main/3rdSemester/13.%20Jobsheet13

Jobsheet 13

Experiment 1

Code



Experiment 2

Code

Question

Modify the program code by adding a new JButton to perform the addition calculation function, so that when the button is clicked (event click) it will display the addition results of the values A and B

1. Declared private JButton addButton; at the top.
2. Inside createButton(): Added code to create the "Addition" button and a new listener (SumListener) that performs int c = a + b;.
3. Inside createPanel(): Added panel.add(addButton); so the button actually shows up on the screen.

Experiment 3

Code

```java
package ManajemenLayout;
import java.awt.BorderLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Border extends JFrame{
    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JPanel panel;

    public Border() {
        panel = new JPanel();
        panel.setLayout(new BorderLayout());

        panel.add(new JButton("1"), BorderLayout.NORTH);
        panel.add(new JButton("2"), BorderLayout.WEST);
        panel.add(new JButton("3"), BorderLayout.CENTER);
        panel.add(new JButton("4"), BorderLayout.EAST);
        panel.add(new JButton("5"), BorderLayout.SOUTH);

        add(panel);
        setSize(FRAME_WIDTH, FRAME_HEIGHT);
        setTitle("Border Layout Demo");
    }
}
```

```java
package ManajemenLayout;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Box extends JFrame {
    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JPanel panel;

    public Box() {
        panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.add(new JButton("1"));
        panel.add(new JButton("2"));
        panel.add(new JButton("3"));
        panel.add(new JButton("4"));
        panel.add(new JButton("5"));
        add(panel);
        setSize(FRAME_WIDTH, FRAME_HEIGHT);
        setTitle("Box Layout Demo");
    }
}
```

```java
package ManajemenLayout;
import java.awt.GridLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class Grid extends JFrame {
    private static final int FRAME_WIDTH = 600;
    private static final int FRAME_HEIGHT = 200;
    private JPanel panel;

    public Grid() {
        panel = new JPanel();
        panel.setLayout(new GridLayout(4, 2));
        panel.add(new JButton("1"));
        panel.add(new JButton("2"));
        panel.add(new JButton("3"));
        panel.add(new JButton("4"));
        panel.add(new JButton("5"));
        panel.add(new JButton("6"));
        panel.add(new JButton("7"));
        panel.add(new JButton("8"));

        add(panel);
        setSize(FRAME_WIDTH, FRAME_HEIGHT);
        setTitle("Grid Layout Demo");
    }
}
```

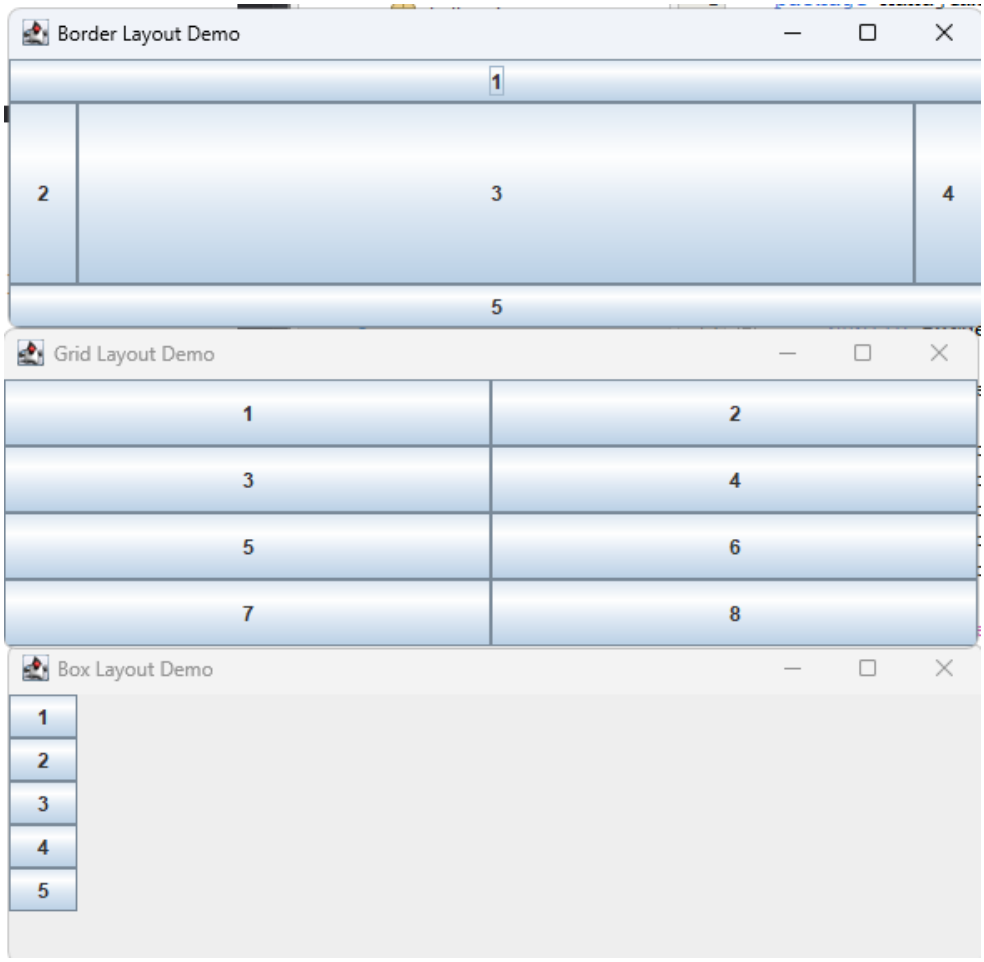
```java
package ManajemenLayout;

import javax.swing.JFrame;

public class LayoutGUI {
    public static void main(String debby[]) {
        JFrame frame = new Border();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

        JFrame frame2 = new Grid();
        frame2.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame2.setVisible(true);

        JFrame frame3 = new Box();
        frame3.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame3.setVisible(true);
    }
}
```



Question

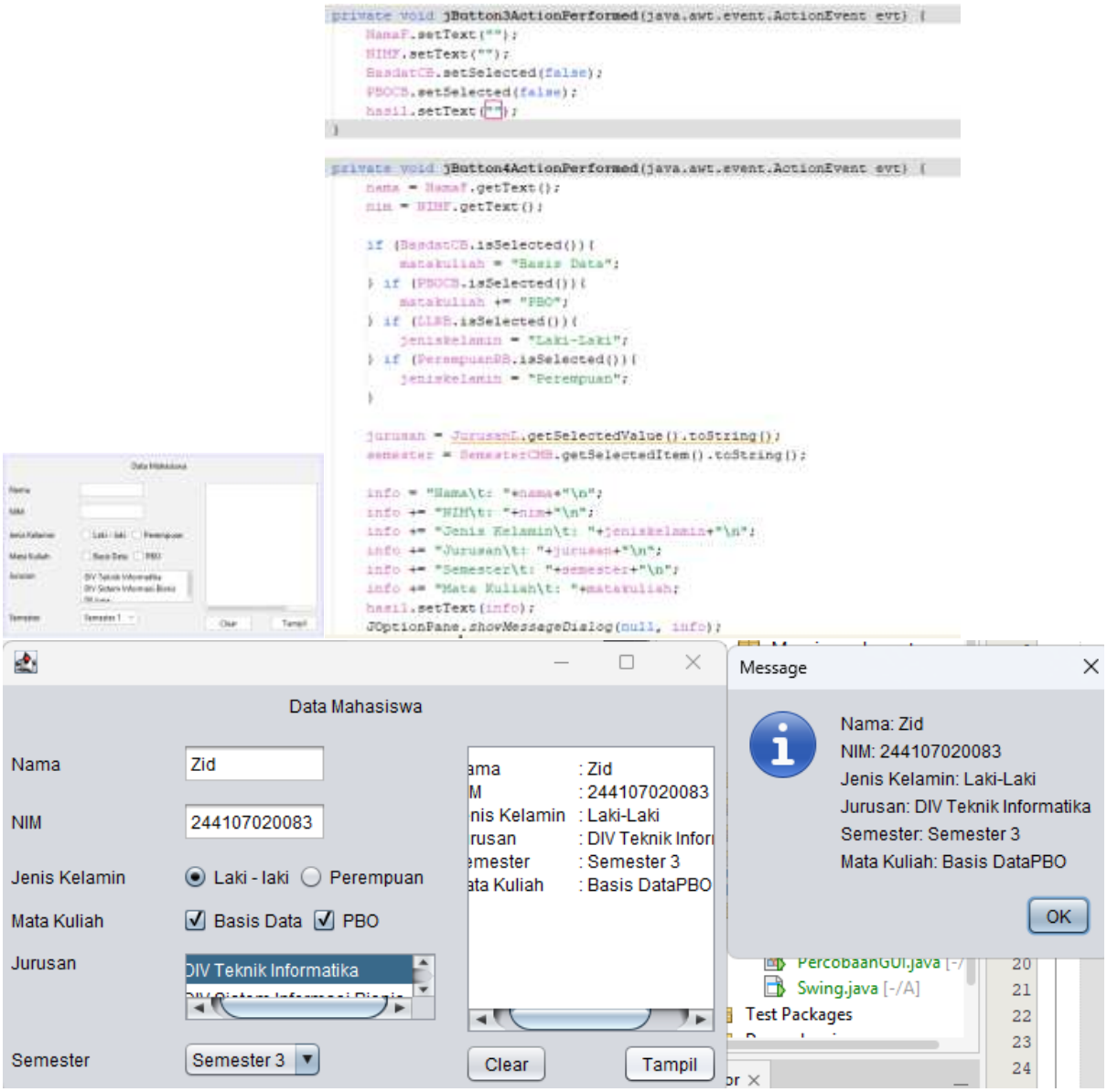
a) What is the difference between Grid Layout, Box Layout and Border Layout?

- BorderLayout: Divides the window into 5 fixed areas (North, South, East, West, Center).
- GridLayout: Arranges components in a grid of equal-sized cells.
- BoxLayout: Stacks components in a single straight line (either vertical or horizontal).

b) What is the function of each of the following codes?
These lines create the window object for that specific layout, ensure the program actually stops running when you close the window, and make the window visible to the user.

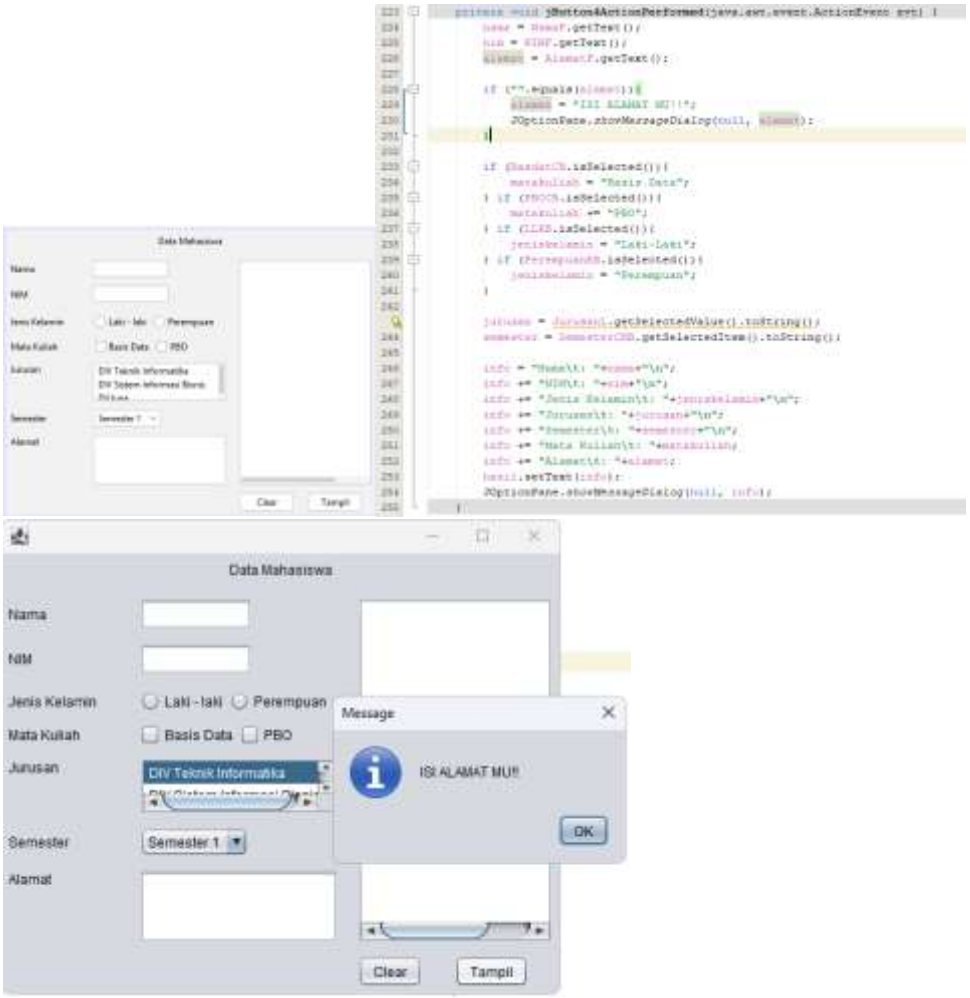
Experiment 4

Code



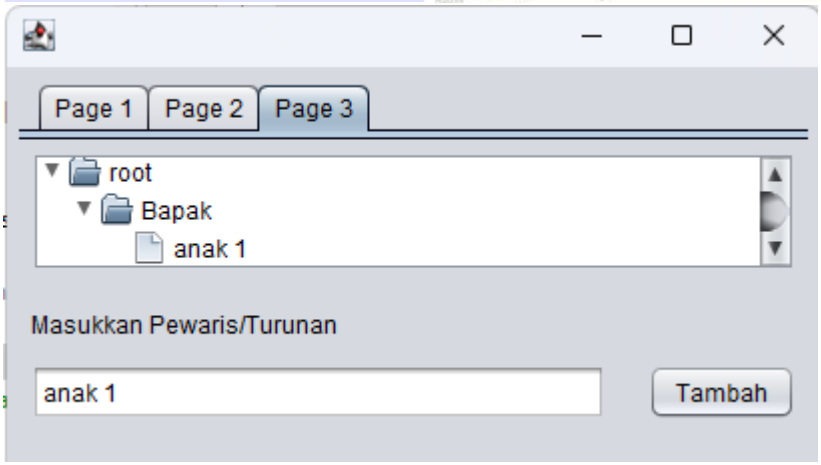
Questions

1. It ensures the application starts safely and correctly.
2. Because the radio buttons and checkbox have 2 state true and false
3. Just add another text field (also declare variables and add in the clear button) and if logic like this



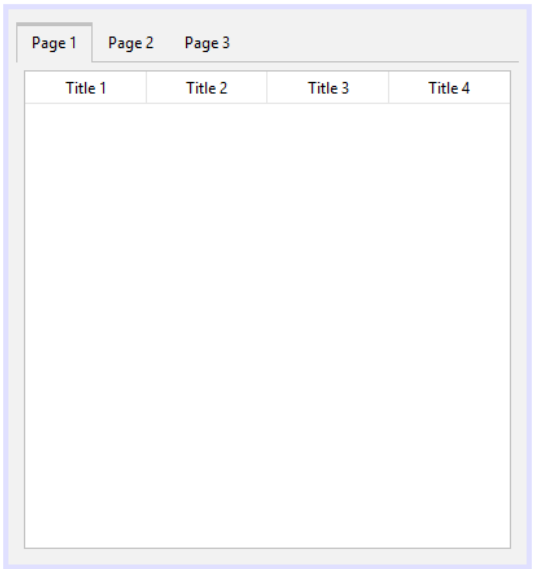
Experiment 5

Code

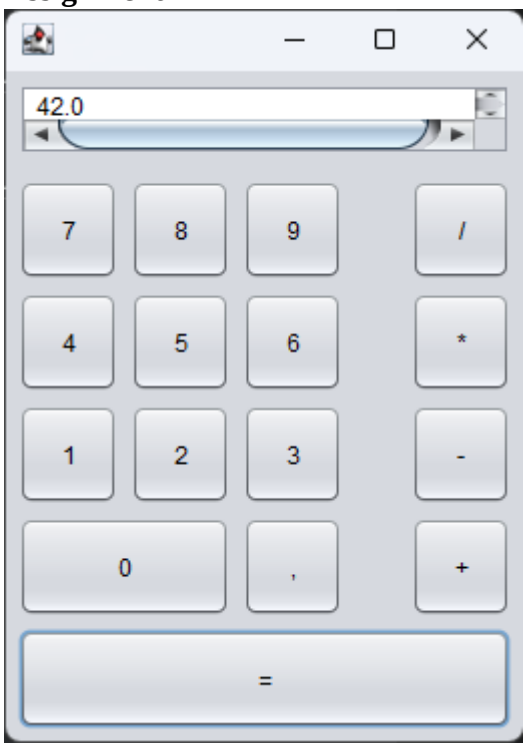


Questions

1. To make tabs
2. Just add the table like this



Assignment



the program is on github https://github.com/ZidAvwa/CollegeStudy/tree/main/3rdSemester/13.%20Jobsheet13/Assignment