

**LAPORAN PRAKTIKUM ALGORITMA
PEMROGRAMAN
“PENERAPAN GRAPHICAL USER INTERFACE (GUI)
DALAM PEMROGRAMAN JAVA”**

DISUSUN OLEH:

NAYSHA APRILIA RIZKI

2511531014

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

**ASISTEN PRAKTIKUM: MUHAMMAD ZAKI AL
HAFIZ**



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

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Laporan praktikum ini dapat diselesaikan dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas dalam rangka memenuhi tugas yang diberikan oleh dosen dan juga asisten praktikum.

Padang, 23 November 2025

Naysha Aprilia Rizki

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk di bidang pendidikan, industri, dan komunikasi. Salah satu pilar utama dalam perkembangan teknologi tersebut adalah **pemrograman komputer**. Di antara banyak bahasa pemrograman yang ada, **Java** merupakan salah satu bahasa yang sangat populer dan banyak digunakan di berbagai belahan dunia.

Java adalah bahasa pemrograman tingkat tinggi yang bersifat **object-oriented**, dikembangkan oleh Sun Microsystems pada tahun 1995, dan kini dikelola oleh Oracle. Keunggulan utama Java terletak pada kemampuannya untuk berjalan di berbagai platform (platform-independent) berkat konsep **Write Once, Run Anywhere (WORA)** yang dimilikinya. Hal ini memungkinkan program yang dibuat dalam Java untuk dijalankan di berbagai sistem operasi tanpa perlu diubah ulang.

Selain itu, Java memiliki sintaks yang relatif mudah dipahami bagi pemula, dokumentasi yang lengkap, dan komunitas pengguna yang besar, sehingga sangat mendukung proses belajar dan pengembangan perangkat lunak. Java juga menjadi bahasa utama dalam pengembangan berbagai aplikasi, seperti aplikasi desktop, mobile (Android), web, hingga sistem enterprise berskala besar.

Dengan melihat pentingnya Java dalam dunia pemrograman dan aplikasinya yang luas, sangatlah relevan untuk mempelajari dasar-dasar bahasa pemrograman Java, baik dari segi teori maupun praktik. Oleh karena itu, laporan ini disusun untuk memberikan pemahaman yang lebih mendalam tentang bahasa Java, struktur dasarnya, serta penerapan konsep pemrograman berorientasi objek dalam bahasa tersebut

1.2 Tujuan

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik

1.3 Manfaat Pratikum

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bekal awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.
6. Memberikan pengalaman langsung dalam merancang dan mengimplementasikan komponen GUI pada aplikasi atau program Java.

BAB II

PEMBAHASAN

2.1 Pengertian Graphical User Interface (GUI)

Dalam bahasa pemrograman Java, Graphical User Interface (GUI) merupakan antarmuka yang memungkinkan user berinteraksi dengan aplikasi/ program melalui elemen visual seperti tombol, jendela, ikon, dan text box. Dalam pemrograman Java, GUI biasanya dibuat menggunakan Java Swing, yaitu library yang menyediakan berbagai komponen untuk menyusun tampilan aplikasi. Dengan adanya GUI, pengguna dapat melakukan input dan navigasi menggunakan komponen visual dengan cara merancang dan mengimplementasikan GUI menggunakan Java Swing serta memanfaatkan WindowBuilder, yaitu plugin visual editor untuk membuat tampilan i secara drag-and-drop di Eclipse. Pemahaman terhadap JFrame, komponen GUI, serta mekanisme event handling menjadi dasar penting dalam membangun aplikasi /program Java yang interaktif.

1. WindowBuilder

WindowBuilder adalah sebuah GUI designer untuk Java yang tersedia sebagai plugin pada IDE Eclipse. Dengan WindowBuilder, dapat merancang tampilan GUI secara visual menggunakan metode drag-and-drop, tanpa harus menulis semua kode secara manual. WindowBuilder mendukung pengembangan antarmuka menggunakan Swing, SWT, maupun GWT.

Berikut beberapa fungsi dari WindowBuilder:

1. Sinkronisasi otomatis antara editor visual dan kode.
2. Editor visual untuk mendesain layout.
3. Mempercepat proses pembuatan antarmuka Java 6

2. Java Swing

Java Swing adalah library GUI yang termasuk dalam Java Foundation Classes (JFC). Swing digunakan untuk membuat aplikasi desktop dengan tampilan yang dapat dikustomisasi. Library ini menyediakan banyak komponen GUI, mulai dari komponen dasar seperti tombol hingga komponen kompleks seperti tabel dan dialogs.

Keunggulan Java Swing:

1. Bersifat lightweight (tidak bergantung pada sistem operasi).
2. Memiliki banyak komponen bawaan.
3. Memungkinkan pembuatan layout yang fleksibel.

3.Container J-Frame

JFrame adalah kelas utama dalam swing yang digunakan untuk membuat jendela aplikasi. JFrame berfungsi sebagai wadah (container) utama yang menampung berbagai komponen GUI seperti tombol, label, dan text field.

4. Component GUI Java Swing

Komponen GUI adalah elemen visual yang digunakan pengguna untuk berinteraksi dengan program. Beberapa komponen dasar yang digunakan pada praktikum ini antara lain:

1. JLabel Digunakan untuk menampilkan teks atau informasi statis. Fungsi: memberi petunjuk, judul, atau deskripsi input.

2. JTextField Komponen input untuk menerima masukan berupa teks dari pengguna. Biasanya digunakan untuk formulir, input nilai, atau data sederhana.

3. JButton Tombol yang dapat ditekan pengguna untuk melakukan aksi tertentu. Umumnya digunakan bersama ActionListener.

4. JPanel Wadah tambahan yang digunakan untuk menata komponen secara rapi berdasarkan layout tertentu. Membantu modularisasi tampilan aplikasi.

5. JComboBox Drop-down menu untuk memilih salah satu opsi dari daftar yang dibuat.

6. JCheckBox & JRadioButton Komponen input pilihan:

i. **JCheckBox** (pilihan bisa lebih dari satu)

ii. **JRadioButton** (pilihan tunggal dalam satu grup)

5.Event Handling

Event handling adalah mekanisme untuk menangani aksi pengguna, seperti menekan tombol, mengetik teks, atau memilih opsi. Dalam Java Swing, event handling dilakukan menggunakan listener, seperti:

i. **ActionListener**, menangani aksi tombol

ii.KeyListener, menangani input keyboard.

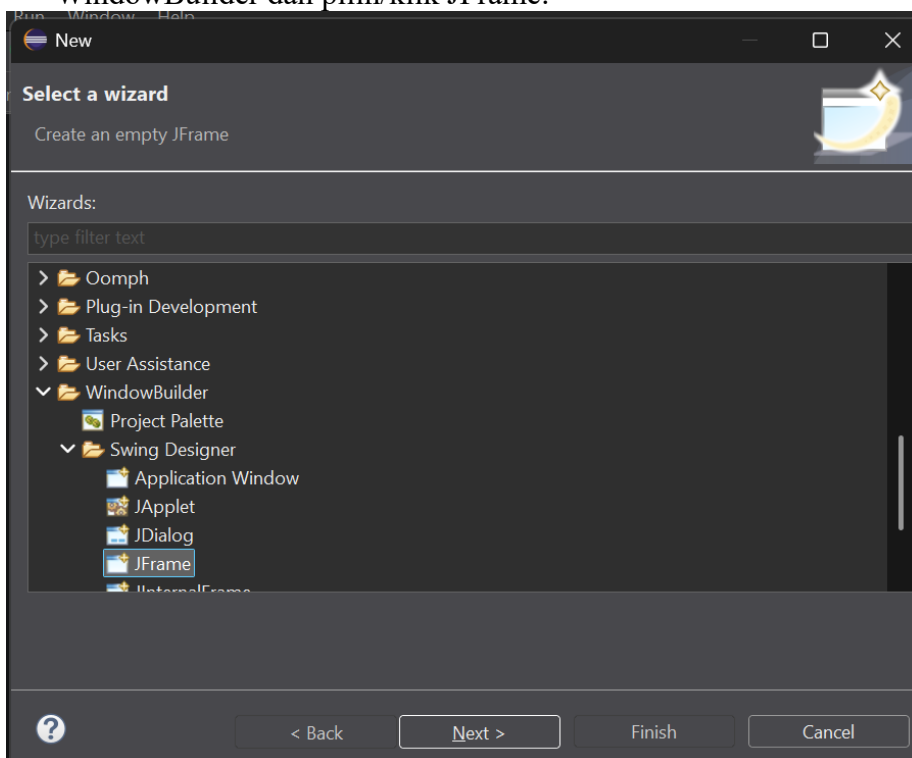
iii.MouseListener, menangani klik mouse.

Event handling memastikan aplikasi merespon interaksi pengguna dengan benar dan dinamis.

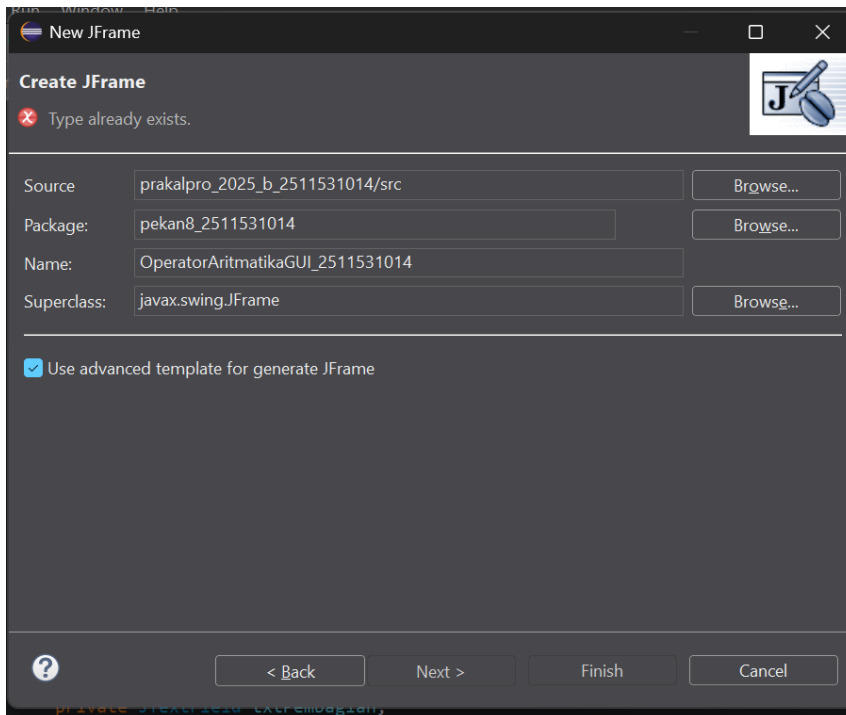
2,2 Praktikum Pekan 8

2.1.1 Membuat Package dan Class baru

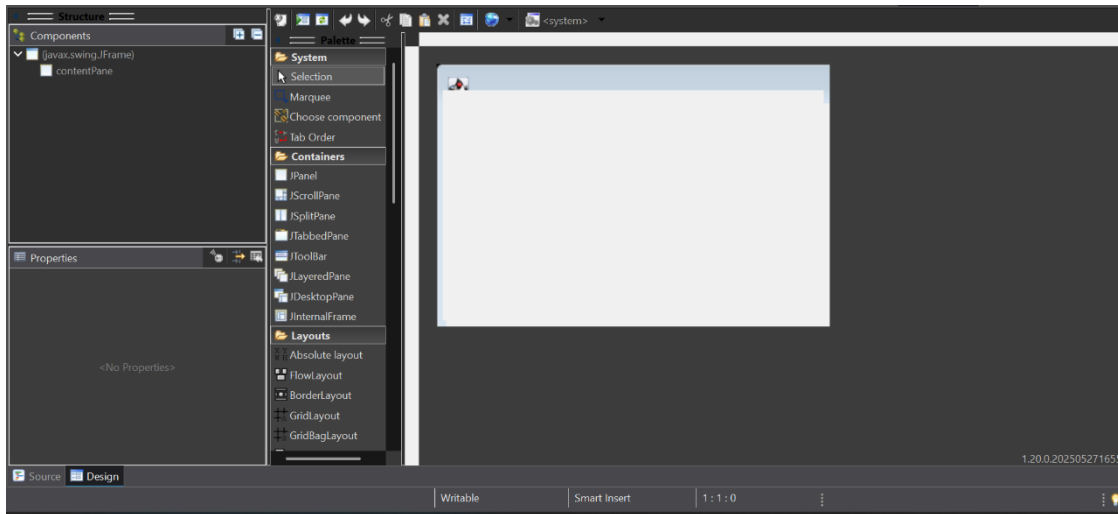
1. Pertama buka apl Eclipse IDE for Java Developers, lalu klik kanan klik other pilih WindowBuilder dan pilih/klik JFrame.



Setelah tampilan ini lanjutkan dengan membuat nama package “pekan8_NIM” dan buat nama class “OperatorAritmatika_NIM.” Setelah itu aktifkan “Use advanced template for generate JFrame”. Dan klik “Finish”. Maka akan kelas baru, dengan tab “Source” dan “Design”.



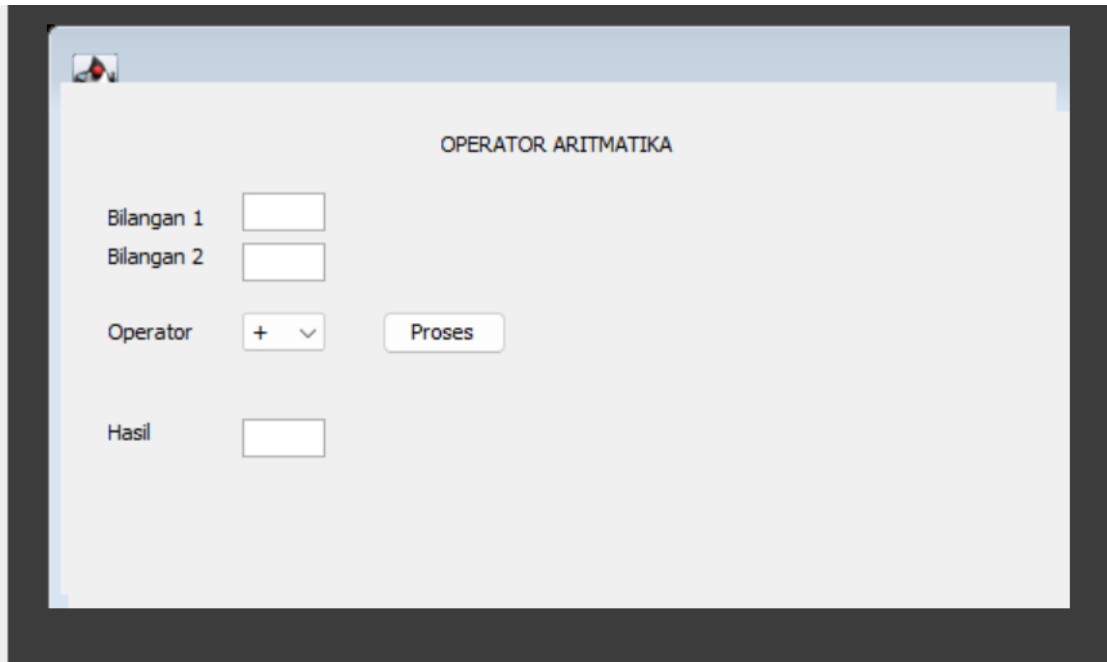
```
30 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8
9 public class text extends JFrame {
10
11     private static final long serialVersionUID = 1L;
12     private JPanel contentPane;
13
14     /**
15      * Launch the application.
16      */
17     public static void main(String[] args) {
18         EventQueue.invokeLater(new Runnable() {
19             public void run() {
20                 try {
21                     text frame = new text();
22                     frame.setVisible(true);
23                 } catch (Exception e) {
24                     e.printStackTrace();
25                 }
26             }
27         });
28     }
29
30     /**
31      * Create the frame.
32      */
33     public text() {
34         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35         setBounds(100, 100, 450, 300);
36         contentPane = new JPanel();
```



2.1.2 Membuat Design GUI

1. User dapat mulai membuat desain tampilan antar muka program dengan drag and drop. Pertama ubah title **JFrame** dengan klik kanan dan pilih rename menjadi 'Operator Aritmatika'. Atur size, warna background, dan layout sesuai yang di inginkan.
2. Menambahkan Komponen GUI dengan menambahkan **JLabel**, drag and drop ke panel. Maka akan muncul text box "New Lable" yang dapat diubah dengan "Bilangan 1, Bilangan 2, Proses, Hasil"
3. Mulai mengatur property dari tiap komponen, seperti memilih font, alignment, warna dan lainnya. Misalkan disini saya mengedit component "OPERATOR ARITMATIKA", mengganti font nya menjadi Arial dengan style Bold dan ukuran 18.
4. Selanjutnya component bertipe **JTextFiled** untuk menerima input atau memberikan output. Seperti berikut:
 - a) txtBil1 untuk input bilangan pertama
 - b) txtBil2 untuk input bilangan kedua
 - c) txtHasil untuk menampilkan hasil operasi
5. Lalu component **JComboBox** untuk pilihan operator, pilihan + (penjumlahan), - (pengurangan), * (perkalian), / (pembagian), dan % (untuk mod atau sisa hasil bagi)
6. Agar bilangan dapat di kalkulasikan, perlu menggunakan **Jbutton** untuk tombol proses.

Akan menghasilkan sebuah Design seperti gambar dibawah



2.1.3 Membuat Syntax

Dimulai dengan Menginisiasikan variabel variabel

```

53  */
54  public OperatorAritmatikaGUI_2511531014() {
55      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
56      setBounds(100, 100, 363, 302);
57      contentPane = new JPanel();
58      contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
59      setContentPane(contentPane);
60
61      txtBil1 = new JTextField();
62      txtBil1.setColumns(10);
63      txtBil1.setBounds(82, 84, 96, 20);
64      contentPane.add(txtBil1);
65
66      txtBil2 = new JTextField();
67      txtBil2.setColumns(10);
68
69      JComboBox comboBox = new JComboBox();
70      comboBox.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
71
72      JButton btnNewButton = new JButton("Proses");
73      btnNewButton.addActionListener(new ActionListener() {
74          public void actionPerformed(ActionEvent e) {
75              int hasil = 0;
76
77              if (txtBil1.getText().trim().isEmpty()) {
78                  pesanPeringatan("Bilangan 1 harus diisi");
79              } else if (txtBil2.getText().trim().isEmpty()) {
80                  pesanPeringatan("Bilangan 2 harus diisi");
81              } else {
82                  try {
83                      int a = Integer.parseInt(txtBil1.getText());
84                      int b = Integer.parseInt(txtBil2.getText());
85                      int c = comboBox.getSelectedIndex();

```

Dilanjutkan dengan proses perhitungannya atau cara design dioperasikan

```

        if (c == 0) {
            hasil = a + b;
        }
        if (c == 1) {
            hasil = a - b;
        }
        if (c == 2) {
            hasil = a * b;
        }
        if (c == 3) {
            hasil = a / b;
        }
        if (c == 4) {
            hasil = a % b;
        }

    } catch (NumberFormatException ex) {
        pesanError("Bilangan 1 dan Bilangan 2 harus angka");
    }
    txtHasil.setText(String.valueOf(hasil));
}

btnNewButton.setBounds(219, 119, 88, 22);
contentPane.add(btnNewButton);
});
}

```

Dan dapat dilanjutkan dengan syntax yang berisikan tiap komponen, seperti memilih font, alignment, warna dan lainnya.

```

113     JLabel lblNewLabel = new JLabel("Bilangan 1");
114
115     JLabel lblBilangan = new JLabel("Bilangan 2");
116
117     JLabel lblHasil = new JLabel("Hasil");
118
119     JLabel lblOperatorAritmatika = new JLabel("OPERATOR ARITMATIKA");
120
121     JLabel lblOperator = new JLabel("Operator");
122
123
124     txtHasil = new JTextField();
125     txtHasil.setColumns(10);
126     GroupLayout gl_contentPane = new GroupLayout(contentPane);
127     gl_contentPane.setHorizontalGroup(
128         gl_contentPane.createParallelGroup(Alignment.TRAILING)
129             .addGroup(Alignment.LEADING, gl_contentPane.createSequentialGroup()
130                 .addGap(20)
131                 .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
132                     .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
133                         .addComponent(lblNewLabel)
134                         .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
135                             .addComponent(lblOperator)
136                             .addComponent(lblBilangan)))
137                     .addComponent(lblHasil))
138                 .addGap(18)
139                 .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
140                     .addComponent(txtBil1, 0, 0, Short.MAX_VALUE)
141                     .addComponent(txtBil2, 0, 0, Short.MAX_VALUE)
142                     .addComponent(txtHasil, 0, 0, Short.MAX_VALUE)
143                     .addComponent(comboBox, 0, 43, Short.MAX_VALUE))
144                 .addGap(29)
145                 .addComponent(btnNewButton)
146                 .addGap(277, Short.MAX_VALUE))
147     );
148
149     gl_contentPane.setVerticalGroup(
150         gl_contentPane.createParallelGroup(Alignment.LEADING)
151             .addGroup(gl_contentPane.createSequentialGroup()
152                 .addGap(10)
153                 .addComponent(lblNewLabel)
154                 .addGap(10)
155                 .addComponent(lblBilangan)
156                 .addGap(10)
157                 .addComponent(lblOperatorAritmatika)
158                 .addGap(10)
159                 .addComponent(lblOperator)
160                 .addGap(10)
161                 .addComponent(txtBil1)
162                 .addGap(10)
163                 .addComponent(txtBil2)
164                 .addGap(10)
165                 .addComponent(txtHasil)
166                 .addGap(10)
167                 .addComponent(comboBox)
168                 .addGap(10)
169                 .addComponent(btnNewButton)
170                 .addGap(10)
171                 .addComponent(lblHasil)
172             );
173
174     contentPane.setLayout(gl_contentPane);
175
176     public OperatorAritmatikaGUI_3511531014() {
177         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
178         setBounds(100, 100, 363, 302);
179         contentPane = new JPanel();
180         contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
181         setContentPane(contentPane);
182
183         txtBil1 = new JTextField();
184         txtBil1.setColumns(10);
185         txtBil1.setBounds(62, 84, 96, 20);
186         contentPane.add(txtBil1);
187
188         txtBil2 = new JTextField();
189         txtBil2.setColumns(10);
190
191         JComboBox comboBox = new JComboBox();
192         comboBox.setModel(new DefaultComboBoxModel(new String[] {"+", "-", "*", "/", "%"}));
193
194         JButton btnNewButton = new JButton("Proses");
195         btnNewButton.addActionListener(new ActionListener() {
196             public void actionPerformed(ActionEvent e) {
197                 int hasil = 0;
198
199                 if (txtBil1.getText().trim().isEmpty()) {
200                     pesanPeringatan("Bilangan 1 harus diisi");
201                 } else if (txtBil2.getText().trim().isEmpty()) {
202                     pesanPeringatan("Bilangan 2 harus diisi");
203                 } else {
204                     try {
205                         int a = Integer.parseInt(txtBil1.getText());
206                         int b = Integer.parseInt(txtBil2.getText());
207                         int c = comboBox.getSelectedIndex();
208                     } catch (NumberFormatException ex) {
209                         pesanError("Bilangan 1 dan Bilangan 2 harus angka");
210                     }
211                     txtHasil.setText(String.valueOf(hasil));
212                 }
213             }
214         });
215
216         btnNewButton.setBounds(219, 119, 88, 22);
217         contentPane.add(btnNewButton);
218     }
219
220     public static void main(String[] args) {
221         OperatorAritmatikaGUI_3511531014 operatorAritmatikaGUI_3511531014 = new OperatorAritmatikaGUI_3511531014();
222         operatorAritmatikaGUI_3511531014.setVisible(true);
223     }
224 }

```

Dari Syntax dan Design diatas dapat dihasilkan output seperti berikut

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator ▼

Proses

Hasil

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator ▼

Proses

Hasil

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator ▼

Proses

Hasil

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator ▼

Proses

Hasil

OPERATOR ARITMATIKA

Bilangan 1

Bilangan 2

Operator ▼

Proses

Hasil

BAB III

PENUTUP

3.1. Kesimpulan

Dari seluruh rangkaian praktikum pembuatan program Operator Aritmatika berbasis GUI ini, saya dapat menyimpulkan bahwa penerapan/pemakaian Components antarmuka seperti JLabel, JTextField, JButton, dan JComboBox sangat membantu dalam membangun antarmuka program yang interaktif. Melalui praktikum ini, saya belajar bagaimana cara menangani input dari user, menampilkannya kembali sebagai output, serta mengatur alur proses menggunakan event handling pada tombol. Dan melalui praktik langsung, saya juga menjadi lebih memahami fungsi dari setiap elemen antarmuka, seperti JLabel untuk menampilkan keterangan, JTextField sebagai tempat memasukkan dan menampilkan data, JComboBox sebagai pilihan, serta JButton sebagai pemicu proses perhitungan atau tombol untuk sesuatu hal sesuai keinginan kita yang merancang. Selain itu, penggunaan method khusus seperti pesanPeringatan() dan pesanError() memberikan pemahaman mengenai cara membuat fungsi sendiri untuk menampilkan pesan kesalahan dan peringatan secara lebih rapi dan terstruktur. Proses validasi input, penanganan exception dengan try-catch, serta pemilihan operasi aritmatika berdasarkan index combo box juga memperkuat pemahaman saya terhadap logika kondisi serta pengolahan data numerik. Secara keseluruhan, praktikum ini memberikan pengalaman penting dalam mengombinasikan konsep dasar pemrograman Java mulai dari method, parsing angka, validasi input, penanganan error, hingga pengelolaan event dalam GUI sehingga saya mampu membuat program yang lebih dinamis, responsif, dan mudah digunakan oleh user.

3.2. Saran

Pemahaman konsep dasar perulangan sangat penting bagi pemrogram Java pemula. Disarankan agar mahasiswa mempraktikkan berbagai contoh pemrograman menggunakan GUI untuk memahami perilakunya secara mendalam. Dan diharapkan agar mahasiswa lebih teliti dalam memahami syntax agar meminimalisir kesalahan penulisan kode.

DAFTAR PUSTAKA

Sumber Daring (website):

[1] Wahyudi, “Operator Aritmatika pada Java” PowerPoint slides, Mata Kuliah Algoritma dan pemrograman, Universitas Andalas, Padang, 2025.

[2] Oracle, “Trail: Creating a GUI With JFC/Swing,” Oracle Docs, 2024. Tersedia: <https://docs.oracle.com/javase/tutorial/uiswing/>

[3] GeeksforGeeks, “Introduction to Java Swing.” [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/java/introduction-to-java-swing/>. [Diakses: 11- Nov-2025].