

Razif Al Farisi

2511532028

IF Praktikum B

Laporan Tugas Pekan 8

Buatlah program 'Operasi Logika' dengan menggunakan GUI pada bahasa pemrograman java.

1. Pseudocode

Judul :

Operasi Logika dengan GUI

Deklarasi :

texta, textb, pesanPeringatan, pesanError : String
a1(texta), a2(textb), hasil = false : Boolean
z : Integer

Pseudocode :

If (a1 == null)

 Print pesanPeringatan

Else If (a2 == null)

 Print pesanPeringatan

EndIf

Input a1

Input a2

If (a1 != true, a1 != false)

 Print pesanError

Else If (a2 != true, a2 != false)

 Print pesanError

EndIf

If (z==0)

 hasil = a1 && a2

EndIf

If (z==1)

 hasil = a1 || a2

EndIf

If (z==2)

 hasil = !a1

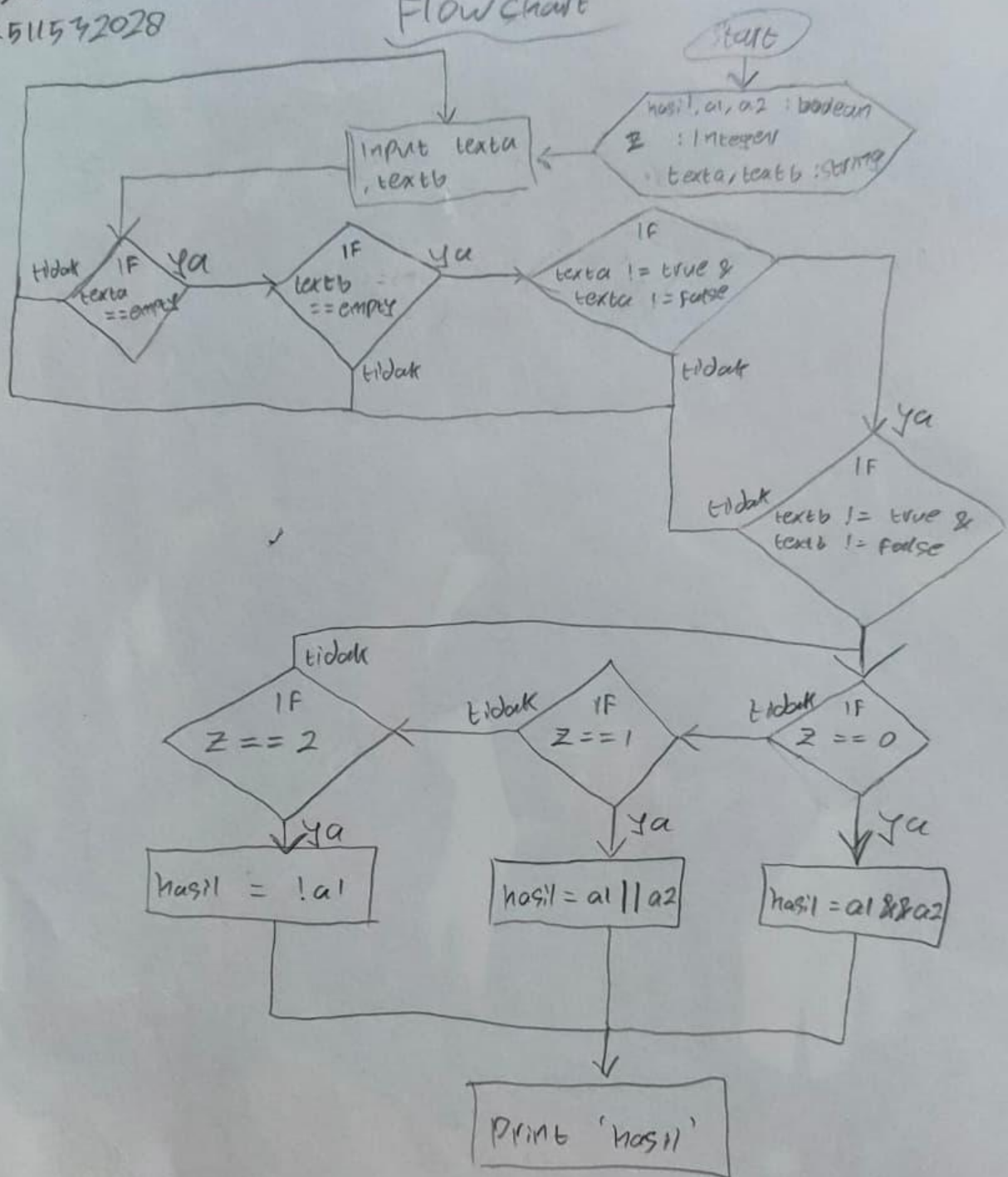
EndIf

Print hasil

2. Flowchart :

Razif Al Fawisi
2511532028

IF Prakteikum B Flowchart



3. Kode Program :

```
4. package pekan8_2511532028;
5.
6. import java.awt.EventQueue;
7.
8. import javax.swing.JFrame;
9. import javax.swing.JPanel;
10. import javax.swing.border.EmptyBorder;
11. import javax.swing.JLabel;
12. import javax.swing.JOptionPane;
13. import javax.swing.SwingConstants;
14. import java.awt.Font;
15. import javax.swing.JTextField;
16. import javax.swing.JComboBox;
17. import javax.swing.JButton;
18. import javax.swing.DefaultComboBoxModel;
19. import java.awt.event.ActionListener;
20. import java.awt.event.ActionEvent;
21.
22. public class OperasiLogikaGUI_2511532028 extends JFrame {
23.
24.     private static final long serialVersionUID = 1L;
25.     private JPanel contentPane;
26.     private JTextField nilai1;
27.     private JTextField nilai2;
28.     private JTextField hasil1;
29.
30.     private void pesanPeringatan(String pesan) {
31.         JOptionPane.showMessageDialog(this, pesan, "Peringatan",
32.             JOptionPane.WARNING_MESSAGE);
33.     }
34.     private void pesanError(String pesan) {
35.         JOptionPane.showMessageDialog(this, pesan, "Kesalahan",
36.             JOptionPane.ERROR_MESSAGE);
37.     }
38.
39.     /**
40.      * Launch the application.
41.      */
42.     public static void main(String[] args) {
43.         EventQueue.invokeLater(new Runnable() {
44.             public void run() {
45.                 try {
46.                     OperasiLogikaGUI_2511532028 frame = new
47. OperasiLogikaGUI_2511532028();
48.                     frame.setVisible(true);
49.                 } catch (Exception e) {
50.                     e.printStackTrace();
51.                 }
52.             }
53.         });
54.     }
55.
56.     /**
57.      * Create the frame.
58.      */
59.     public OperasiLogikaGUI_2511532028() {
60.         setTitle("OperasiLogika");
61.         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
62.         setBounds(100, 100, 450, 300);
63.     }
64. }
```

```

60.     contentPane = new JPanel();
61.     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
62.     setContentPane(contentPane);
63.     contentPane.setLayout(null);
64.
65.     JLabel lblNewLabel = new JLabel("Operasi Logika");
66.     lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 14));
67.     lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
68.     lblNewLabel.setBounds(154, 11, 125, 34);
69.     contentPane.add(lblNewLabel);
70.
71.     JLabel lblNewLabel_1 = new JLabel("Input Nilai ke-1");
72.     lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
73.     lblNewLabel_1.setBounds(34, 59, 86, 14);
74.     contentPane.add(lblNewLabel_1);
75.
76.     JLabel lblNewLabel_2 = new JLabel("Input Nilai ke-2");
77.     lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
78.     lblNewLabel_2.setBounds(302, 59, 86, 14);
79.     contentPane.add(lblNewLabel_2);
80.
81.     nilai1 = new JTextField();
82.     nilai1.setBounds(34, 86, 86, 20);
83.     contentPane.add(nilai1);
84.     nilai1.setColumns(10);
85.
86.     nilai2 = new JTextField();
87.     nilai2.setBounds(302, 86, 86, 20);
88.     contentPane.add(nilai2);
89.     nilai2.setColumns(10);
90.
91.     JComboBox Operator = new JComboBox();
92.     Operator.setModel(new DefaultComboBoxModel(new String[] {"AND", "OR",
"NEGASI"}));
93.     Operator.setBounds(172, 139, 86, 22);
94.     contentPane.add(Operator);
95.
96.     hasil1 = new JTextField();
97.     hasil1.setBounds(172, 203, 86, 20);
98.     contentPane.add(hasil1);
99.     hasil1.setColumns(10);
100.
101.     JLabel lblNewLabel_3 = new JLabel("Hasil");
102.     lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
103.     lblNewLabel_3.setBounds(172, 178, 86, 14);
104.     contentPane.add(lblNewLabel_3);
105.
106.     JLabel lblNewLabel_4 = new JLabel("Operator");
107.     lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
108.     lblNewLabel_4.setBounds(175, 114, 83, 14);
109.     contentPane.add(lblNewLabel_4);
110.
111.     JButton btnNewButton = new JButton("process");
112.     btnNewButton.addActionListener(new ActionListener() {
113.         public void actionPerformed(ActionEvent e) {
114.             boolean hasil = false;
115.             if (nilai1.getText().trim().isEmpty()) {
116.                 pesanPeringatan("nilai 1 harus di isi");
117.             } else
118.             if (nilai2.getText().trim().isEmpty()) {
119.                 pesanPeringatan("nilai 2 harus di isi");

```

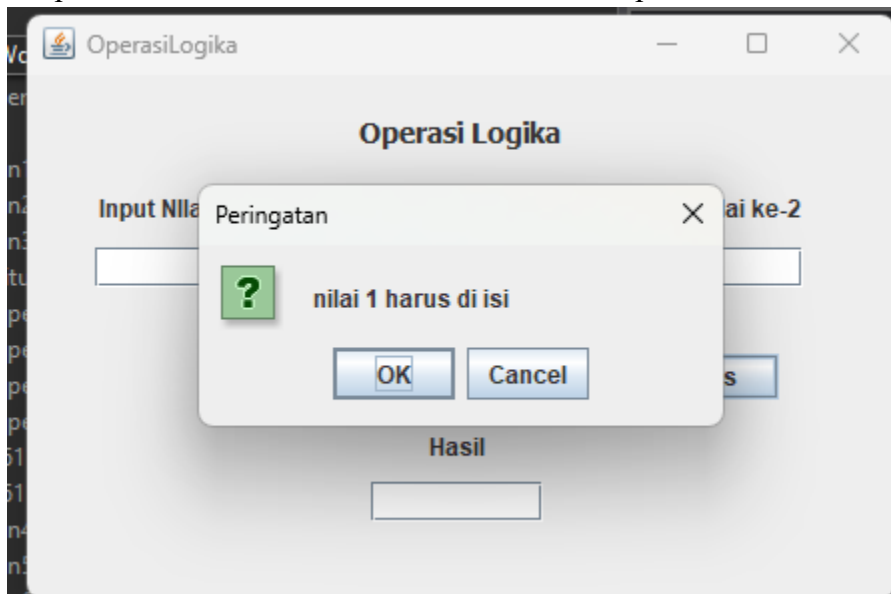
```

120.         }
121.         try {
122.             String texta = nilai1.getText().trim();
123.             String textb = nilai2.getText().trim();
124.             boolean a1 = Boolean.parseBoolean(texta);
125.             boolean a2 = Boolean.parseBoolean(textb);
126.             if (!(texta.equalsIgnoreCase("true") ||
texta.equalsIgnoreCase("false"))) {
127.                 pesanError("nilai 1 harus di isi true atau false");
128.             } else
129.                 if (!(textb.equalsIgnoreCase("true") ||
textb.equalsIgnoreCase("false"))) {
130.                     pesanError("nilai 2 harus di isi true atau false");
131.                 }
132.                 int z = Operator.getSelectedIndex();
133.                 if (z==0) {hasil = a1 && a2;}
134.                 if (z==1) {hasil = a1 || a2;}
135.                 if (z==2) {hasil = !a1;}
136.             }
137.             catch (Exception ex) {
138.                 pesanError("nilai 1 dan 2 harus berupa true atau false");
139.             }
140.             hasil1.setText(String.valueOf(hasil));
141.         }
142.     });
143.     btnNewButton.setBounds(287, 139, 89, 23);
144.     contentPane.add(btnNewButton);
145.
146.
147.     }
148. }

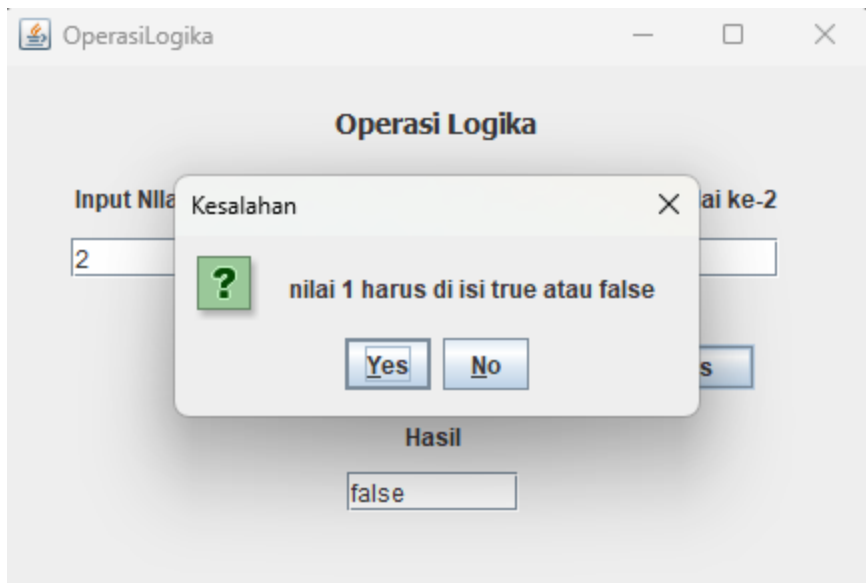
```

Output :

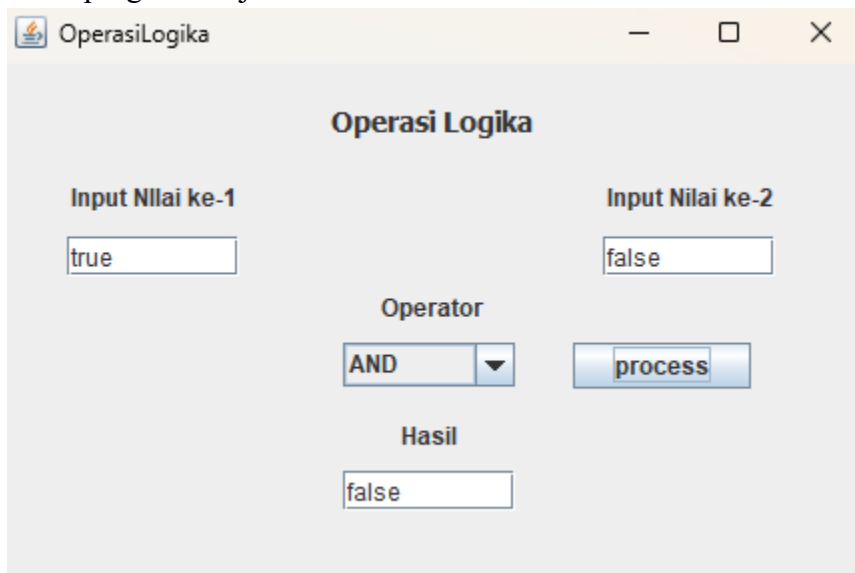
Output error Ketika nilai 1 atau nilai 2 tidak di input :



Output error ketika inputan nilai 1 atau nilai 2 tidak sama dengan true/false :



Kode program berjalan lancar :



Penjelasan :

Program ini memuat logika tentang penggunaan konjung, disjungsi, dan negasi. Jika memilih operator konjungsi(AND), maka outputnya akan bernilai false kecuali a1 bernilai “true” dan a2 bernilai “ true”. Jika memilih operator disjungsi(OR), maka outputnya akan bernilai true kecuali a1 bernilai “false” dan a2 bernilai “false”. Jika kita memilih operator Negasi(!), maka output dari a1 adalah kebalikan dari nilai yang di input, begitu juga a2. Tetapi masih terdapat kekurangan di program di bagian negasi dan kesalahan error saat input selain “true” atau “false”. Negasi cuman bisa menampilkan a1, masih belum tau cara menampilkan kedua nilai tersebut ke GUI. Kesalahan error saat kita tidak menginputkan nilai pada a1 dan a2, pesan yang keluar ada 2, yaitu pesan saat a1 dan a2 tidak di isi/input dan pesan a1 dan a2 inputnya selain “true” dan “false”.