



UMS
UNIVERSITI MALAYSIA SABAH

KK14203 OBJECT ORIENTED PROGRAMMING
SEMESTER II
SESSION 2019/2020

ASSIGNMENT 2
INDIVIDUAL

Mini Calculator

LECTURER: Madam Siti Hasnah Tanalol

NAME	MATRIC NUMBER
Shahmir Azwan Bin Sepuan	BI19160323

Table of Contents

INTRODUCTION.....	3
OBJECTIVES	4
JAVA CODE	5
OBJECT ORIENTED CONCEPT IMPLEMENTATION	10
READ AND WRITE IMPLEMENTATION.....	13
USER MANUAL	14
CONCLUSION.....	16

INTRODUCTION

Calculator is a device that performs arithmetic operation on numbers. In this project I build a simple calculators that can do only addition, subtraction, multiplication, and division. However, this project can only do operation for two numbers. I choose to make this kind of project because to improve my understanding the basic function of GUI and to understand object oriented programming.

The main of this project is to provide better user friendly calculator. It is very easy to use with a simple layout to make user didn't feel stress while using the calculator.

OBJECTIVES

1. to perform a number of calculations in response to user supplied input.
2. To make sure that all keys are correctly performing operation on the screen.
3. The databases used by calculators may be very simple.

JAVA CODE

```
//Name: Shahmir Azwan Bin Sepuan
//Matric No: BI19160323
public class BI19160323 extends javax.swing.JFrame {

    /**
     * Creates new form BI19110130
     */
    public BI19160323() {
        initComponents();
        setResizable(false);
        setSize(535, 438);
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jButton4 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jButton5 = new javax.swing.JButton();
        jTextField1 = new javax.swing.JTextField();
        jTextField2 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();
        jumlah = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jPanel1 = new javax.swing.JPanel();

        jButton4.setText("jButton4");

        jLabel4.setText("jLabel4");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(null);

        jLabel1.setText("NOMBOR PERTAMA");
        getContentPane().add(jLabel1);
        jLabel1.setBounds(57, 118, 191, 20);
```

```

jLabel2.setText("NOMBOR KEDUA");
getContentPane().add(jLabel2);
jLabel2.setBounds(57, 192, 167, 20);

jButton1.setBackground(new java.awt.Color(51, 102, 255));
jButton1.setText("Tambah");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
getContentPane().add(jButton1);
jButton1.setBounds(82, 335, 91, 29);

jButton2.setBackground(new java.awt.Color(51, 102, 255));
jButton2.setText("Tolak");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
getContentPane().add(jButton2);
jButton2.setBounds(191, 335, 71, 29);

jButton3.setBackground(new java.awt.Color(51, 102, 255));
jButton3.setText("Darab");
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
getContentPane().add(jButton3);
jButton3.setBounds(280, 335, 75, 29);

jButton5.setBackground(new java.awt.Color(51, 102, 255));
jButton5.setText("Bahagi");
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});
getContentPane().add(jButton5);
jButton5.setBounds(373, 335, 79, 29);

jTextField1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1ActionPerformed(evt);
    }
});

```

```

getContentPane().add(jTextField1);
jTextField1.setBounds(257, 115, 202, 26);

jTextField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField2ActionPerformed(evt);
    }
});
getContentPane().add(jTextField2);
jTextField2.setBounds(257, 189, 202, 26);

jLabel3.setText("JUMLAH");
getContentPane().add(jLabel3);
jLabel3.setBounds(57, 253, 86, 20);
getContentPane().add(jumlah);
jumlah.setBounds(257, 248, 202, 25);

jLabel5.setBackground(new java.awt.Color(255, 51, 51));
jLabel5.setText("MINI CALCULATOR");
getContentPane().add(jLabel5);
jLabel5.setBounds(173, 29, 170, 20);

jLabel6.setText("Made by Shahmir Azwan");
getContentPane().add(jLabel6);
jLabel6.setBounds(335, 418, 200, 20);

jPanel1.setBackground(new java.awt.Color(51, 153, 255));

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 540, Short.MAX_VALUE)
);
jPanel1Layout.setVerticalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 440, Short.MAX_VALUE)
);

getContentPane().add(jPanel1);
jPanel1.setBounds(0, 0, 540, 440);

pack();
} // </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

```

```

        int num1= Integer.parseInt(jTextField1.getText());
        int num2= Integer.parseInt(jTextField2.getText());
        int result=num1+num2;
        jumlah.setText("Hasil tambah "+num1+ " dan "+num2+" ialah "+result);

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

        int num1= Integer.parseInt(jTextField1.getText());
        int num2= Integer.parseInt(jTextField2.getText());
        int result=num1-num2;
        jumlah.setText("Hasil tolak "+num1+ " dan "+num2+" ialah "+result);
        // TODO add your handling code here:
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

        int num1= Integer.parseInt(jTextField1.getText());
        int num2= Integer.parseInt(jTextField2.getText());
        int result=num1*num2;
        jumlah.setText("Hasil darab "+num1+ " dan "+num2+" ialah "+result);
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

        int num1= Integer.parseInt(jTextField1.getText());
        int num2= Integer.parseInt(jTextField2.getText());
        int result=num1/num2;
        jumlah.setText("Hasil bahagi "+num1+ " dan "+num2+" ialah "+result);
    }

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
                javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

```



```

        javax.swing.UIManager.setLookAndFeel(info.getClassName());
        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(BI19160323.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(BI19160323.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(BI19160323.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(BI19160323.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
//</editor-fold>
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new BI19160323().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField2;
private javax.swing.JLabel jumlah;
// End of variables declaration
}

```

OBJECT ORIENTED CONCEPT IMPLEMENTATION

1. Encapsulation

Encapsulation allows us to protect the data stored in a class from system-wide access. As its name suggests, it safeguards the internal contents of a class like a real-life capsule.

1. public void actionPerformed

```
public void actionPerformed(java.awt.event.ActionEvent evt) {  
    jButton2ActionPerformed(evt);  
}  
});  
getContentPane().add(jButton2);  
jButton2.setBounds(191, 335, 71, 29);  
  
jButton3.setBackground(new java.awt.Color(51, 102, 255));  
jButton3.setText("Darab");  
jButton3.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton3ActionPerformed(evt);  
    }  
});
```

2. public static void main (string...s)

```
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    Look and feel setting code (optional)  
    //</editor-fold>  
  
    /* Create and display the form */  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new BI19160323().setVisible(true);  
        }  
    });  
}
```

In above they all the data fields are private. which cannot be accessed directly. These fields can be accessed via public methods only. Fields are made hidden data fields using encapsulation technique of OOPs.

2. Object & Classes

A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

1. class BI19160323

```
public class BI19160323 extends javax.swing.JFrame {  
    /**  
     * Creates new form BI19110130  
     */  
    public BI19160323() {  
        initComponents();  
        setResizable(false);  
        setSize(535, 438);  
    }  
}
```

3. Interface

Inside any implementation class, cannot change the variables declared in interface because by default, they are public, static and final. If there are two or more same methods in two interfaces and a class implements both interfaces, implementation of the method once is enough.

1. Implements ActionListener

```
jButton1.setBackground(new java.awt.Color(51, 102, 255));  
jButton1.setText("Tambah");  
jButton1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jButton1ActionPerformed(evt);  
    }  
})
```

4. Inner class

In the above program, there are two nested classes. Processor and RAM inside the outer class CPU. We can declare the inner class as protected. Hence, we have declared the RAM class as protected. Inside the Main class, we first created an instance of an outer class basiccalculator named basiccalculator. Using the instance of the outer class, then created objects of inner classes. use the dot (.) operator to create an instance of the inner class using the outer class.

1. new BI19160323

```
public void run() {  
    new BI19160323().setVisible(true);  
}
```

5. Abstraction

Abstraction means using simple things to represent complexity. In Java, abstraction means simple things like objects, classes, and variables represent more complex underlying code and data. This is important because it lets avoid repeating the same work multiple times.

```
jLabel3.setText("JUMLAH");  
getContentPane().add(jLabel3);  
jLabel3.setBounds(57, 253, 86, 20);  
getContentPane().add(jumlah);  
jumlah.setBounds(257, 248, 202, 25);
```

READ AND WRITE IMPLEMENTATION

In this project I use JTextField. JTextField is a lightweight component that allows the editing of a single line of text. JTextField is intended to be source-compatible with java.awt.TextField where it is reasonable to do so. This component has capabilities not found in the java.awt.TextField class. The superclass should be consulted for additional capabilities.

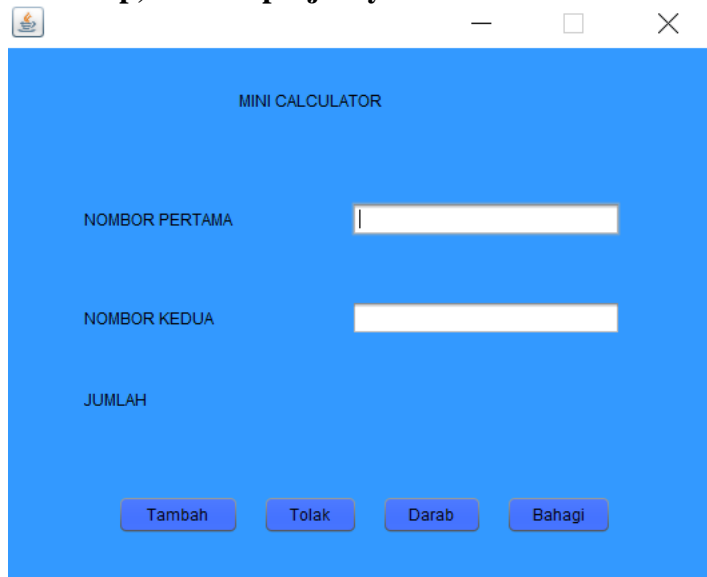
JTextField has a method to establish the string used as the command string for the action event that gets fired. The java.awt.TextField used the text of the field as the command string for the ActionEvent. JTextField will use the command string set with the setActionCommand method if not null, otherwise it will use the text of the field as a compatibility with java.awt.TextField.

```
jTextField1.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jTextField1ActionPerformed(evt);  
    }  
});  
getContentPane().add(jTextField1);  
jTextField1.setBounds(257, 115, 202, 26);  
  
jTextField2.addActionListener(new java.awt.event.ActionListener() {  
    public void actionPerformed(java.awt.event.ActionEvent evt) {  
        jTextField2ActionPerformed(evt);  
    }  
});
```

USER MANUAL

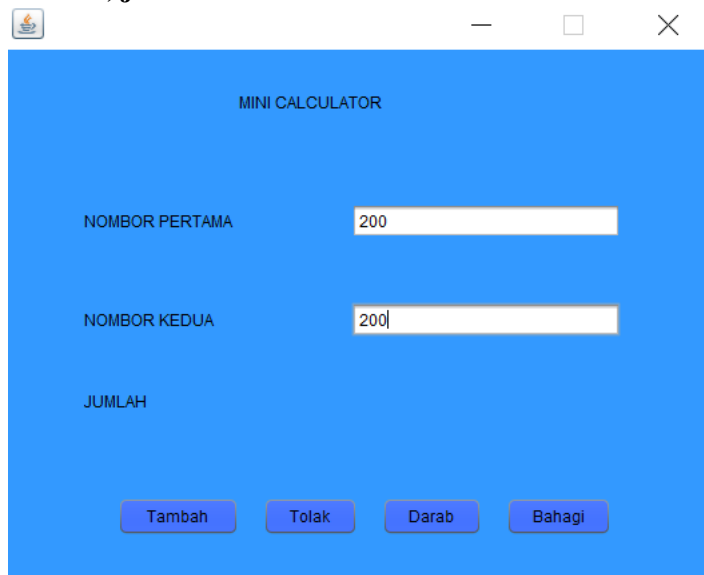
This Mini calculator project is calculator with a basic operation. For example addition, subtraction, divided and multiplication.

First step, run the project you will see this GUI.



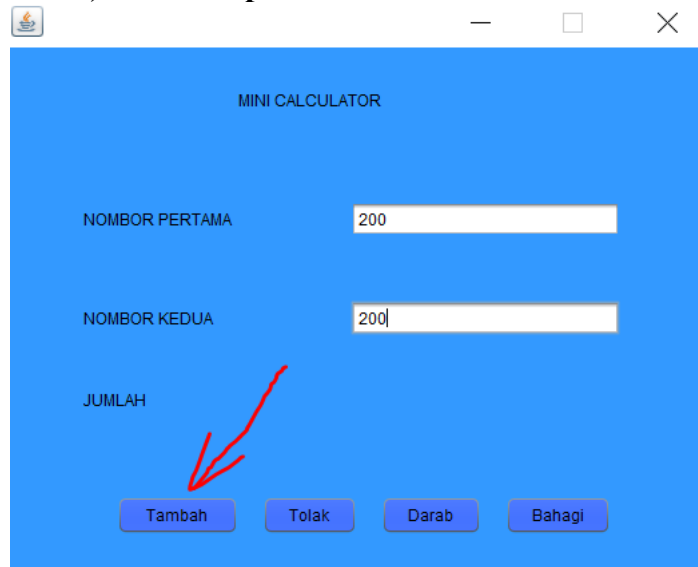
A screenshot of a Java Swing window titled "MINI CALCULATOR". The window has a blue background and a standard Windows-style title bar with a minimize button, a maximize button (disabled), and a close button. Inside the window, there are three labels: "NOMBOR PERTAMA", "NOMBOR KEDUA", and "JUMLAH". Below "NOMBOR PERTAMA" and "NOMBOR KEDUA" are empty text input fields. At the bottom of the window, there are four buttons: "Tambah", "Tolak", "Darab", and "Bahagi".

Second, just insert the number on “Nombor Pertama” and “Nombor Kedua”.



A screenshot of the same "MINI CALCULATOR" window. The "NOMBOR PERTAMA" input field now contains the number "200" and the "NOMBOR KEDUA" input field now contains the number "200". The "JUMLAH" label and the four operation buttons ("Tambah", "Tolak", "Darab", "Bahagi") remain unchanged.

Third, click the operation name.



MINI CALCULATOR

NOMBOR PERTAMA

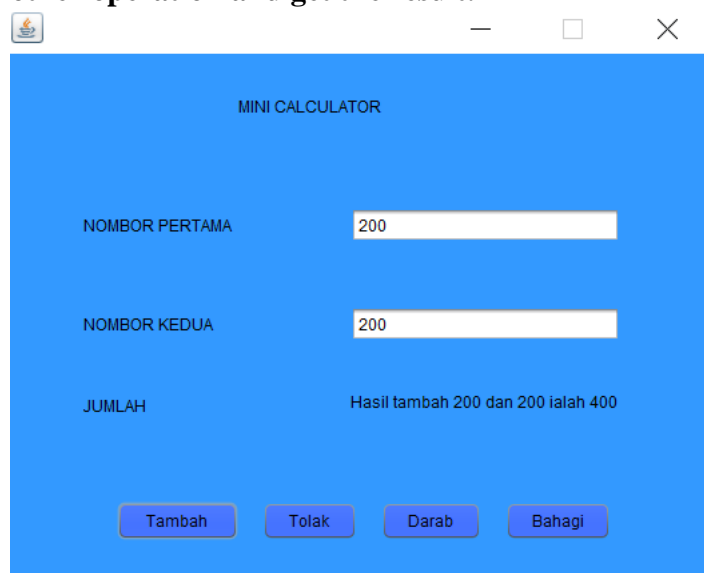
NOMBOR KEDUA

JUMLAH

Tambah Tolak Darab Bahagi

A red arrow points to the 'Tambah' button.

Fourth, just click the operation name button to get the result. You can directly change to other operation and get the result.



MINI CALCULATOR

NOMBOR PERTAMA

NOMBOR KEDUA

JUMLAH Hasil tambah 200 dan 200 ialah 400

Tambah Tolak Darab Bahagi

CONCLUSION

In conclusion, this is a very simple calculator since this is my first time making GUI project. However, making this project is a good way to expand my knowledge in java. Last but not least, I will keep improve this calculator system if there have any issues from other user.