

**LAPORAN TUGAS PEKAN 7**  
**ALGORITMA DAN PEMROGAMAN**

DI SUSUN OLEH :

DIGO YUANDRA

NIM 2511533017

DOSEN PENGAMPU : Dr. WAHYUDI, S.T, M.T

ASISTEN LABORATORIUM: JOVANTRI IMMANUEL GULO



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**PADANG, 2025**

**Soal:**

**Buatlah sebuah program GUI menggunakan Java Swing yang berfungsi sebagai kalkulator operator relasional**

### Kode program

```
package pekan8_2511533017;

import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.SwingConstants;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class OperatorRelasionalGUI_2511533017 extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField txtBil1;
    private JTextField txtBil2;
    private JTextField txtHasil;
    private JComboBox<String> cbOperator;

    private void pesanPeringatan(String pesan) {
        JOptionPane.showMessageDialog(this, pesan, "Peringatan",
JOptionPane.WARNING_MESSAGE);
    }

    private void pesanError(String pesan) {
        JOptionPane.showMessageDialog(this, pesan, "Kesalahan",
JOptionPane.ERROR_MESSAGE);
    }
}
```

```

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                OperatorRelasionalGUI_2511533017 frame = new
OperatorRelasionalGUI_2511533017();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

/**
 * Create the frame.
 */
public OperatorRelasionalGUI_2511533017() {
    setTitle("Operator Relasional");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 326, 262);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("OPERATOR RELASIONAL");
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    lblNewLabel.setFont(new Font("Segoe UI Semibold", Font.BOLD, 15));
    lblNewLabel.setBounds(10, 4, 292, 21);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
    lblNewLabel_1.setBounds(10, 37, 72, 14);
    contentPane.add(lblNewLabel_1);

    JLabel lblNewLabel_2 = new JLabel("Bilangan 2");

```

```
lblNewLabel_2.setBounds(10, 61, 72, 14);
contentPane.add(lblNewLabel_2);
```

```
JLabel lblNewLabel_2_1 = new JLabel("Operator");
lblNewLabel_2_1.setBounds(10, 112, 72, 14);
contentPane.add(lblNewLabel_2_1);
```

```
JLabel lblNewLabel_2_1_1 = new JLabel("Hasil");
lblNewLabel_2_1_1.setBounds(10, 160, 72, 14);
contentPane.add(lblNewLabel_2_1_1);
```

```
txtBil1 = new JTextField();
txtBil1.setHorizontalAlignment(SwingConstants.CENTER);
txtBil1.setBounds(82, 33, 52, 20);
contentPane.add(txtBil1);
txtBil1.setColumns(10);
```

```
txtBil2 = new JTextField();
txtBil2.setHorizontalAlignment(SwingConstants.CENTER);
txtBil2.setColumns(10);
txtBil2.setBounds(82, 58, 52, 20);
contentPane.add(txtBil2);
```

```
txtHasil = new JTextField();
txtHasil.setBounds(82, 157, 96, 20);
contentPane.add(txtHasil);
txtHasil.setColumns(10);
txtHasil.setEditable(false);
```

```
cbOperator = new JComboBox<>();
cbOperator.setModel(new DefaultComboBoxModel<String>(
    new String[] { "=", "!=", ">", "<", ">=", "<=" }));
cbOperator.setBounds(82, 108, 50, 22);
contentPane.add(cbOperator);
```

```
JButton btnHitung = new JButton("Hitung");
btnHitung.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        hitungOperatorRelasional();
    }
});
```

```

});
btnHitung.setBounds(156, 108, 88, 22);
contentPane.add(btnHitung);
}

private void hitungOperatorRelasional() {
    if (txtBil1.getText().trim().isEmpty()) {
        pesanPeringatan("Silahkan Masukkan Bilangan 1");
        txtBil1.requestFocus();
        return;
    }

    if (txtBil2.getText().trim().isEmpty()) {
        pesanPeringatan("Silahkan Masukkan Bilangan 2");
        txtBil2.requestFocus();
        return;
    }

    try {
        double bilangan1 = Double.parseDouble(txtBil1.getText());
        double bilangan2 = Double.parseDouble(txtBil2.getText());

        String operator = (String) cbOperator.getSelectedItem();
        boolean hasil = false;

        switch(operator) {
            case "==":
                hasil = (bilangan1 == bilangan2);
                break;
            case "!=":
                hasil = (bilangan1 != bilangan2);
                break;
            case ">":
                hasil = (bilangan1 > bilangan2);
                break;
            case "<":
                hasil = (bilangan1 < bilangan2);
                break;
            case ">=":
                hasil = (bilangan1 >= bilangan2);

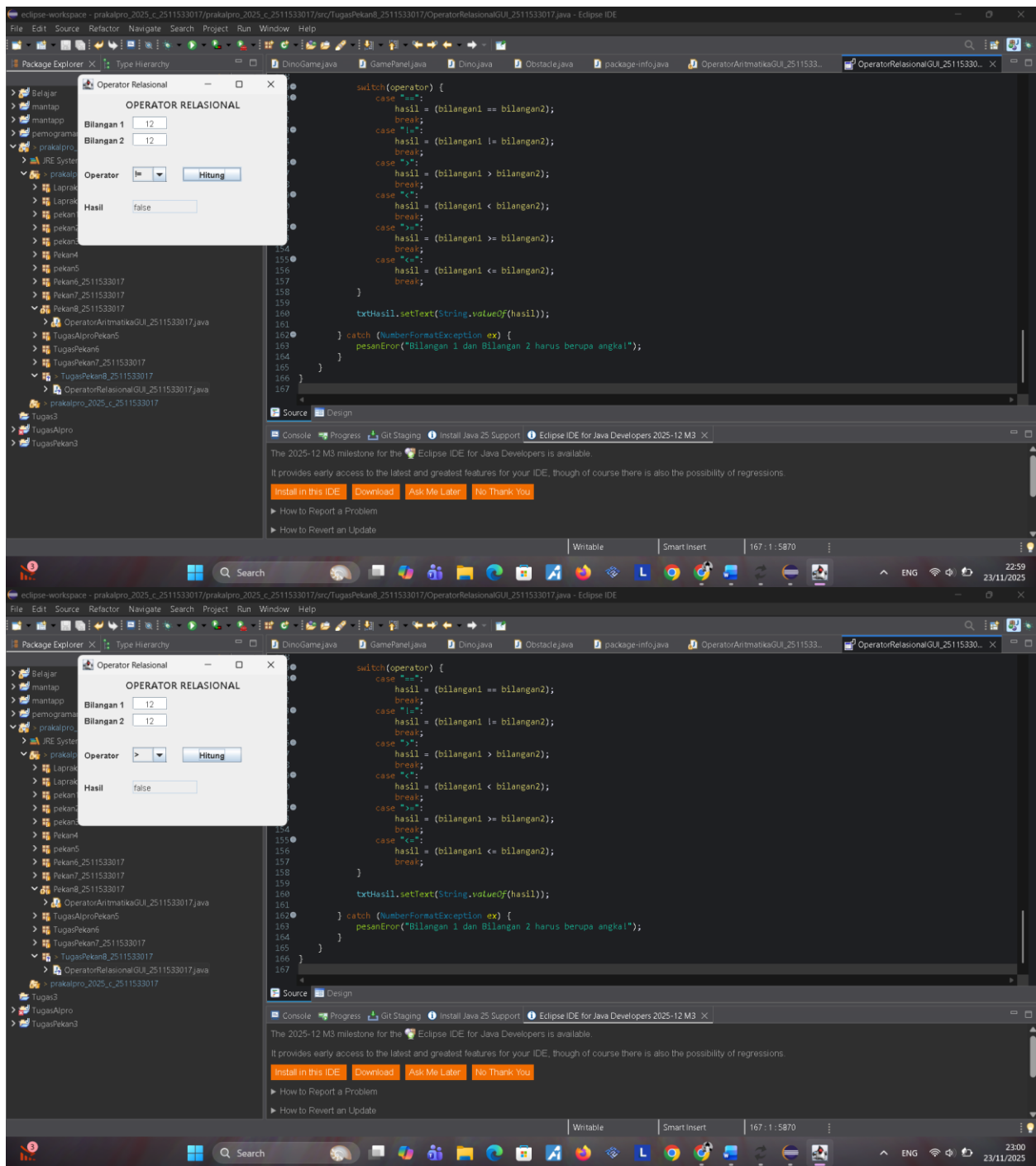
```

```
        break;
    case "<=":
        hasil = (bilangan1 <= bilangan2);
        break;
    }

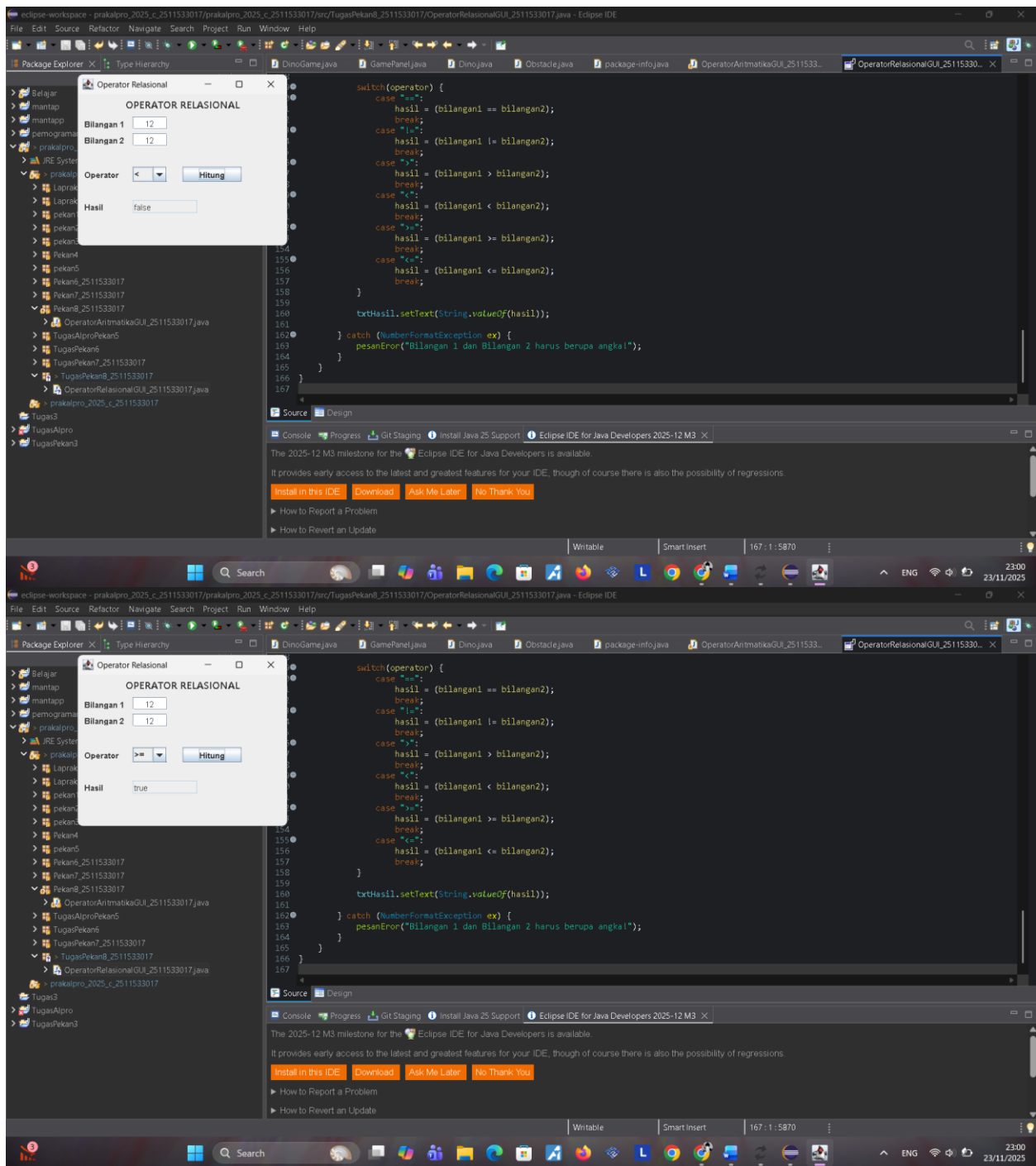
    txtHasil.setText(String.valueOf(hasil));

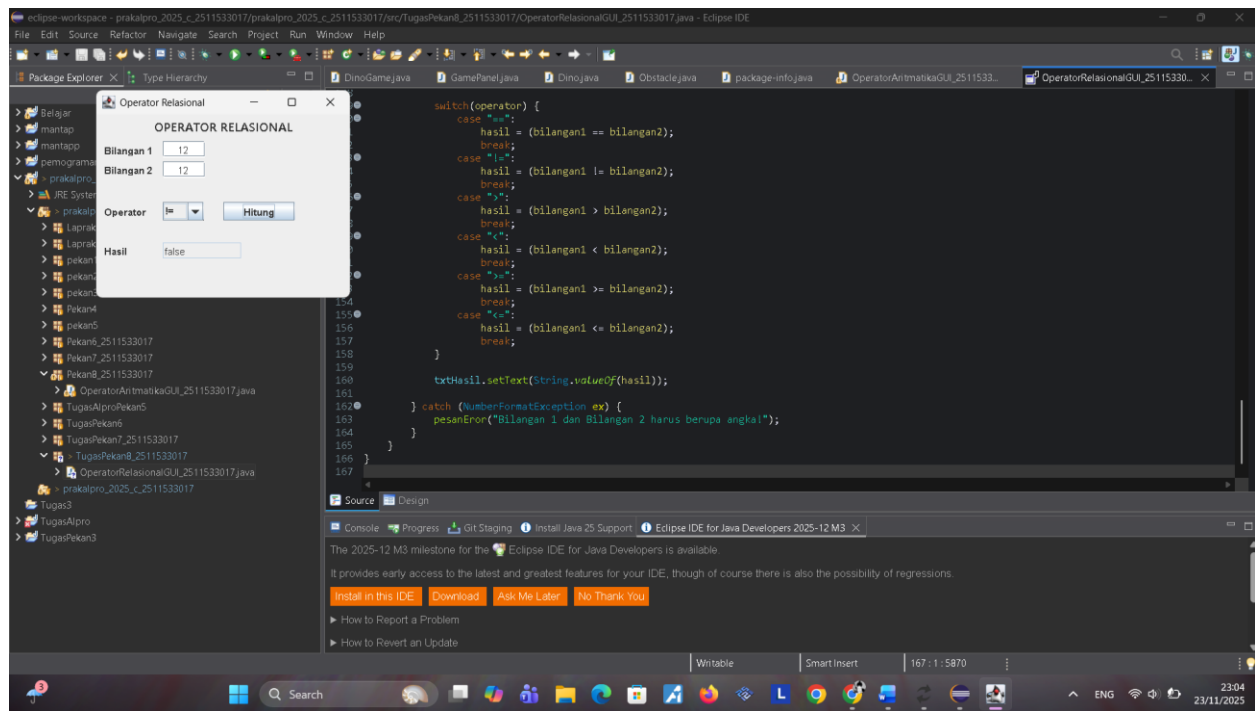
} catch (NumberFormatException ex) {
    pesanError("Bilangan 1 dan Bilangan 2 harus berupa angka!");
}
}
```

**Output yang dihasilkan**



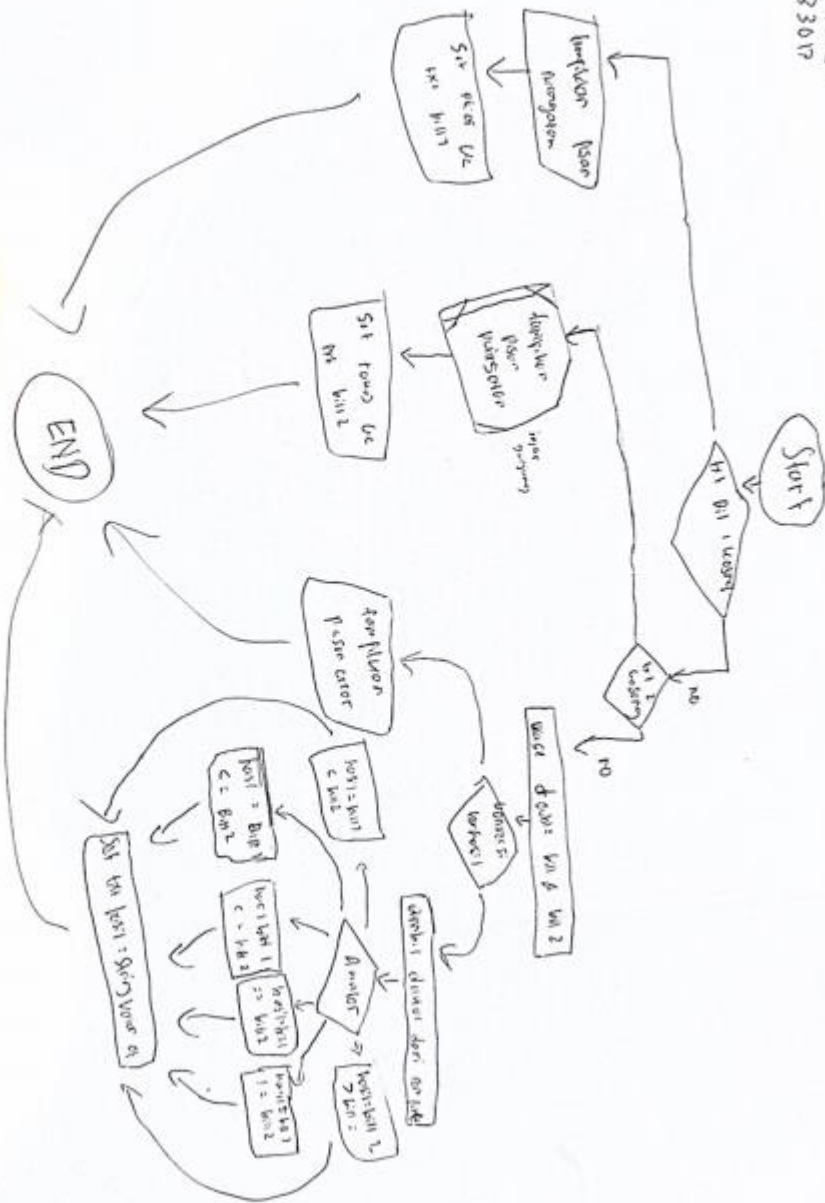






## Flowchart

Num 9 7511533017



## **Pseudocode**

Judul

Program Operator Relasional GUI

(Program GUI untuk melakukan operasi perbandingan antara dua bilangan dengan berbagai operator relasional)

CLASS OperatorRelasionalGUI\_2511533017

· ATTRIBUTES

- contentPane : JPanel
- txtBil1, txtBil2 : JTextField
- txtHasil : JTextField
- cbOperator : JComboBox<String>

· METHODS

- OperatorRelasionalGUI\_2511533017() (Konstruktor)
- pesanPeringatan(pesan: String)
- pesanError(pesan: String)
- hitungOperatorRelasional()
- main(args: String[])

Pseudocode

METHOD pesanPeringatan(pesan: String)

1. TAMPILKAN dialog peringatan dengan pesan tertentu

METHOD pesanError(pesan: String)

1. TAMPILKAN dialog error dengan pesan tertentu

METHOD hitungOperatorRelasional()

1. JIKA txtBil1 kosong ATAU hanya berisi spasi MAKA
  1. PANGGIL pesanPeringatan("Silahkan Masukkan Bilangan 1")
  2. SET fokus ke txtBil1
  3. RETURN
2. JIKA txtBil2 kosong ATAU hanya berisi spasi MAKA
  1. PANGGIL pesanPeringatan("Silahkan Masukkan Bilangan 2")
  2. SET fokus ke txtBil2
  3. RETURN
3. COBA
  1. bilangan1 = konversi txtBil1.getText() ke Double
  2. bilangan2 = konversi txtBil2.getText() ke Double
  3. operator = ambil nilai dari cbOperator
  4. hasil = false
  5. SWITCH (operator)

- CASE "==" : hasil = (bilangan1 == bilangan2)
  - CASE "!=" : hasil = (bilangan1 != bilangan2)
  - CASE ">" : hasil = (bilangan1 > bilangan2)
  - CASE "<" : hasil = (bilangan1 < bilangan2)
  - CASE ">=" : hasil = (bilangan1 >= bilangan2)
  - CASE "<=" : hasil = (bilangan1 <= bilangan2)
6. SET txtHasil.setText(hasil sebagai String)

4. CATCH (NumberFormatException)

5. PANGGIL pesanError("Bilangan 1 dan Bilangan 2 harus berupa angka!")

## METHOD MAIN

1. JALANKAN dalam Event Dispatch Thread:

1. BUAT objek OperatorRelasionalGUI\_2511533017
2. SET frame visible = true

KONSTRUKTOR OperatorRelasionalGUI\_2511533017()

1. SET judul window: "Operator Relasional"
2. SET operasi default saat tutup: EXIT\_ON\_CLOSE
3. SET ukuran dan posisi window
4. BUAT contentPane dengan border
5. ATUR layout contentPane
6. TAMBAH komponen GUI:
  - Label "OPERATOR RELASIONAL" (judul utama)
  - Label "Bilangan 1", "Bilangan 2", "Operator", "Hasil"
  - TextField untuk input Bilangan 1 dan Bilangan 2
  - TextField untuk menampilkan Hasil (non-editable)
  - ComboBox dengan pilihan operator: "==", "!=", ">", "<", ">=", "<="
  - Button "Hitung" yang memanggil hitungOperatorRelasional() ketika diklik