

LAPORAN TUGAS ALGORITMA PEMROGRAMAN
OPERATOR RELASIONAL PADA WINDOW BUILDER TUGAS PEKAN 8



Oleh: Afif Naufal Zahran

NIM: 2511533009

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

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

DAFTAR ISI

BAB I PENDAHULUAN
1.1 Latar Belakang
1.2 Tujuan
1.3 Manfaat Praktikum.....
BAB II PEMBAHASAN
2,1 Penjelasan Singkat
2,2 Penjelasan Program.....
2,3 FlowChart, Pseudocode dan kode java.....
BAB III KESIMPULAN.....
DAFTAR PUSTAKA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Window Builder merupakan *plugin* yang membantu pengguna untuk membangun aplikasi GUI dengan mudah. Pada pertemuan pekan 8 saya telah mempelajari hal-hal dasar mengenai *Window Builder* seperti *JButton*, *JPanel*, dan sebagainya lalu membuat program operator aritmatika dengan menggunakan GUI. Pada penugasan kali ini penulis ditugaskan untuk membuat program yang serupa dengan operator relasional

.

1.2 Tujuan

Tujuan dari laporan tugas praktikum ini agar pembaca dan penulis dapat mendalami dan memahami tentang *Window Builder* dan implementasinya di bahasa pemrograman *Java*.

1.3 Manfaat

Manfaat dari tugas praktikum kali ini agar pembaca mendapatkan pengetahuan tentang *Window Builder*. Manfaat untuk penulis agar dapat memahami penggunaan *Window Builder* pada bahasa pemrograman *Java*.

BAB II

PEMBAHASAN

2,1 Penjelasan Singkat

Window Builder merupakan sebuah *plugin* yang memudahkan pengguna untuk mengembangkan program GUI (*Graphical User Interface*). Pada *Window Builder* pengguna dapat menerima *input*, membuat tombol dan sebagainya. Pada pertemuan pekan ini saya akan membuat program operator relasional dimana terdapat dua bilangan yang di-*inputkan* pengguna dan akan dibandingkan dengan operator yang juga di-*inputkan* oleh pengguna dan menghasilkan hasil antara *true* dan *false*.

2,2 Penjelasan Program

Pada program kali ini terdapat 1 class yakni “*tugasAlproPekan8_2511533009.java*” Untuk *helper method* terdapat dua *helper method* yakni “*pesanPeringatan()*” untuk memberikan pesan peringatan jika bilangan pertama atau kedua kosong dan ‘*PesanError()*’ untuk memberikan pesan error jika bilangan pertama atau kedua terdapat *character* selain angka

```
18 public class tugasAlproPekan8_2511533009 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      * Launch the application.
28      */
29     private void pesanPeringatan(String pesan) {
30         JOptionPane.showMessageDialog(this, pesan,"Peringatan",JOptionPane.WARNING_MESSAGE);
31     }
32
33     private void PesanEror(String pesan) {
34         JOptionPane.showMessageDialog(this, pesan,"Kesalahan",JOptionPane.ERROR_MESSAGE);
35     }
```

Gambar 2.1

2,3 Flowchart, Pseudocode Dan Kode Java

Berikut kode java dari program tugas kali ini.

```

36●    public static void main(String[] args) {
37●        EventQueue.invokeLater(new Runnable() {
38●            public void run() {
39●                try {
40                    tugasAlproPekan8_2511533009 frame = new tugasAlproPekan8_2511533009();
41                    frame.setVisible(true);
42                } catch (Exception e) {
43                    e.printStackTrace();
44                }
45            }
46        });
47    }
48
49● /**
50 * Create the frame.
51 */
52 public tugasAlproPekan8_2511533009() {
53     setTitle("Operator Relasional");
54     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
55     setBounds(100, 100, 450, 300);
56     contentPane = new JPanel();
57     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
58     setContentPane(contentPane);
59     contentPane.setLayout(null);
60
61     JLabel lblNewLabel = new JLabel("OPERATOR RELASIONAL");
62     lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
63     lblNewLabel.setBounds(61, 10, 216, 12);
64     contentPane.add(lblNewLabel);
65
66     JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
67     lblNewLabel_1.setBounds(10, 41, 124, 12);
68     contentPane.add(lblNewLabel_1);
69
70     JLabel lblNewLabel_2 = new JLabel("Bilangan 2");

```

Gambar 2.2

```

71     JLabel lblNewLabel_2 = new JLabel("Bilangan 2");
72     lblNewLabel_2.setBounds(10, 70, 93, 12);
73     contentPane.add(lblNewLabel_2);
74
75     JLabel lblNewLabel_3 = new JLabel("Operator");
76     lblNewLabel_3.setBounds(10, 98, 59, 12);
77     contentPane.add(lblNewLabel_3);
78
79     JLabel lblNewLabel_4 = new JLabel("Hasil");
80     lblNewLabel_4.setBounds(10, 138, 93, 12);
81     contentPane.add(lblNewLabel_4);
82
83     txtBill1 = new JTextField();
84     txtBill1.setHorizontalAlignment(SwingConstants.CENTER);
85     txtBill1.setBounds(71, 38, 96, 18);
86     contentPane.add(txtBill1);
87     txtBill1.setColumns(10);
88
89     txtBill2 = new JTextField();
90     txtBill2.setHorizontalAlignment(SwingConstants.CENTER);
91     txtBill2.setColumns(10);
92     txtBill2.setBounds(71, 67, 96, 18);
93     contentPane.add(txtBill2);
94
95     txtHasil = new JTextField();
96     txtHasil.setEditable(false);
97     txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
98     txtHasil.setColumns(10);
99     txtHasil.setBounds(71, 135, 96, 18);
100    contentPane.add(txtHasil);
101
102    JComboBox cbOperator = new JComboBox();
103    cbOperator.setModel(new DefaultComboBoxModel(new String[] {">", "<", ">=", "<=", "==" , "!="}));
104    cbOperator.setBounds(71, 94, 52, 20);
105    contentPane.add(cbOperator);

```

Gambar 2.3

```

106
107     JButton btnNewButton = new JButton("Bandingkan");
108     btnNewButton.addActionListener(new ActionListener() {
109         boolean hasil;
110         public void actionPerformed(ActionEvent e) {
111             if(txtBil1.getText().trim().isEmpty()) {
112                 pesanPeringatan("Bilangan 1 harus diisi");int a= Integer.valueOf(txtBil1.getText());
113             } else if(txtBil2.getText().trim().isEmpty()) {
114                 pesanPeringatan("Bilangan 2 harus diisi");
115             } else []
116             try {
117                 int a= Integer.valueOf(txtBil1.getText());
118                 int b= Integer.valueOf(txtBil2.getText());
119                 int c= cbOperator.getSelectedIndex();
120                 if(c==0) {
121                     hasil=a>b;
122                 }
123                 if(c==1) {
124                     hasil = a<b;
125                 }
126                 if (c==2) {
127                     hasil = a>=b;
128                 }
129                 if (c==3) {
130                     hasil = a<=b;
131                 }
132                 if (c==4) {
133                     hasil = a==b;
134                 }
135                 if (c==5) {
136                     hasil = a!=b;
137                 }
138                 txtHasil.setText(String.valueOf(hasil));
139             }catch(NumberFormatException ex) {
140                 PesanError("Bilangan 1 dan Bilangan 2 harus angka");
141             }

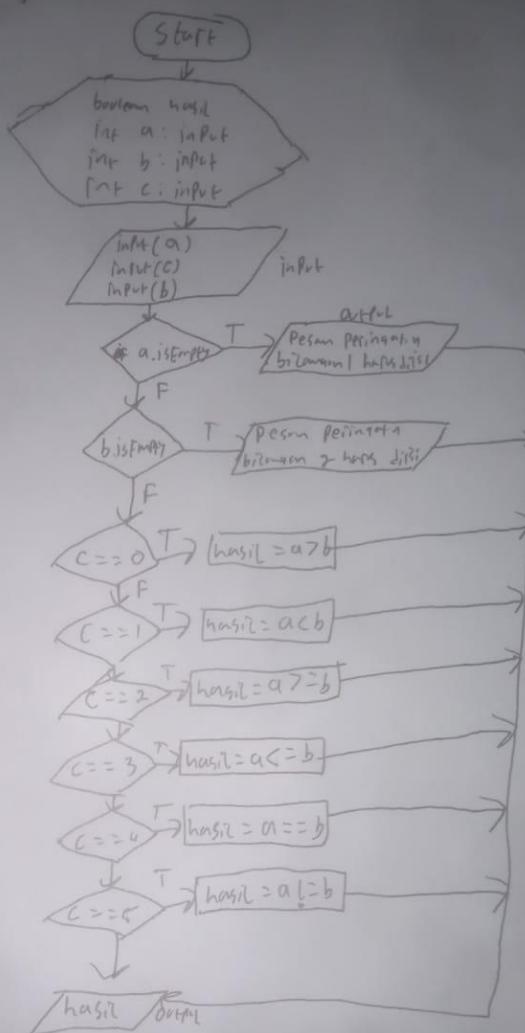
```

Gambar 2.4

Pada program kali ini kita akan berfokus pada bagian Gambar 2.4 karena bagian tersebut algoritma yang dibuat. Pertama program akan melakukan validasi apakah input pertama atau kedua kosong jika ya akan memberikan pesan peringatan, lalu setelah itu program juga akan menggunakan *try* dan *catch* untuk *exception* “*NumberFormatException*” agar memastikan jika masukan yang di *input* oleh pengguna adalah angka, setelah itu dilakukan pengecekan jenis operator apa yang dipilih oleh pengguna agar program dapat membandingkan dengan menggunakan operator yang sesuai oleh pilihan pengguna.

a. FlowChart

Afif Maulid Zanthy
2011633009



b. PseudoCode

JUDUL
Program Operator Relasional
DEKLARASI
VAR hasil:Boolean

```
VAR a:Imteger
VAR b:Integer
VAR c: Integer

INPUT(a,b,c)

If (c == 0) then hasil = a>b
If (c == 1) then hasil = a < b
If (c == 2) then hasil = a >= b
If (c == 3) then hasil = a <= b
If (c == 4) then hasil = a == b
If (c == 5) then hasil = a != b

PRINT(hasil)
```

BAB III

KESIMPULAN

Kesimpulan dari praktikum kali ini adalah Window Builder pada bahasa pemograman *Java* cukup penting karena dapat digunakan untuk membuat program GUI dengan java.

DAFTAR PUSTAKA