

**LAPORAN TUGAS**  
**ALGORITMA PEMROGRAMAN KELAS C**

**“OPERASI RELASIONAL GUI”**

**DISUSUN OLEH:**

**ALIYATAR RAFI AHMAD**

**2511533031**

**DOSEN PENGAMPU:**

**Dr. WAHYUDI, S.T, M.T**

**ASISTEN PRAKTIKUM:**

**JOVANTRI IMMANUEL GULO**



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS ANDALAS**

**2025**

**Tugas pekan 8 praktikum algoritma pemrograman :**

1. Buatlah program Graphical User Interface (GUI) menggunakan WindowBuilder untuk mensimulasikan penggunaan operator relasional dalam pemrosesan data input user. Program harus dapat menerima dua nilai bilangan, melakukan perbandingan menggunakan operator relasional, dan menampilkan hasilnya secara interaktif.
2. Cantumkan :
  - Pseudocode
  - Kode program java
  - Flowchart
  - Output
3. Buat penjelasan singkat mengenai program tersebut.

## 1. Pseudocode

### **Judul**

{program untuk membandingkan dua bilangan menggunakan operator relasional}

### **Deklarasi**

a,b : integer

operator : string

Hasil : Boolean

### **Algoritma**

1. input (a), (b), (Operator)
2. if a Isempty, then  
    print "Bilangan 1 Harus diisi"
3. if b Isempty, then  
    print "Bilangan 2 Harus diisi"
4. if catch NumberFormatException, then  
    print "Bilangan 1 dan 2 harus angka"
5. if (c = 0) then  $\rightarrow (a > b)$
6. else if (c = 1) then  $\rightarrow (a < b)$
7. else if (c = 2) then  $\rightarrow (a >= b)$
8. else if (c = 3) then  $\rightarrow (a <= b)$
9. else if (c = 4) then  $\rightarrow (a == b)$
10. else if (c = 5) then  $\rightarrow (a != b)$
11. End if
12. Tampilkan Hasil
13. End

## 2. Kode Program Java

```
1 package pekan8_2511533031;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8 import java.awt.GridLayout;
9 import java.awt.Font;
10 import javax.swing.JTextField;
11 import javax.swing.JLabel;
12 import javax.swing.JOptionPane;
13 import javax.swing.JComboBox;
14 import javax.swing.JButton;
15 import javax.swing.DefaultComboBoxModel;
16 import java.awt.event.ActionListener;
17 import java.awt.event.ActionEvent;
18 import java.awt.Color;
19
20 public class tugasPraktikumOperasiRelasionalGUI_2511533031 extends JFrame {
21
22     private static final long serialVersionUID = 1L;
23     private JPanel contentPane;
24     private JTextField txtBill1;
25     private JTextField txtBill2;
26     private JTextField txtHasil;
27
28     private void pesanPeringatan(String pesan) {
29         JOptionPane.showMessageDialog(this, pesan, "peringatan", JOptionPane.WARNING_MESSAGE);
30     }
31     private void pesanError(String pesan) {
32         JOptionPane.showMessageDialog(this, pesan, "kesalahan", JOptionPane.ERROR_MESSAGE);
33     }
34
35     /**
36      * Launch the application.
37      */
38     public static void main(String[] args) {
39         EventQueue.invokeLater(new Runnable() {
40             public void run() {
```

```
41             try {
42                 tugasPraktikumOperasiRelasionalGUI_2511533031 frame = new tugasPraktikumOperasiRelasionalGUI_2511533031();
43                 frame.setVisible(true);
44             } catch (Exception e) {
45                 e.printStackTrace();
46             }
47         });
48     }
49 }
50
51 /**
52  * Create the frame.
53  */
54 public tugasPraktikumOperasiRelasionalGUI_2511533031() {
55     setBackground(new Color(39, 104, 152));
56     setFont(new Font("Times New Roman", Font.BOLD, 12));
57     setTitle("Operator Relasional");
58     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
59     setBounds(100, 100, 476, 359);
60     contentPane = new JPanel();
61     contentPane.setBackground(new Color(180, 191, 205));
62     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
63     setContentPane(contentPane);
64     contentPane.setLayout(null);
65
66     txtBill1 = new JTextField();
67     txtBill1.setBounds(133, 52, 69, 20);
68     txtBill1.setBackground(new Color(255, 255, 255));
69     contentPane.add(txtBill1);
70     txtBill1.setColumns(10);
71
72     JLabel lblNewLabel = new JLabel("OPERATOR RELASIONAL");
73     lblNewLabel.setForeground(new Color(255, 255, 255));
74     lblNewLabel.setBounds(133, 11, 181, 14);
75     lblNewLabel.setBackground(new Color(192, 192, 192));
76     lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 14));
77     contentPane.add(lblNewLabel);
78
79     JLabel lblNewLabel_1 = new JLabel("Billangan 1");
80     lblNewLabel_1.setForeground(new Color(255, 255, 255));
```

```

81 lblNewLabel_1.setBounds(39, 51, 108, 20);
82 lblNewLabel_1.setBackground(new Color(240, 240, 240));
83 lblNewLabel_1.setFont(new Font("Times New Roman", Font.PLAIN, 12));
84 contentPane.add(lblNewLabel_1);
85
86 JLabel lblNewLabel_2 = new JLabel("Bilangan 2");
87 lblNewLabel_2.setForeground(new Color(255, 255, 255));
88 lblNewLabel_2.setBounds(39, 82, 69, 14);
89 lblNewLabel_2.setFont(new Font("Times New Roman", Font.PLAIN, 12));
90 contentPane.add(lblNewLabel_2);
91
92 txtBil2 = new JTextField();
93 txtBil2.setBounds(133, 83, 69, 20);
94 txtBil2.setBackground(new Color(255, 255, 255));
95 contentPane.add(txtBil2);
96 txtBil2.setColumns(10);
97
98 JLabel lblNewLabel_3 = new JLabel("Operator");
99 lblNewLabel_3.setForeground(new Color(255, 255, 255));
100 lblNewLabel_3.setBounds(41, 126, 48, 14);
101 lblNewLabel_3.setFont(new Font("Times New Roman", Font.PLAIN, 12));
102 contentPane.add(lblNewLabel_3);
103
104 JLabel lblNewLabel_4 = new JLabel("Hasil");
105 lblNewLabel_4.setForeground(new Color(255, 255, 255));
106 lblNewLabel_4.setBounds(41, 179, 48, 14);
107 lblNewLabel_4.setFont(new Font("Times New Roman", Font.PLAIN, 12));
108 contentPane.add(lblNewLabel_4);
109
110 txtHasil = new JTextField();
111 txtHasil.setBounds(133, 177, 79, 20);
112 txtHasil.setBackground(new Color(255, 255, 255));
113 contentPane.add(txtHasil);
114 txtHasil.setColumns(10);
115
116 JComboBox cbOperator = new JComboBox();
117 cbOperator.setBounds(143, 123, 48, 22);
118 cbOperator.setBackground(new Color(255, 255, 255));
119 cbOperator.setModel(new DefaultComboBoxModel(new String[] { ">", "<", ">=", "<=", "==", "!=" }));
120 contentPane.add(cbOperator);

```

```

121
122 JButton btnNewButton = new JButton("Hitung");
123 btnNewButton.setBounds(225, 123, 89, 23);
124 btnNewButton.setBackground(new Color(255, 255, 255));
125 btnNewButton.addActionListener(new ActionListener() {
126     boolean hasil;
127     public void actionPerformed(ActionEvent e) {
128         if (txtBil1.getText().trim().isEmpty()) {
129             pesanPeringatan("Bilangan 1 Harus diisi");
130         } else if (txtBil2.getText().trim().isEmpty()) {
131             pesanPeringatan("Bilangan 2 Harus diisi");
132         } else {
133             try {
134                 int a = Integer.valueOf(txtBil1.getText());
135                 int b = Integer.valueOf(txtBil2.getText());
136                 int c = cbOperator.getSelectedIndex();
137                 if (c==0) { hasil = a > b ; }
138                 if (c==1) { hasil = a < b ; }
139                 if (c==2) { hasil = a >= b ; }
140                 if (c==3) { hasil = a <= b ; }
141                 if (c==4) { hasil = a == b ; }
142                 if (c==5) { hasil = a != b ; }
143                 txtHasil.setText(String.valueOf(hasil));
144             } catch (NumberFormatException ex) {
145                 pesanError("Bilangan 1 dan 2 harus angka");
146             }
147         }
148     }
149 });
150 btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 10));
151 contentPane.add(btnNewButton);
152
153
154 }
155 }

```

### 3. Penjelasan Singkat Kode Program

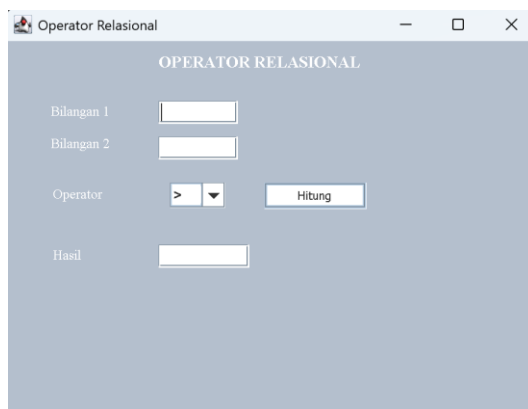
Program ini merupakan aplikasi Java berbasis GUI yang berfungsi untuk melakukan operasi perbandingan antara dua bilangan menggunakan operator relasional. Pengguna memasukkan dua nilai pada textfield yang tersedia, kemudian memilih operator relasional seperti >, <, >=, <=, ==, atau != melalui sebuah combo box. Saat tombol *Hitung* ditekan, program terlebih dahulu melakukan pengecekan apakah kedua input sudah terisi. Jika salah satu textfield masih kosong, program akan menampilkan pesan peringatan.

Setelah itu, program mencoba mengubah kedua input menjadi bilangan bulat. Jika terjadi kesalahan karena input bukan angka, maka program menampilkan pesan error.

Apabila input valid, program membaca operator yang dipilih dan melakukan proses perbandingan sesuai operator tersebut. Misalnya, jika pengguna memilih  $>$ , program membandingkan apakah bilangan pertama lebih besar dari bilangan kedua. Hal yang sama berlaku untuk operator lainnya. Hasil dari perbandingan tersebut berupa nilai boolean `true` atau `false`, kemudian ditampilkan pada textfield hasil. Dengan demikian, logika utama program adalah *validasi input*  $\rightarrow$  *konversi angka*  $\rightarrow$  *menentukan operator*  $\rightarrow$  *melakukan perbandingan*  $\rightarrow$  *menampilkan hasil*. Program ini membantu pengguna memahami cara kerja operator relasional dalam pemrograman dengan menunjukkan hasil perbandingan secara langsung.

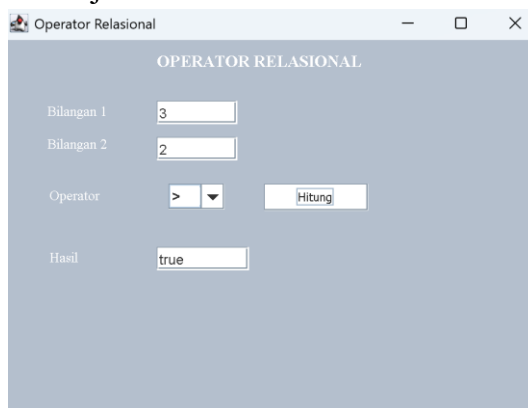
#### 4. Output

- Awal



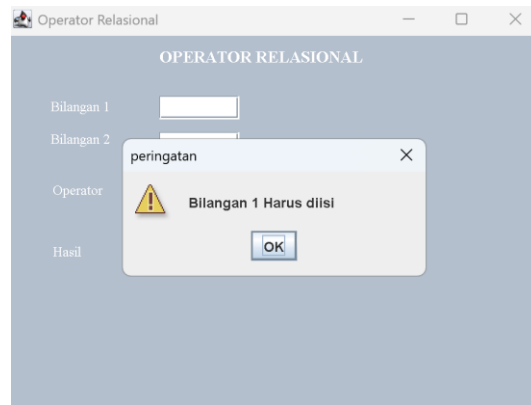
The screenshot shows a window titled "Operator Relasional". Inside, the title "OPERATOR RELASIONAL" is centered. There are four input fields: "Bilangan 1", "Bilangan 2", "Operator", and "Hasil". The "Operator" field has a dropdown menu currently showing ">". To the right of the "Operator" field is a button labeled "Hitung". The "Hasil" field is empty.

- Jika dijalankan

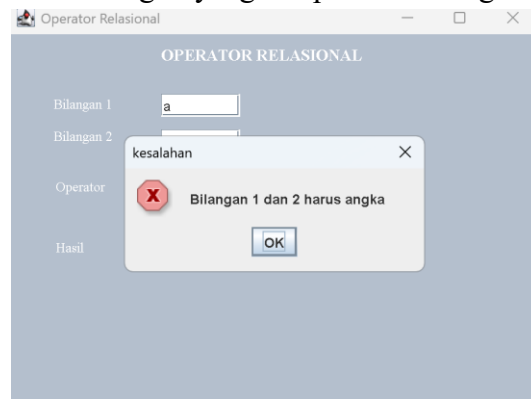


The screenshot shows the same window after execution. The "Bilangan 1" field now contains the number "3" and the "Bilangan 2" field contains the number "2". The "Operator" dropdown still shows ">". The "Hasil" field now contains the boolean value "true". The "Hitung" button remains visible.

- Jika salah satu bilangan kosong



- Jika Bilangan yang diinput bukan angka



## 5. Pseudocode

