

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

“GUI Operator Aritmatika”



disusun Oleh:

FILZI JELILA INDA ROBBANI

2511533019

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T.

Asisten Praktikum:

JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT, atas limpahan rahmat dan karunia-Nya sehingga praktikum Algoritma dan Pemrograman kali ini dapat saya selesaikan dengan baik dan lancar. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW.

Laporan ini saya susun untuk memenuhi salah satu tugas pada mata kuliah Praktikum Algoritma dan Pemrograman di Universitas Andalas. Laporan ini diharapkan dapat menambah wawasan lebih lanjut mengenai penggunaan GUI (*Graphical User Interface*) dalam pemrograman Java.

Saya menyampaikan terima kasih kepada dosen pengampu dan asisten praktikum yang telah membimbing serta memberikan arahan selama proses pembelajaran di kelas maupun di laboratorium komputer. Selain itu, saya juga berterima kasih kepada teman-teman praktikan atas bantuan dan dukungan yang diberikan, sehingga laporan ini dapat terselesaikan dengan baik.

Saya menyadari laporan ini masih jauh dari sempurna. Oleh karena itu, saya mohon maaf apabila terdapat kekurangan maupun kesalahan dalam penyusunan laporan ini. Harapan saya, semoga laporan ini dapat memberikan manfaat serta menambah pengetahuan bagi pembaca sekalian.

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan	2
1.3 Manfaat	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Kerja Praktikum	3
2.1.1 Membuat package dan class baru di Eclipse.....	3
2.1.2 Susun GUI atau men-Design/tampilan outputnya (aplikasi)	3
2.1.3 Menambah/sedikit mengubah kode program	7
BAB III PENUTUP	12
3.1 Kesimpulan	12
DAFTAR PUSTAKA	13

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada praktikum kali ini, saya mempelajari lebih lanjut mengenai pembuatan Graphical User Interface (GUI) menggunakan WindowBuilder serta penerapan logika pemrograman yang sebelumnya telah dipelajari pada materi dasar Java. Setelah memahami konsep dasar seperti variabel, operator, percabangan, dan perulangan, praktikum ini menjadi tahap lanjutan untuk mengimplementasikan konsep tersebut ke dalam sebuah aplikasi yang memiliki tampilan interaktif.

Dalam proses pembuatan GUI, WindowBuilder memberikan kemudahan bagi mahasiswa untuk mendesain tata letak/layout komponen secara visual, *seperti text field, combo box, label, hingga button*. Meskipun demikian, mahasiswa tetap dituntut untuk mengedit dan menambahkan kode program secara manual untuk menentukan cara kerja aplikasi.

Studi kasus yang digunakan pada praktikum ini adalah operator aritmatika, di mana kami membuat aplikasi penghitung sederhana. Program ini menerima input bilangan, memilih jenis operator (+, -, ×, ÷), lalu menampilkan hasil perhitungan. Dengan kasus ini, mahasiswa dapat mempraktikkan penerapan berbagai struktur logika seperti if-else, parsing data dari String ke tipe numerik, serta validasi input. Selain itu, konsep dasar object-oriented programming seperti pembuatan method juga digunakan untuk memisahkan proses perhitungan agar program lebih terstruktur dan mudah dipahami.

Melalui rangkaian kegiatan praktikum ini, diharapkan mampu memahami cara menggabungkan desain GUI dengan logika pemrograman Java sehingga dapat membangun aplikasi sederhana yang interaktif, fungsional, dan mudah digunakan. Pemahaman ini menjadi bekal penting untuk pengembangan aplikasi yang lebih kompleks pada materi pembelajaran selanjutnya.

1.2 Tujuan

Tujuan dilaksanakannya praktikum ini adalah sebagai berikut:

- a. Memahami dasar-dasar pembuatan *Graphical User Interface* (GUI) menggunakan WindowBuilder pada Java.
- b. Menerapkan logika pemrograman dalam proses pengolahan input pengguna, seperti membaca nilai dari *text field* dan menentukan operasi aritmatika yang dipilih.
- c. Mengembangkan kemampuan dalam membuat dan menghubungkan *event handler* sehingga tombol maupun komponen GUI dapat berfungsi sesuai kebutuhan.
- d. Melatih keterampilan dalam memisahkan bagian logika program ke dalam method agar kode lebih terstruktur, rapi, dan mudah dipelihara.
- e. Mengasah kemampuan dalam merancang aplikasi sederhana berbasis GUI yang fungsional, interaktif, dan sesuai dengan studi kasus operator aritmatika.

1.3 Manfaat

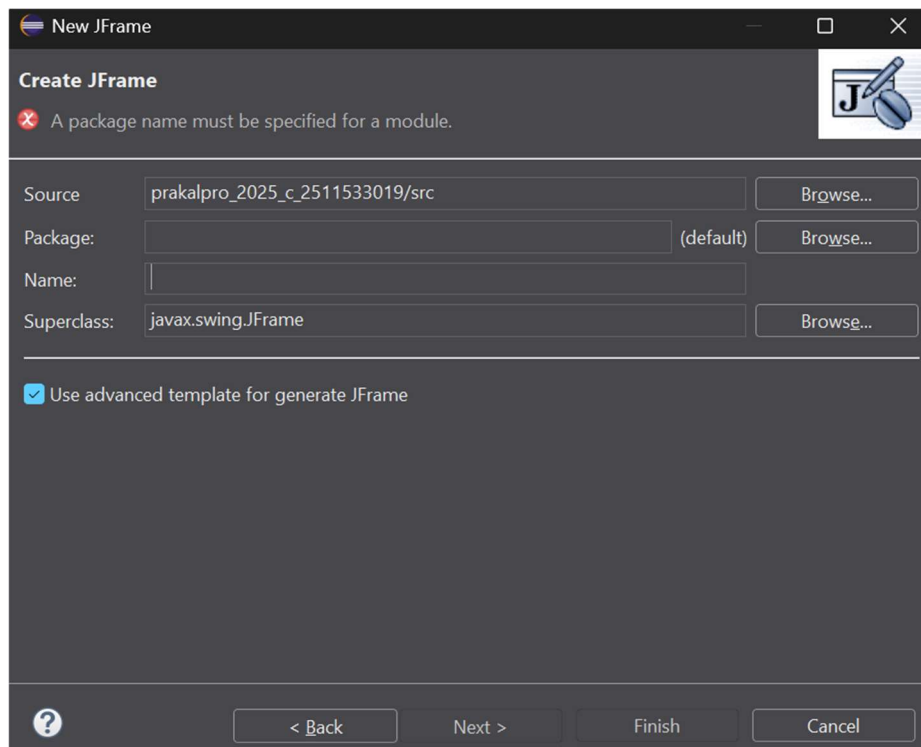
- a. Dapat memahami cara kerja komponen GUI dan bagaimana data diproses melalui input user dalam aplikasi Java.
- b. Memperoleh pengalaman dalam membuat program interaktif yang responsif dengan tampilan visual yang lebih mudah digunakan.
- c. Terlatih dalam membangun logika pemrograman melalui penerapan proses perhitungan aritmatika dan pengaturan *event* pada tombol maupun komponen lainnya.
- d. Menambah wawasan tentang bagaimana menggabungkan logika pemrograman dengan elemen GUI untuk menyelesaikan suatu studi kasus yang nyata.
- e. Dapat mengimplementasikan hasil pembelajaran untuk membangun aplikasi yang lebih dinamis, fleksibel, dan mudah dikembangkan pada materi berikutnya

BAB II PEMBAHASAN

2.1 Langkah Kerja Praktikum

2.1.1 Membuat package dan class baru di Eclipse

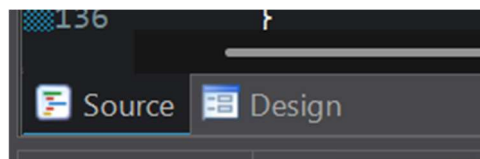
- 1) Pertama buka apl Eclipse IDE for Java Developers, lalu klik kanan klik other dan pilih WindowBuilder dan pilih/klik JFrame



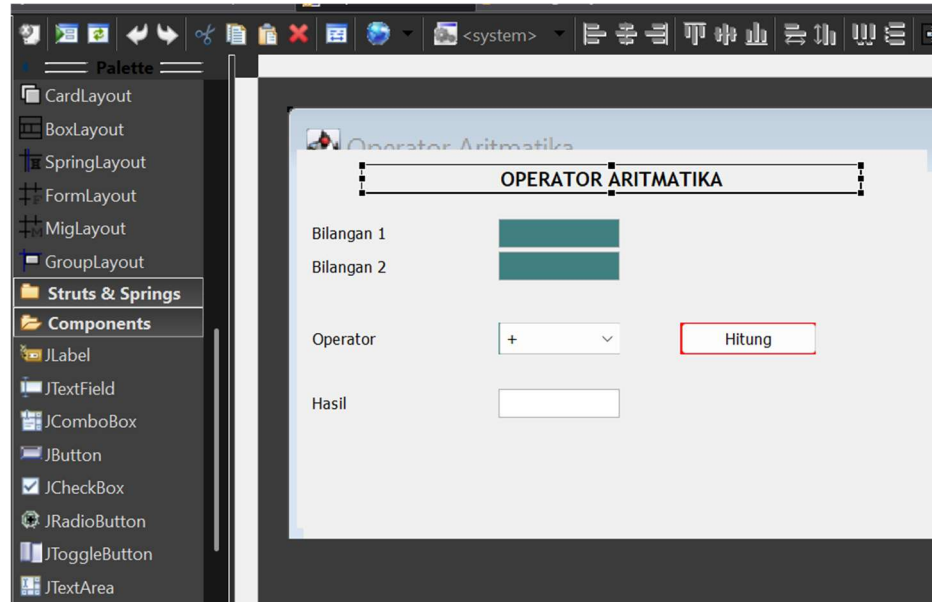
- 2) Setelah keluar tampilan seperti pada gambar, buat nama package “pekan8_NIM” dan buat nama class “OperatorAritmatika_NIM.”

2.1.2 Susun GUI atau men-Design/tampilan outputnya (aplikasi)

- 1) Klik Design button dibagian bawah dekat jendela kode program.



- 2) Pertama +label untuk judul dengan meng-klik JLabel pada bagian Components dan meletakannya dibagian Tengah atas. Lalu ganti properties pada label tsb seperti yang terlihat pada gambar berikut.



Properties	
Variable	lblNewLabel
Constructor	(Constructor prop...
Bounds	(45, 11, 346, 19)
Class	javax.swing.JLabel
background	240,240,240
displayed...	
enabled	<input checked="" type="checkbox"/> true
font	Trebuchet MS ...
foreground	0,0,0
horizontal...	CENTER
icon	
labelFor	
text	OPERATOR AR...
toolTipText	
verticalAlig...	CENTER

- 3) Selanjutnya +Label lagi untuk sub judul (Bilangan 1 dan 2, Operator serta Hasil) dengan properties seperti berikut.

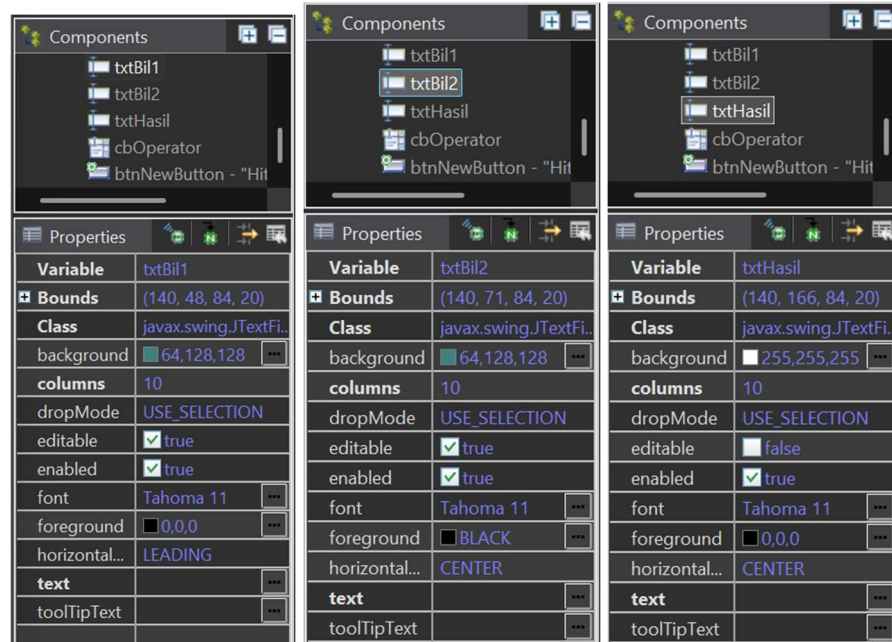
Properties	
Variable	lblNewLabel_1
Constructor	(Constructor prop...
Bounds	(10, 51, 71, 14)
Class	javax.swing.JLabel
background	240,240,240
displayed...	
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	0,0,0
horizontal...	LEADING
icon	
labelFor	
text	Bilangan 1
toolTipText	
verticalAlig...	CENTER

Properties	
Variable	lblNewLabel_1_1
Constructor	(Constructor prop...
Bounds	(10, 74, 71, 14)
Class	javax.swing.JLabel
background	240,240,240
displayed...	
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	0,0,0
horizontal...	LEADING
icon	
labelFor	
text	Bilangan 2
toolTipText	
verticalAlig...	CENTER

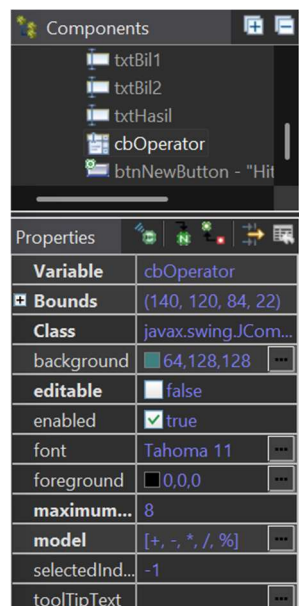
Properties	
Variable	lblNewLabel_1_2
Constructor	(Constructor prop...
Bounds	(10, 124, 71, 14)
Class	javax.swing.JLabel
background	240,240,240
displayed...	
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	0,0,0
horizontal...	LEADING
icon	
labelFor	
text	Operator
toolTipText	
verticalAlig...	CENTER

Properties	
Variable	lblNewLabel_1_3
Constructor	(Constructor prop...
Bounds	(10, 169, 71, 14)
Class	javax.swing.JLabel
background	240,240,240
displayed...	
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 11
foreground	0,0,0
horizontal...	LEADING
icon	
labelFor	
text	Hasil
toolTipText	
verticalAlig...	CENTER

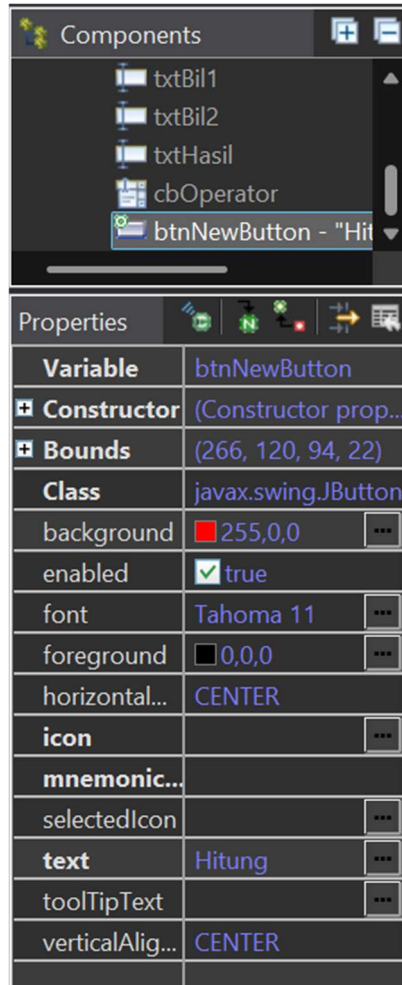
- 4) Setelah label, saya menambahkan Components JTextField untuk kolom penempatan inputan Bilangan 1 dan 2 dan outputan Hasil. Dengan properties seperti berikut.



- 5) Saya juga menambahkan JComboBox untuk tempat pemilihan operator yang nantinya akan digunakan user dalam melakukan operasi aritmatika. Dengan properties seperti berikut.



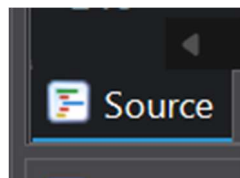
- 6) Dan yang terakhir menambahkan Component JButton yang nantinya digunakan untuk tombol mulai menghitung. Dengan properties seperti berikut.



- 7) Ctrl+S agar design/layout yang telah dirancang tersimpan.

2.1.3 Menambah/sedikit mengubah kode program

- 1) Klik source dibagian bawah dekat tombol design sebelumnya.



- 2) Lalu tambahkan kode program untuk melakukan operasi terhadap operator aritmatika serta kode program/syntax untuk menampilkan pesan peringatan dan pesan error. Perhatikan kode program tersebut.

```
1 package pekan8_2511533019;
2
30 import java.awt.BorderLayout;
20
21 public class OperatorAritmatikaGUI_2511533019 extends JFrame {
22
23     private static final long serialVersionUID = 1L;
24     private JPanel contentPane;
25     private JTextField txtBil1;
26     private JTextField txtBil2;
27     private JTextField txtHasil;
28
29
30     private void pesanPeringatan(String pesan) {
31         JOptionPane.showMessageDialog(this, pesan, "Peringatan", JOptionPane.WARNING_MESSAGE);
32     }
33     private void pesanError(String pesan) {
34         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
35     }
36     /**
37      * Launch the application.
38      */
39     public static void main(String[] args) {
40         EventQueue.invokeLater(new Runnable() {
41             public void run() {
42                 try {
43                     OperatorAritmatikaGUI_2511533019 frame = new OperatorAritmatikaGUI_2511533019();
44                     frame.setVisible(true);
45                 } catch (Exception e) {
46                     e.printStackTrace();
47                 }
48             }
49         });
50     }
51 }
```

```
50
51
52 /**
53  * Create the frame.
54  */
55 public OperatorAritmatikaGUI_2511533019() {
56     setTitle("Operator Aritmatika ");
57     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
58     setBounds(100, 100, 451, 300);
59     contentPane = new JPanel();
60     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
61     setContentPane(contentPane);
62     contentPane.setLayout(null);
63
64     JLabel lblNewLabel = new JLabel("OPERATOR ARITMATIKA");
65     lblNewLabel.setFont(new Font("Trebuchet MS", Font.BOLD, 14));
66     lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);
67     lblNewLabel.setBounds(45, 11, 346, 19);
68     contentPane.add(lblNewLabel);
69
70     JLabel lblNewLabel_1 = new JLabel("Bilangan 1");
71     lblNewLabel_1.setBounds(10, 51, 71, 14);
72     contentPane.add(lblNewLabel_1);
73
74     JLabel lblNewLabel_1_1 = new JLabel("Bilangan 2");
75     lblNewLabel_1_1.setBounds(10, 74, 71, 14);
76     contentPane.add(lblNewLabel_1_1);
77
78     JLabel lblNewLabel_1_2 = new JLabel("Operator");
79     lblNewLabel_1_2.setBounds(10, 124, 71, 14);
80     contentPane.add(lblNewLabel_1_2);
81
82     JLabel lblNewLabel_1_3 = new JLabel("Hasil");
```

```

83 lblNewLabel_1_3.setBounds(10, 169, 71, 14);
84 contentPane.add(lblNewLabel_1_3);
85
86 txtBil1 = new JTextField();
87 txtBil1.setBackground(new Color(64, 128, 128));
88 txtBil1.setForeground(new Color(0, 0, 0));
89 txtBil1.setBounds(140, 48, 84, 20);
90 contentPane.add(txtBil1);
91 txtBil1.setColumns(10);
92
93 txtBil2 = new JTextField();
94 txtBil2.setHorizontalAlignment(SwingConstants.CENTER);
95 txtBil2.setForeground(Color.BLACK);
96 txtBil2.setColumns(10);
97 txtBil2.setBackground(new Color(64, 128, 128));
98 txtBil2.setBounds(140, 71, 84, 20);
99 contentPane.add(txtBil2);
100
101 txtHasil = new JTextField();
102 txtHasil.setBackground(new Color(255, 255, 255));
103 txtHasil.setHorizontalAlignment(SwingConstants.CENTER);
104 txtHasil.setEditable(false);
105 txtHasil.setBounds(140, 166, 84, 20);
106 contentPane.add(txtHasil);
107 txtHasil.setColumns(10);
108
109 JComboBox cbOperator = new JComboBox();
110 cbOperator.setModel(new DefaultComboBoxModel(new String[] { "+", "-", "*", "/", "%"}));
111 cbOperator.setBackground(new Color(64, 128, 128));
112 cbOperator.setBounds(140, 120, 84, 22);
113 contentPane.add(cbOperator);
114
115 JButton btnNewButton = new JButton("Hitung");
116 btnNewButton.addActionListener(new ActionListener() {
117     int hasil;
118     public void actionPerformed(ActionEvent e) {
119         if(txtBil1.getText().trim().isEmpty()) {
120             pesanPeringatan("Bilangan 1 harus diisi");
121         } else if (txtBil2.getText().trim().isEmpty()) {
122             pesanPeringatan("Bilangan 2 harus diinput");
123         } else {
124             try {
125                 int a = Integer.valueOf(txtBil1.getText());
126                 int b = Integer.valueOf(txtBil2.getText());
127                 int c = cbOperator.getSelectedIndex();
128                 if (c==0) {hasil= a+b;}
129                 if (c==1) {hasil= a-b;}
130                 if (c==2) {hasil= a*b;}
131                 if (c==3) {hasil= a/b;}
132                 if (c==4) {hasil= a%b;}
133                 txtHasil.setText(String.valueOf(hasil));
134             } catch (NumberFormatException ex) {
135                 pesanError("Bilangan 1 dan Bilangan 2 harus angka!");
136             }
137         }
138     }
139 });
140 btnNewButton.setBackground(new Color(255, 0, 0));
141 btnNewButton.setBounds(266, 120, 94, 22);
142 contentPane.add(btnNewButton);
143
144 }
145 }
146

```

Pada program ini kode yang saya tambahkan sendiri tu dimulai pada kode baris 30-35 dan 117-135.

Pada baris 30-35, saya membuat 2 method khusus untuk menampilkan pesan kepada user ketika terjadi kesalahan dalam pengisian data. Method pertama yaitu *pesanPeringatan()*, yang berfungsi untuk menampilkan pesan peringatan ketika user melakukan kesalahan ringan, seperti tidak mengisi salah satu kolom input. Method ini menggunakan *JOptionPane.showMessageDialog()* dan menampilkan jendela pop-up dengan ikon peringatan berwarna kuning.

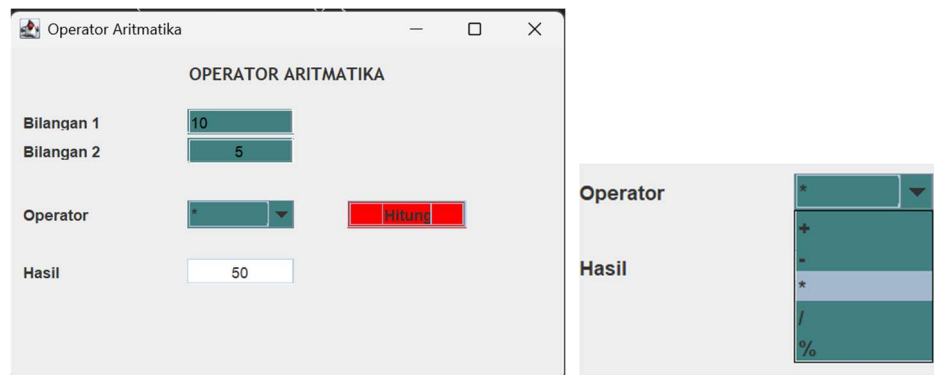
Sedangkan method kedua adalah *pesanError()*, yang digunakan untuk menampilkan pesan kesalahan yang sifatnya lebih serius. Misalnya, ketika user memasukkan karakter yang bukan angka sehingga program tidak dapat memproses perhitungan. Method ini juga menggunakan *JOptionPane.showMessageDialog()*, tetapi dengan tipe `ERROR_MESSAGE` sehingga ikon yang muncul berwarna merah untuk menandakan kesalahan.

Setelah itu pada baris 117-135, saya membuat sebuah variabel bernama *hasil* yang digunakan untuk menyimpan hasil operasi aritmatika. Ketika tombol ditekan, program awalnya akan memeriksa apakah kolom Bilangan 1 atau Bilangan 2 masih kosong. Jika ada yang belum diisi, program akan memanggil method *pesanPeringatan()* untuk menampilkan pop-up agar user mengisi data tersebut terlebih dahulu.

Jika kedua kolom sudah terisi, program masuk ke dalam blok try untuk mencoba mengubah teks pada kedua input menjadi tipe data integer menggunakan *Integer.valueOf()*. Setelah berhasil diubah menjadi angka, program mengambil operator yang dipilih user pada combo box melalui *getSelectedIndex()*. Index ini menentukan operasi apa yang akan dijalankan, seperti penjumlahan, pengurangan, perkalian, pembagian, atau modulus. Berdasarkan nilai index tersebut, program menghitung hasil operasi antara dua bilangan dan menyimpannya ke dalam variabel *hasil*. Setelah operasi selesai, nilai hasil perhitungan ditampilkan ke kotak

teks output menggunakan `txtHasil.setText()`. Apabila proses konversi angka gagal, misalnya user memasukkan huruf atau karakter lain, program akan otomatis pindah ke blok catch dan memanggil method `pesanError()` untuk memberi tahu bahwa kedua input harus berupa angka.

- 3) Run program dengan mengklik tombol run (bulatan hijau di bar menu) dan nanti akan muncul output dari program berupa tampilan GUI seperti berikut.



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.

DAFTAR PUSTAKA

- [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] H. M. Deitel and P. J. Deitel, *Java: How to Program*, 10th ed. Pearson, 2016.
Tersedia: <https://www.rmhjaved.com/oop/Java%20-%20How%20to%20Program%2010th%20Ed%20-%20Early%20Objects%20Version%20-%20Deitel.pdf>
- [4] Oracle, “*The Java™ Tutorials: Exceptions*,” Oracle Docs, 2024.
Available: <https://docs.oracle.com/javase/tutorial/essential/exceptions/>
- [5] Oracle, “*Using Swing Components*,” Java Platform SE 8, 2024.
Available: <https://docs.oracle.com/javase/tutorial/uiswing/components/>