

LAPORAN PRAKTIKUM
PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

“Pemrograman GUI 1”

disusun oleh:

Dennis Shauqi Akbar

2511533020

Dosen pengampu: Dr. Wahyudi. S.T.M.T

Asisten Pratikum: Rahmad DRO



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan tugas kelompok ini dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, keluarga, sahabat, serta para pengikutnya hingga akhir zaman.

Tugas Individu ini kami susun dengan tujuan untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, dengan Judul “ Perulangan while dan do=while” yang mencakup Lempar dadu, Game Penjumlahan, Sentinel Loop, doWhile

Kami menyadari bahwa penyusunan tugas ini masih jauh dari sempurna, baik dari segi materi maupun penyajiannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pihak demi kesempurnaan tugas kami di masa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada dosen pembimbing yang telah memberikan arahan sehingga tugas ini dapat terselesaikan tepat waktu. Semoga tugas ini dapat bermanfaat bagi kami khususnya, dan bagi pembaca pada umumnya.

Padang, 23 November 2025

Dennis Shauqi Akbar

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	2
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN	3
2.1 Program GUI Operator Logika	3
BAB III KESIMPULAN.....	14
3.1 Kesimpulan	14
3.2 Saran Pengembangan	14
DAFTAR PUSTAKA	14

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak, Java dikenal sebagai salah satu bahasa pemrograman yang stabil, aman, dan banyak digunakan karena sifatnya yang dapat dijalankan di berbagai platform. Selain kemampuan logika dan perhitungan yang dilakukan di balik layar, sebuah aplikasi juga perlu memiliki tampilan yang memudahkan pengguna untuk berinteraksi. Karena itulah, pemrograman Graphical User Interface (GUI) menjadi aspek penting yang harus dipahami oleh mahasiswa informatika maupun bidang terkait.

Java menyediakan berbagai pustaka yang mempermudah pembuatan GUI, seperti AWT, Swing, dan JavaFX. Melalui pustaka tersebut, pengembang bisa membuat elemen-elemen visual seperti tombol, label, kotak teks, hingga jendela aplikasi yang lengkap. Kehadiran GUI membuat aplikasi menjadi lebih intuitif, tidak lagi bergantung pada input dan output berbasis teks, sehingga pengalaman pengguna menjadi jauh lebih nyaman.

Di lingkungan pendidikan, mempelajari pemrograman GUI tidak hanya membantu mahasiswa memahami cara membuat tampilan yang menarik, tetapi juga melatih mereka menghubungkan antara desain tampilan dan logika program. Keterampilan ini menjadi pondasi penting untuk mengembangkan aplikasi desktop yang lebih besar dan kompleks di masa depan.

Melalui kegiatan praktikum, mahasiswa diharapkan dapat memahami cara kerja komponen GUI, proses penanganan event, dan bagaimana logika program diintegrasikan dengan antarmuka. Dengan penguasaan dasar-dasar ini, mahasiswa memiliki bekal yang kuat untuk mengembangkan perangkat lunak yang interaktif dan bernilai guna dalam dunia nyata.

1.2 Tujuan

1. Memahami konsep dasar Graphical User Interface (GUI) pada Java, termasuk fungsi dan peran GUI dalam sebuah aplikasi.
2. Mengenal serta menggunakan komponen–komponen GUI seperti tombol (button), label, text field, combo box, dan komponen lainnya yang disediakan oleh Java Swing atau pustaka GUI terkait.
3. Mempelajari cara menangani event (event handling) seperti aksi klik tombol, input pengguna, dan perubahan nilai pada komponen GUI.

1.3 Manfaat Praktikum

1. Meningkatkan pemahaman mahasiswa terhadap konsep antarmuka grafis, sehingga mampu membedakan antara program berbasis teks dan program berbasis GUI.
2. Melatih kemampuan menggunakan komponen–komponen GUI seperti tombol, label, kotak teks, dan lainnya, sehingga mahasiswa lebih memahami cara membangun tampilan aplikasi yang interaktif.
3. Membantu mahasiswa memahami event handling, yaitu bagaimana aplikasi merespons aksi pengguna seperti klik tombol atau perubahan

BAB II

PEMBAHASAN

2.1 Program GUI Operator Logika

```
1 package alprotugas;
2
30 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                 }
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);
```

```
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);

```

```

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

```

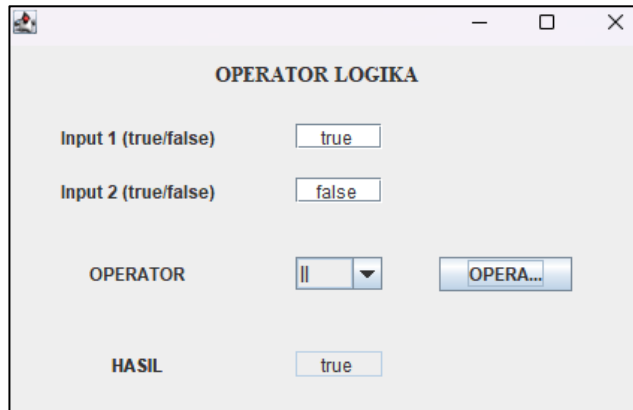
Kode Program 2.1

Program ini merupakan aplikasi sederhana berbasis GUI menggunakan Java Swing yang berfungsi untuk melakukan operasi logika seperti AND (&&), OR (||), dan NOT (!). Ketika tombol *Operasikan* ditekan, program terlebih dahulu mengambil kedua input dari pengguna, kemudian merapikan teks dengan menghapus spasi yang tidak perlu dan mengubahnya menjadi huruf kecil agar mudah divalidasi. Setelah itu, program memastikan bahwa Input 1 tidak boleh kosong dan harus bernilai “true” atau “false”. Jika operator yang digunakan bukan tanda negasi “!”, maka Input 2 juga harus diisi dan wajib berisi nilai boolean yang valid. Jika ada input yang tidak sesuai, program langsung menampilkan pesan peringatan atau error dan menghentikan proses.

Apabila semua input sudah benar, program mengubah nilai teks tersebut menjadi tipe data boolean dan menjalankan operasi logika sesuai operator yang dipilih melalui struktur switch-case. Operator “&&” mengevaluasi keduanya harus bernilai true, “||” memeriksa apakah minimal salah satunya true, sedangkan “!” membalik nilai dari input pertama. Setelah operasi selesai dihitung, hasilnya ditampilkan kembali di kotak output. Dengan alur yang terstruktur dan validasi

yang ketat, program ini memastikan bahwa setiap proses berjalan dengan aman dan hasil yang diberikan sesuai dengan aturan logika boolean.

Output dari program diatas jika input 1 : true dan input 2 : false dan memakai operator or lalu dijalankan adalah



The screenshot shows a Java Swing window titled "OPERATOR LOGIKA". It contains the following elements:

- Input 1 (true/false):** A text field containing the value "true".
- Input 2 (true/false):** A text field containing the value "false".
- OPERATOR:** A dropdown menu showing the logical OR operator "||".
- OPERA...:** A button to execute the operation.
- HASIL:** A text field showing the result "true".

Gambar 2.1

BAB III

KESIMPULAN

3.1 Kesimpulan

Melalui praktikum ini dapat dipahami bahwa pembuatan GUI di Java mampu menghadirkan pengalaman penggunaan yang lebih nyaman dan mudah dipahami dibandingkan program berbasis teks. Dengan memanfaatkan komponen-komponen dari Java Swing, seperti tombol, kotak input, dan pilihan operator, program bisa berinteraksi langsung dengan pengguna secara visual. Pada contoh aplikasi operasi logika yang dibuat, penerapan validasi input memastikan bahwa hanya nilai boolean yang benar yang diproses, sehingga program berjalan lebih aman dan bebas dari kesalahan pengguna.

Penggunaan struktur if, switch-case, serta konversi tipe data dalam proses logika membuat alur kerja program lebih jelas dan terarah. Secara keseluruhan, praktikum ini memberikan gambaran nyata tentang bagaimana antarmuka pengguna, pengecekan data, dan proses logika dapat digabungkan menjadi sebuah aplikasi kecil yang fungsional dan mudah digunakan dalam kehidupan sehari-hari.

3.2 Saran Pengembangan

Ke depan, program operasi logika berbasis GUI ini masih dapat dikembangkan menjadi lebih baik lagi. Proses validasi input, misalnya, bisa dibuat lebih cerdas agar pengguna mendapatkan pesan kesalahan yang lebih jelas dan membantu. Fitur program juga dapat diperluas—seperti menambahkan jenis operasi logika lainnya, misalnya XOR atau NAND—agar aplikasi tidak hanya terbatas pada tiga operator dasar.

Dari segi tampilan, antarmuka bisa dibuat lebih menarik dan modern, baik dengan memperbaiki tata letak, menambahkan gaya visual tertentu, maupun dengan mempertimbangkan penggunaan JavaFX sebagai alternatif. Untuk meminimalkan kesalahan pengetikan, nilai boolean bahkan dapat diganti menjadi pilihan radio button sehingga pengguna tinggal memilih tanpa perlu mengetik. Fitur tambahan seperti tombol reset, penyimpanan riwayat, atau bahkan pengembangan ke versi

web dan mobile juga dapat membuat aplikasi ini lebih fleksibel dan mudah digunakan.

Secara keseluruhan, berbagai pengembangan ini ditujukan untuk membuat aplikasi lebih lengkap, ramah pengguna, dan lebih relevan untuk kebutuhan pembelajaran maupun pemakaian sehari-hari.

DAFTAR PUSTAKA

[1] K. H. Rosen, Discrete Mathematics and Its Applications, 7th ed. New York, NY, USA: McGraw-Hill, 2012.

[2] K. Ribas and J. Deitel, Java for Programmers, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2009.

[3] H. M. Deitel and P. J. Deitel, Java: How to Program, 11th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2017.

[4] Oracle, “Java Swing Tutorial,” Oracle Documentation, 2024.

<https://docs.oracle.com/javase/tutorial/uiswing/>

[5] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.