

TUGAS PEKAN 8 ALPRO

MEMBUAT PROGRAM GUI TENTANG OPERATOR LOGIKA

NAMA : Dennis Shauqi Akbar

NIM : 2511533020

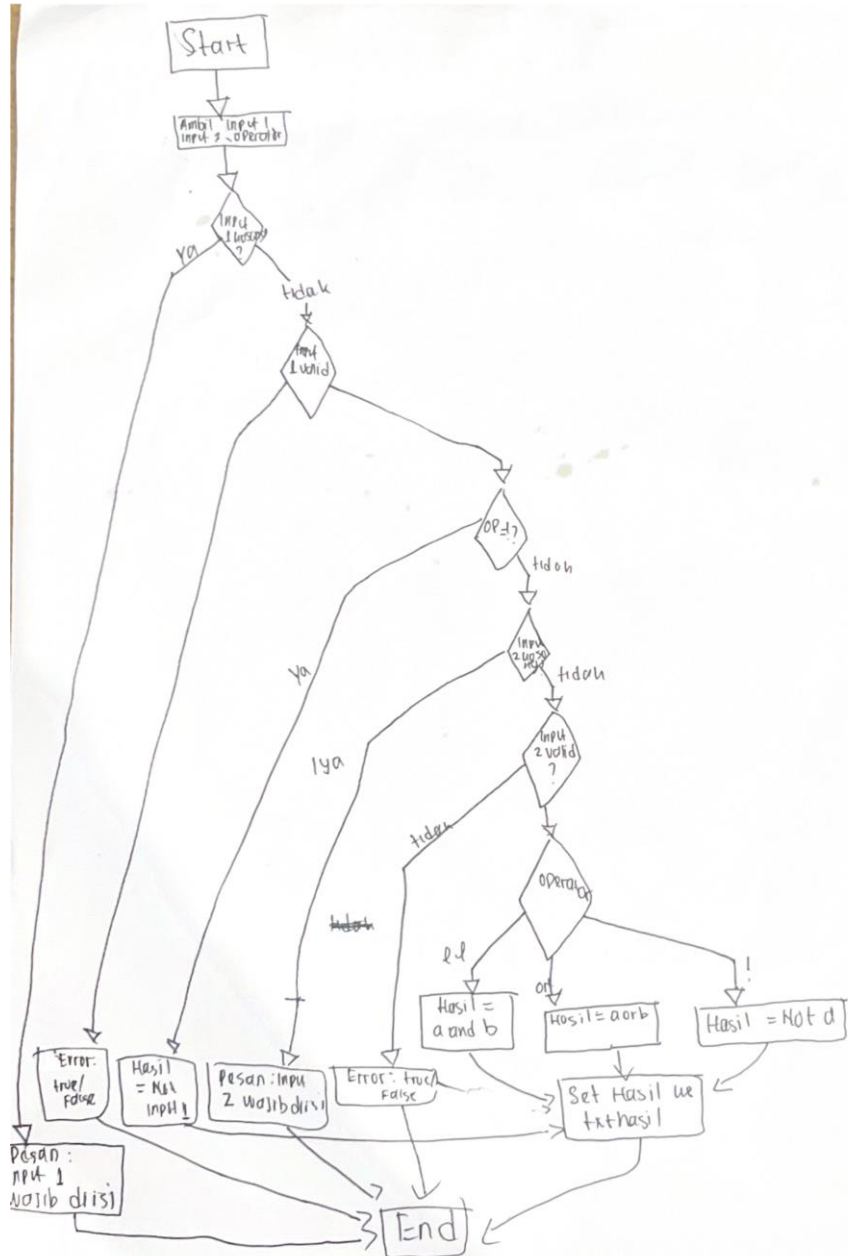
DOSEN PENGAMPU : Dr. Wahyudi, S.T, M.T



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

1. FLOWCHART



2. PSEUDOCODE

JUDUL	Program GUI Operator LOGika {Program ini digunakan untuk memproses operator logika AND, OR, dan NOT}
DEKLARASI	input1, input2 : string op : string a, b, hasil : boolean
ALGORITMA	<pre> input(input1) input1 ← lowercase(trim(input1)) jika input1 = "" maka tampilkan "Input 1 wajib diisi" selesai endif jika input1 ≠ "true" dan input1 ≠ "false" maka tampilkan "Input harus berupa true atau false" selesai endif input(op) jika op ≠ "!" maka input(input2) input2 ← lowercase(trim(input2)) jika input2 = "" maka tampilkan "Input 2 wajib diisi" selesai endif jika input2 ≠ "true" dan input2 ≠ "false" maka tampilkan "Input harus berupa true atau false" selesai endif endif a ← konversiBoolean(input1) b ← konversiBoolean(input2) hasil ← false pilih(op) kasus "&&" : hasil ← a ∧ b kasus " " : hasil ← a ∨ b kasus "!" : hasil ← ¬a </pre>

	selesai pilih output(hasil)
--	------------------------------------

3. KODE PROGRAM

```

1 package alprotugas;
2
3 import java.awt.BorderLayout;
21
22 public class TugasAlproPekan8_2511533020 extends JFrame {
23
24     private static final long serialVersionUID = 1L;
25     private JPanel contentPane;
26     private JTextField txtLogika1;
27     private JTextField txtLogika2;
28     private JTextField txtHasil;
29     private JLabel lblNewLabel_1;
30     private JLabel lblNewLabel_2;
31     private JLabel lblNewLabel_3;
32     private JLabel lblNewLabel_4;
33     private JComboBox CbOperator;
34     private JButton btnNewButton_1;
35
36     private void pesanPeringatan(String pesan) {
37         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE );
38     }
39     private void pesanError(String pesan) {
40         JOptionPane.showMessageDialog(this, pesan, "Error", JOptionPane.ERROR_MESSAGE );
41     }
42     /**
43      * Launch the application.
44      */
45     public static void main(String[] args) {
46         EventQueue.invokeLater(new Runnable() {
47             public void run() {
48                 try {
49                     TugasAlproPekan8_2511533020 frame = new TugasAlproPekan8_2511533020();
50                     frame.setVisible(true);
51                 } catch (Exception e) {
52                     e.printStackTrace();
53                 }

```

```

1 package alprotugas;
2
3 import java.awt.BorderLayout;
4
21
22 public class TugasAlproPekan8_2511533020 extends JFrame {
23
24     private static final long serialVersionUID = 1L;
25     private JPanel contentPane;
26     private JTextField txtLogika1;
27     private JTextField txtLogika2;
28     private JTextField txtHasil;
29     private JLabel lblNewLabel_1;
30     private JLabel lblNewLabel_2;
31     private JLabel lblNewLabel_3;
32     private JLabel lblNewLabel_4;
33     private JComboBox CbOperator;
34     private JButton btnNewButton_1;
35
36     private void pesanPeringatan(String pesan) {
37         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE );
38     }
39     private void pesanError(String pesan) {
40         JOptionPane.showMessageDialog(this, pesan, "Error", JOptionPane.ERROR_MESSAGE );
41     }
42     /**
43      * Launch the application.
44      */
45     public static void main(String[] args) {
46         EventQueue.invokeLater(new Runnable() {
47             public void run() {
48                 try {
49                     TugasAlproPekan8_2511533020 frame = new TugasAlproPekan8_2511533020();
50                     frame.setVisible(true);
51                 } catch (Exception e) {
52                     e.printStackTrace();
53                 }

```

```

54             }
55         });
56     }
57
58     /**
59      * Create the frame.
60      */
61     public TugasAlproPekan8_2511533020() {
62         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
63         setBounds(100, 100, 450, 300);
64         contentPane = new JPanel();
65         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
66         setContentPane(contentPane);
67         contentPane.setLayout(null);
68
69         txtLogika1 = new JTextField();
70         txtLogika1.setHorizontalAlignment(SwingConstants.CENTER);
71         txtLogika1.setBounds(197, 52, 58, 17);
72         contentPane.add(txtLogika1);
73         txtLogika1.setColumns(10);
74
75         JLabel lblNewLabel = new JLabel("OPERATOR LOGIKA");
76         lblNewLabel.setBounds(143, 11, 152, 17);
77         lblNewLabel.setFont(new Font("Times New Roman", Font.BOLD, 14));
78         lblNewLabel.setHorizontalAlignment(SwingConstants.LEFT);
79         contentPane.add(lblNewLabel);
80
81
82         txtLogika2 = new JTextField();
83         txtLogika2.setHorizontalAlignment(SwingConstants.CENTER);
84         txtLogika2.setColumns(10);
85         txtLogika2.setBounds(197, 88, 58, 17);
86         contentPane.add(txtLogika2);
87
88         lblNewLabel_4 = new JLabel("HASIL");
89         lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);

```

```

90     lblNewLabel_4.setBounds(10, 206, 163, 14);
91     contentPane.add(lblNewLabel_4);
92
93     txtHasil = new JTextField();
94     txtHasil.setEditable(false);
95     txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
96     txtHasil.setColumns(10);
97     txtHasil.setBounds(197, 204, 58, 17);
98     contentPane.add(txtHasil);
99
100    CbOperator = new JComboBox();
101    CbOperator.setModel(new DefaultComboBoxModel(new String[] {"&&", "|", "!"}));
102    CbOperator.setBounds(197, 141, 58, 22);
103    contentPane.add(CbOperator);
104
105
106    lblNewLabel_1 = new JLabel("Input 1 (true/false)");
107    lblNewLabel_1.setHorizontalAlignment(SwingConstants.CENTER);
108    lblNewLabel_1.setBounds(23, 54, 136, 14);
109    contentPane.add(lblNewLabel_1);
110
111    lblNewLabel_2 = new JLabel("Input 2 (true/false)");
112    lblNewLabel_2.setHorizontalAlignment(SwingConstants.CENTER);
113    lblNewLabel_2.setBounds(23, 90, 136, 14);
114    contentPane.add(lblNewLabel_2);
115
116    lblNewLabel_3 = new JLabel("OPERATOR");
117    lblNewLabel_3.setHorizontalAlignment(SwingConstants.CENTER);
118    lblNewLabel_3.setBounds(23, 145, 136, 14);
119    contentPane.add(lblNewLabel_3);
120
121    lblNewLabel_4 = new JLabel("HASIL");
122    lblNewLabel_4.setHorizontalAlignment(SwingConstants.CENTER);
123    lblNewLabel_4.setBounds(23, 206, 136, 14);
124    contentPane.add(lblNewLabel_4);
125
126    btnNewButton_1 = new JButton("OPERASIKAN");
127    btnNewButton_1.addActionListener(new ActionListener() {
128        public void actionPerformed(ActionEvent e) {
129            String input1 = txtLogika1.getText().trim().toLowerCase();
130            String input2 = txtLogika2.getText().trim().toLowerCase();
131            String op = CbOperator.getSelectedItem().toString();
132
133
134            if (input1.isEmpty()) {
135                pesanPeringatan("Input 1 wajib diisi");
136                return;
137            }
138
139
140            if (!input1.equals("true") && !input1.equals("false")) {
141                pesanError("Input harus berupa true atau false");
142                return;
143            }
144
145
146            if (!op.equals("!")) {
147
148
149                if (input2.isEmpty()) {
150                    pesanPeringatan("Input 2 wajib diisi");
151                    return;
152                }
153
154
155                if (!input2.equals("true") && !input2.equals("false")) {
156                    pesanError("Input harus berupa true atau false");
157                    return;
158                }
159            }
160

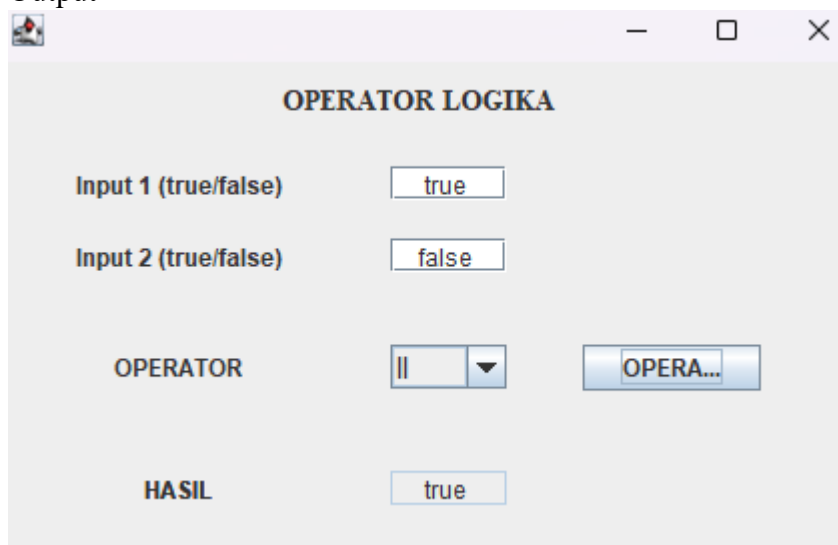
```

```

161
162     boolean a = Boolean.parseBoolean(input1);
163     boolean b = Boolean.parseBoolean(input2);
164     boolean hasil = false;
165
166     switch (op) {
167         case "&&":
168             hasil = a && b;
169             break;
170
171         case "||":
172             hasil = a || b;
173             break;
174
175         case "!":
176             hasil = !a;
177             break;
178     }
179
180     txtHasil.setText(String.valueOf(hasil));
181 }
182
183 });
184 btnNewButton_1.setBounds(293, 141, 89, 23);
185 contentPane.add(btnNewButton_1);
186
187
188 }
189 }
190

```

4. Output



OPERATOR LOGIKA

Input 1 (true/false)

Input 2 (true/false)

OPERATOR ▼

HASIL