

**LAPORAN ALGORITMA DAN PEMROGRAMAN
TUGAS PEKAN 8 PEMROGRAMAN GUI 1**



Oleh :

DIKA GIWANDA

NIM 2511533025

Kelas :

A

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

JOVANTRI IMMANUEL GULO

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

INTRUKSI

Buatlah kode program java Pemrograman GUI tentang Operator Relasional. Beserta Bahasa Natural, Flowchart, Pseudocode, dan Penjelasan singkat mengenai kode program.

KODE PROGRAM

```
package TugasPraktikumPekan8_2511533025;

import java.awt.EventQueue;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JTextField;
import javax.swing.JComboBox;
import javax.swing.JButton;
import javax.swing.DefaultComboBoxModel;

public class OperatorRelasional_2511533025 extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField txtBil1;
    private JTextField txtBil2;
    private JTextField txtHasil;

    private void pesanPeringatan(String pesan) {
        JOptionPane.showMessageDialog(this,pesan,"Peringatan",JOptionPane.WARNING_MESSAGE);
    }
    private void PesanEror(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 {
                    OperatorRelasional_2511533025 frame = new OperatorRelasional_2511533025();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

        }
    });
}

/**
 * Create the frame.
 */
public OperatorRelasional_2511533025() {
    setTitle("Operator Relasional");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblNewLabel = new JLabel("OPERATOR RELASIONAL");
    lblNewLabel.setBounds(150, 10, 149, 16);
    lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
    contentPane.add(lblNewLabel);

    JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
    lblNewLabel_1.setBounds(17, 61, 93, 16);
    contentPane.add(lblNewLabel_1);

    JLabel lblNewLabel_2 = new JLabel("Bilangan 2");
    lblNewLabel_2.setBounds(17, 100, 93, 16);
    contentPane.add(lblNewLabel_2);

    JLabel lblNewLabel_3 = new JLabel("Operator");
    lblNewLabel_3.setBounds(17, 161, 61, 16);
    contentPane.add(lblNewLabel_3);

    JLabel lblNewLabel_4 = new JLabel("Hasil");
    lblNewLabel_4.setBounds(17, 202, 61, 16);
    contentPane.add(lblNewLabel_4);

    txtBil1 = new JTextField();
    txtBil1.setBounds(122, 56, 130, 26);
    contentPane.add(txtBil1);
    txtBil1.setColumns(10);

    txtBil2 = new JTextField();
    txtBil2.setBounds(122, 95, 130, 26);
    contentPane.add(txtBil2);
    txtBil2.setColumns(10);

    txtHasil = new JTextField();
    txtHasil.setBounds(122, 197, 130, 26);
    contentPane.add(txtHasil);
}

```

```

txtHasil.setColumns(10);

JComboBox cbOperator = new JComboBox();
cbOperator.setModel(new DefaultComboBoxModel(new String[] {"==",
"!=", ">", "<", ">=", "<="}));
cbOperator.setBounds(122, 157, 84, 27);
contentPane.add(cbOperator);

JButton btnNewButton = new JButton("Hitung");
btnNewButton.addActionListener(new ActionListener() {
    int hasil;
    public void actionPerformed(ActionEvent e) {
        if (txtBil1.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 1 Harus diisi");
        } else if (txtBil2.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 2 Harus diisi");
        } else
        {
            try {
                int a= Integer.valueOf(txtBil1.getText());
                int b= Integer.valueOf(txtBil2.getText());
                int c= cbOperator.getSelectedIndex();
                boolean hasil = false;

                switch (c) {
                    case 0:
                        hasil = (a == b);
                        break;
                    case 1:
                        hasil = (a != b);
                        break;
                    case 2:
                        hasil = (a > b);
                        break;
                    case 3:
                        hasil = (a < b);
                        break;
                    case 4:
                        hasil = (a >= b);
                        break;
                    case 5:
                        hasil = (a <= b);
                        break;
                    default:
                        hasil = false;
                        break;
                }
                txtHasil.setText(String.valueOf(hasil));
            } catch (NumberFormatException ex) {
                PesanError("Bilangan 1 dan Bilangan 2 harus
angka");
            }
        }
    }
});
```

```
        }

    }

    });

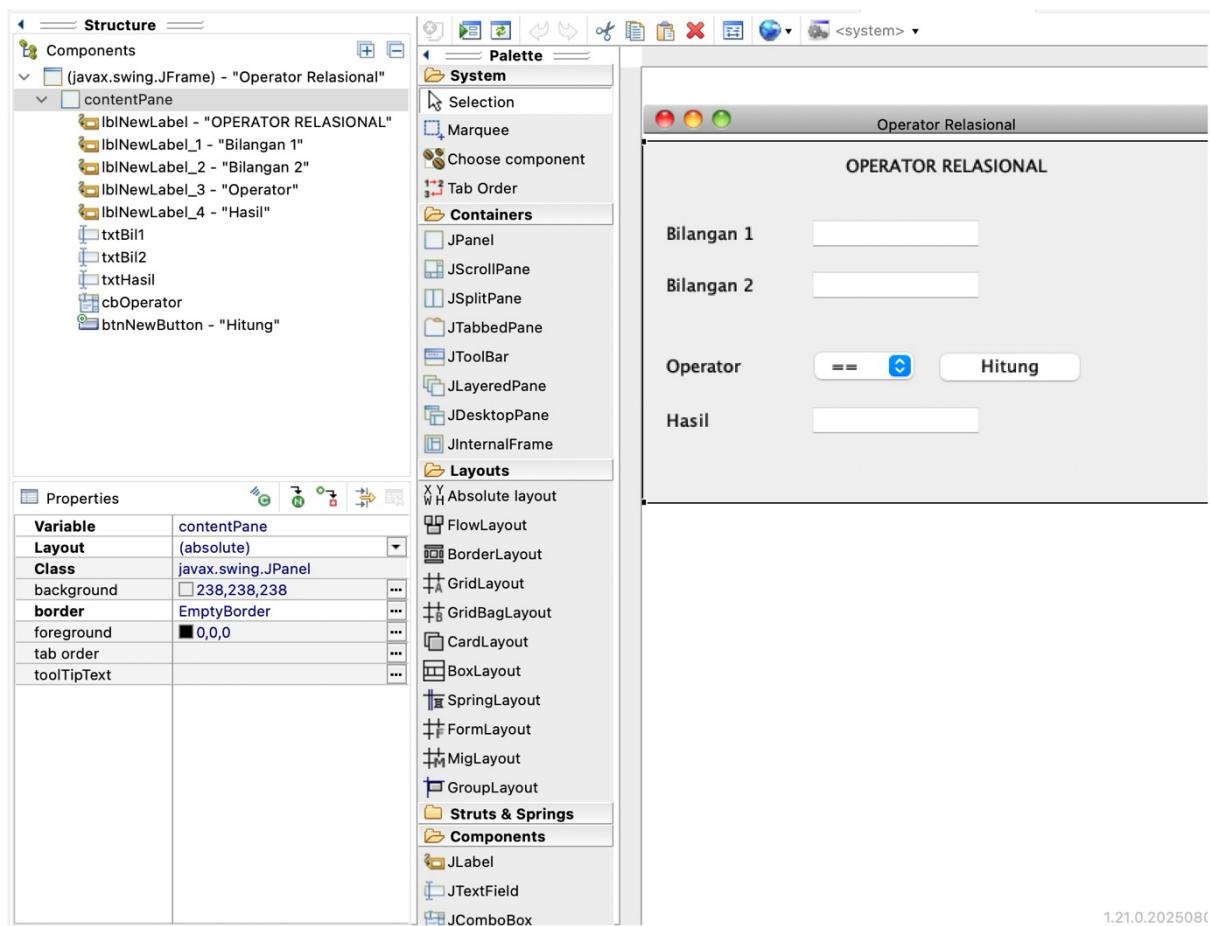
btnNewButton.setBounds(213, 156, 117, 29);
contentPane.add(btnNewButton);

}

}
```

SCREENSHOTS KODE PROGRAM DAN HASIL KODE PROGRAM

A. Kode Progam



```
1 package TugasPraktikumPekan8_2511533025;
2
3+import java.awt.EventQueue;[]
17
18 public class OperatorRelasional_2511533025 extends JFrame {
19
20     private static final long serialVersionUID = 1L;
21     private JPanel contentPane;
22     private JTextField txtBil1;
23     private JTextField txtBil2;
24     private JTextField txtHasil;
25
26
27+    private void pesanPeringatan(String pesan) {
28         JOptionPane.showMessageDialog(this,pesan,"Peringatan",JOptionPane.WARNING_MESSAGE);
29     }
30+    private void PesanEror(String pesan) {
31         JOptionPane.showMessageDialog(this,pesan,"Kesalahan",JOptionPane.ERROR_MESSAGE);
32     }
33
34+    /**
35     * Launch the application.
36     */
37+    public static void main(String[] args) {
38         EventQueue.invokeLater(new Runnable() {
39             public void run() {
40                 try {
41                     OperatorRelasional_2511533025 frame = new OperatorRelasional_2511533025();
42                     frame.setVisible(true);
43                 } catch (Exception e) {
44                     e.printStackTrace();
45                 }
46             }
47         });
48     }
49
50+    /**
51     * Create the frame.
52     */
53+    public OperatorRelasional_2511533025() {
54         setTitle("Operator Relasional");
55         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
56         setBounds(100, 100, 450, 300);
57         contentPane = new JPanel();
58         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
59         setContentPane(contentPane);
60         contentPane.setLayout(null);
61
62         JLabel lblNewLabel = new JLabel("OPERATOR RELASIONAL");
63         lblNewLabel.setBounds(150, 10, 149, 16);
64         lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
65         contentPane.add(lblNewLabel);
66     }

```

```

102
103     JButton btnNewButton = new JButton("Hitung");
104     btnNewButton.addActionListener(new ActionListener() {
105         int hasil;
106         public void actionPerformed(ActionEvent e) {
107             if (txtBil1.getText().trim().isEmpty()) {
108                 pesanPeringatan("Bilangan 1 Harus diisi");
109             } else if (txtBil2.getText().trim().isEmpty()) {
110                 pesanPeringatan("Bilangan 2 Harus diisi");
111             } else
112             {
113                 try {
114                     int a= Integer.valueOf(txtBil1.getText());
115                     int b= Integer.valueOf(txtBil2.getText());
116                     int c= cbOperator.getSelectedIndex();
117                     boolean hasil = false;
118
119                     switch (c) {
120                         case 0:
121                             hasil = (a == b);
122                             break;
123                         case 1:
124                             hasil = (a != b);
125                             break;
126                         case 2:
127                             hasil = (a > b);
128                             break;
129                         case 3:
130                             hasil = (a < b);
131                             break;
132                         case 4:
133                             hasil = (a >= b);
134                             break;
135                         case 5:
136                             hasil = (a <= b);
137                             break;
138                         default:
139                             hasil = false;
140                             break;
141                     }
142                     txtHasil.setText(String.valueOf(hasil));
143                 } catch (NumberFormatException ex) {
144                     PesanError("Bilangan 1 dan Bilangan 2 harus angka");
145                 }
146             }
147         }
148     });
149     btnNewButton.setBounds(213, 156, 117, 29);
150     contentPane.add(btnNewButton);
151
152
153 }
154 }
```

B. Hasil Kode Program

OPERATOR RELASIONAL

Bilangan 1

2

Bilangan 2

2

Operator

`==`



Hitung

Hasil

true

OPERATOR RELASIONAL

Bilangan 1

2

Bilangan 2

2

Operator

`<`



Hitung

Hasil

false

OPERATOR RELASIONAL

Bilangan 1

2

Bilangan 2

2

Operator

!=



Hitung

Hasil

false

OPERATOR RELASIONAL

Bilangan 1

2

Bilangan 2

2

Operator

>=



Hitung

Hasil

true

OPERATOR RELASIONAL

Bilangan 1

2

Bilangan 2

2

Operator

>



Hitung

Hasil

false

OPERATOR RELASIONAL

Bilangan 1

2

Bilangan 2

2

Operator

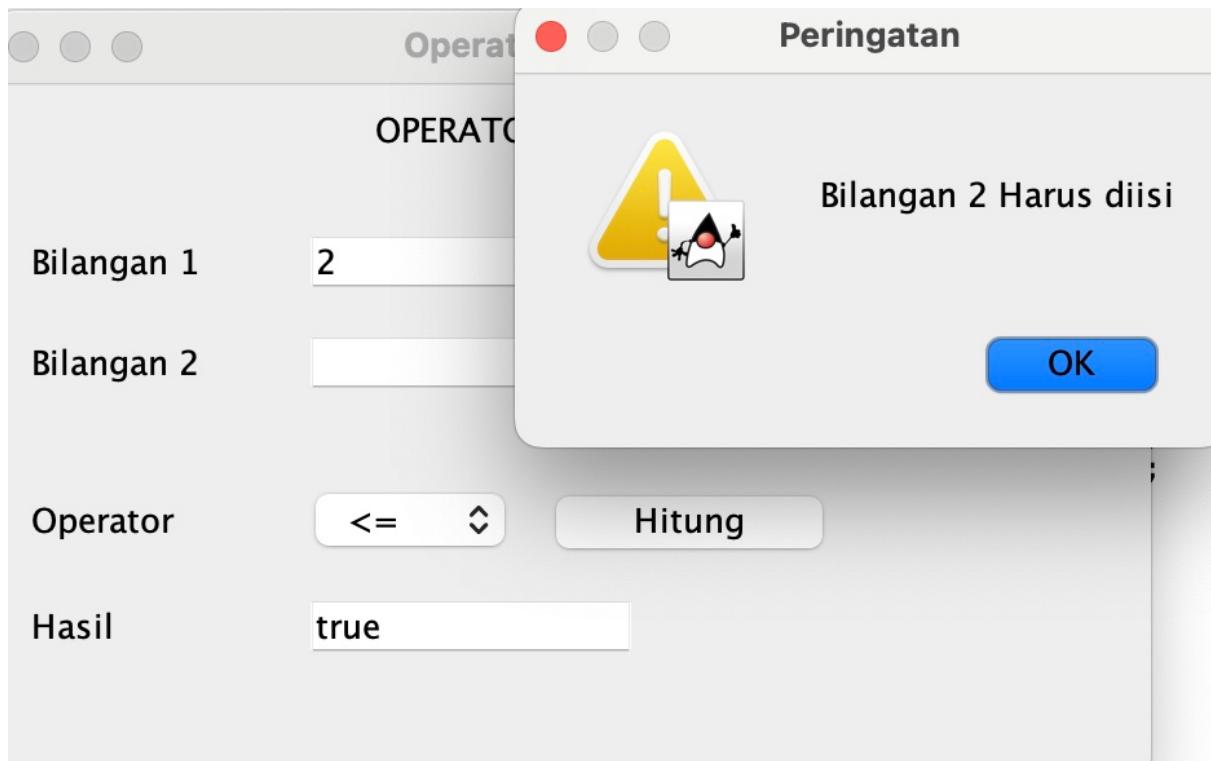
<=



Hitung

Hasil

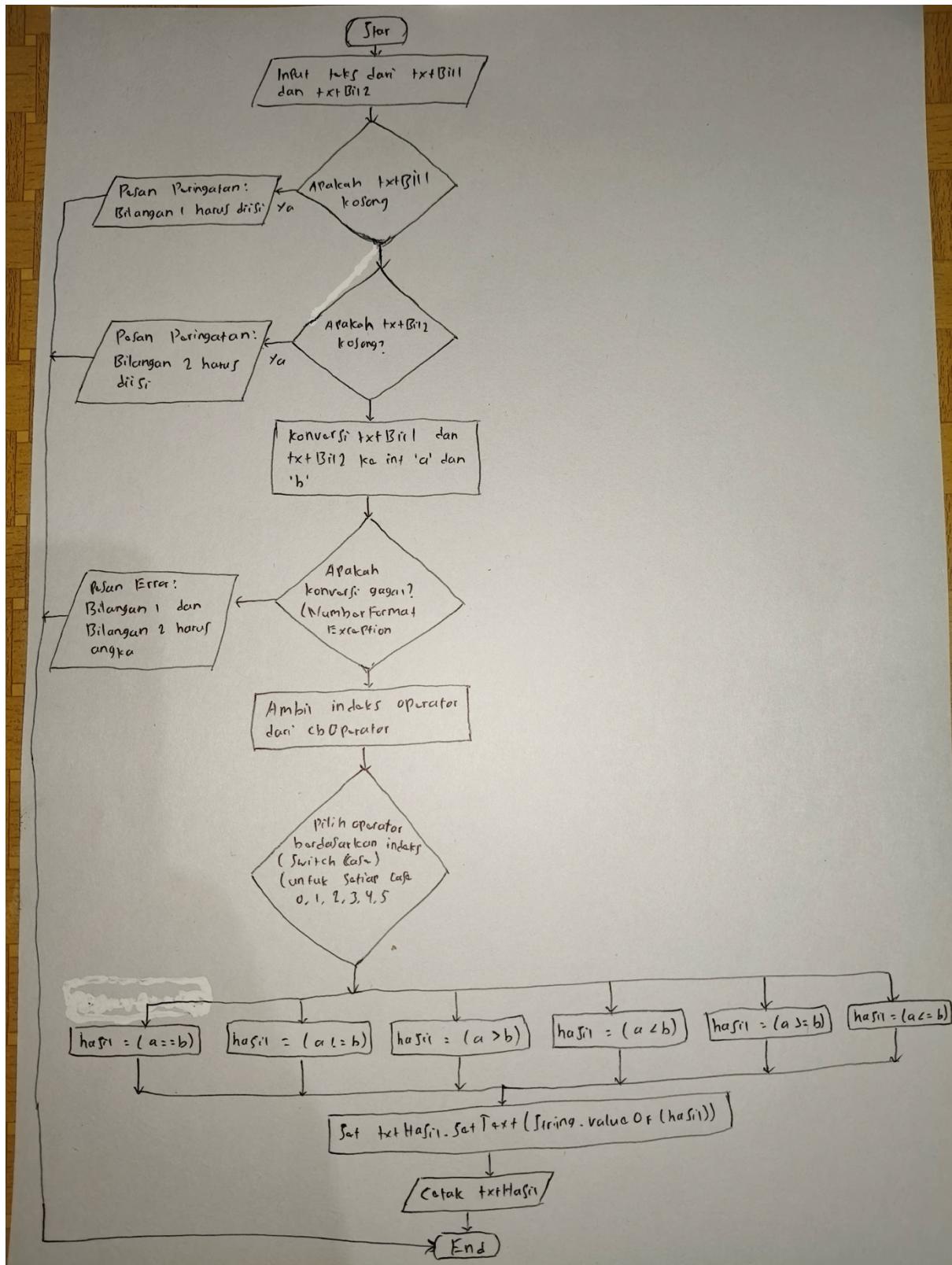
true



BAHASA NATURAL

1. Pembukaan Aplikasi: Saat program dijalankan sebuah jendela atau (frame) dengan judul “Operator Relasional” akan muncul dilayar.
2. Tampilan Antarmuka: Jendela memiliki beberapa elemen:
 - Sebuah label utama: “OPERATOR RELASIONAL”.
 - Dua input teks bernama “Bilangan 1” dan “Bilangan 2” untuk memasukan angka.
 - Sebuah components JComboBox bernama “Operator” yang berisi pilihan operator: ==, !=, >, <, >=, <=.
 - Satu tombol bernama “Hitung”.
 - Satu area hasil bernama “Hasil” untuk menampilkan output perbandingan.
3. Proses Perhitungan:
 - Pengguna memasukan dua angka ke dalam kolom “Bilangan 1” dan “Bilangan 2”, lalu memilih satu operator dari JComboBox.
 - Ketika pengguna menekan tombol “Hitung”, program akan memeriksa apakah kedua kolom input sudah diisi.
 - Jika salah satu kolom kosong, program akan menampilkan pesan peringatan: “Bilangan 1 harus diisi” atau “Bilangan 2 harus diisi”.
 - Jika kedua kolom terisi, program akan mencoba mengubah teks input menjadi angka bulat (int).
 - Jika konversi gagal (misalnya, pengguna memasukan huruf), program akan menampilkan pesan eror. “Bilangan 1 dan Bilangan 2 harus diisi dengan angka”.
 - Jika konversi berhasil, program akan membaca indeks operator yang dipilih dari JComboBox.
 - Berdasarkan indeks tersebut, program akan melakukan perbandingan antara dua angka menggunakan operator yang sesuai.
 - Misalnya, jika indeks 0 dipilih, program akan membandingkan apakah Bilangan 1 == Bilangan 2.
 - Hasil dari perbandingan ini adalah nilai true (benar) atau false (salah).
 - Hasil perbandingan (true atau false) kemudian ditampilkan di kolom “Hasil”.

FLOWCHART



BAHASA PSEUDOCODE

Judul:

Program Operator Relasional

{program untuk membandingkan dua bilangan berdasarkan operator relasional yang dipilih}

Deklarasi:

```
Var txtBil1      : String  
Var txtBil2      : String  
Var cbOperator   : Int  
Var a            : Int  
Var b            : Int  
Var hasil        : boolean  
Var pesanError. : String
```

Pseudocode:

1. start
2. Inisialisasi komponen GUI: frame, panel, textfield, combobox, button
3. Cetak jendela aplikasi
4. input(Bil1)
5. input(Bil2)
6. input(cbOperator)
7. if Bil1 kosong then
8. cetak("Bilangan 1 harus diisi")
9. else if Bil2 kosong then
10. cetak("Bilangan 2 harus diisi")
11. else
12. coba:
13. a ← konversi ke integer(Bil1)
14. b ← konversi ke integer(Bil2)
15. if cbOperator == 0 then
16. hasil ← (a == b)

```
17.     else if cbOperator == 1 then
18.         hasil ← (a != b)
19.     else if cbOperator == 2 then
20.         hasil ← (a > b)
21.     else if cbOperator == 3 then
22.         hasil ← (a < b)
23.     else if cbOperator == 4 then
24.         hasil ← (a >= b)
25.     else if cbOperator == 5 then
26.         hasil ← (a <= b)
27.     else
28.         hasil ← false
29.     end if
30.     cetak("Hasil: ", hasil)
31. tangkap error:
32     cetak("Bilangan 1 dan Bilangan 2 harus angka")
33. end coba
34. end if
35. end
```

Penjelasan Singkat Kode Program

Kode program ini adalah desktop sederhana berbasis Java Swing yang dirancang untuk membandingkan dua bilangan bulat menggunakan operator relasional (seperti ==, !=, >, <, >=, <=). Cara kerja:

1. Pengguna memasukan dua angka dikolom “Bilangan 1” dan Bilangan 2”.
2. Pengguna memilih operator relasional dari JComboBox menu.
3. Saat tombol “Hitung” ditekan, program akan:
 - Memvalidasi apakah kedua input sudah diisi
 - Mengonversi input menjadi angka bulat.
 - Melakukan perbandingan sesuai dengan operator yang dipilih.
 - Menampilkan hasilnya sebagai true atau false dikolom “Hasil”

Kemudian pada fitur keamanan:

- Jika salah satu kolom kosong → muncul pesan peringatan.
- Jika input bukan angka → muncul pesan eror.
- Semua operasi dilindungi dengan blok try-catch untuk menangani kesalahan konversi.

Dan program ini merupakan contoh sederhana penerapan logika percabangan (switch/if-else) dan validasi input dalam GUI java.