

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
PENERAPAN GRAPHICAL USER INTERFACE (GUI) DALAM JAVA



disusun Oleh:
Annisa Layli Ramadhani
2511532024
Informatika Kelas B

Dosen Pengampu: Wahyudi. Dr.. S.T.M.T
Asisten Pratikum: Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena dengan rahmat-Nya laporan praktikum Algoritma dan Pemrograman tanggal 17 November 2025 dapat diselesaikan. Laporan ini membahas materi tentang Penerapan Graphical User Interface (GUI) dalam Java sebagai dasar penting dalam memahami pemrograman. Penulis mengucapkan terima kasih kepada dosen, asisten, dan rekan-rekan yang telah membantu selama praktikum. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan ini bermanfaat bagi pembaca.

Padang, 17 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Tujuan	5
1.3 Manfaat	5
BAB II PEMBAHASAN.....	6
2.1 Dasar Teori	6
2.1.1 Window Builder	6
2.1.2 Java Swing	7
2.1.3 JFrame	7
2.1.4 Komponen JFrame	8
2.1.5 Event Handling.....	9
2.1.6 Validasi Input	10
2.2 Langkah Kerja.....	10
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i> Pekan 8	10
2.2.2 <i>Commit</i> dan <i>Push</i> ke GitHub.....	18
BAB III PENUTUP.....	20
3.1 Kesimpulan	20
3.2 Saran.....	20
DAFTAR PUSTAKA	21

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah mendorong kebutuhan akan aplikasi yang mampu membantu pengguna dalam melakukan perhitungan secara cepat, tepat, dan mudah. Salah satu operasi dasar yang paling sering digunakan dalam berbagai aktivitas komputasi adalah operasi aritmatika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulo. Operasi-operasi dasar ini menjadi fondasi bagi banyak proses perhitungan dalam pemrograman maupun pengolahan data.

Dalam lingkungan pengembangan perangkat lunak, bahasa pemrograman java merupakan salah satu bahasa yang banyak digunakan karena sifatnya yang berorientasi objek, portabel, serta didukung oleh berbagai library untuk membangun antarmuka grafis. Salah satu library tersebut adalah Java Swing, yang memungkinkan pengembang membuat aplikasi berbasis GUI (Graphical User Interface) sehingga lebih interaktif dan mudah digunakan oleh pengguna.

Pembuatan aplikasi perhitungan menggunakan operator aritmatika berbasis GUI bertujuan untuk memberikan contoh implementasi dasar pemrograman java dalam mengolah input pengguna, melakukan validasi data, serta menampilkan hasil perhitungan secara langsung. Program ini juga dapat menjadi sarana pembelajaran bagi mahasiswa atau pemula untuk memahami cara kerja komponen-komponen java swing seperti JTextField, JButton, JComboBox, dan metode penanganan event (ActionListener). Dengan adanya antarmuka yang sederhana, pengguna dapat memasukkan dua bilangan, memilih operator, dan memperoleh hasil perhitungan secara instan tanpa perlu memahami detail kode program yang ada dibalikny.

Selain itu, implementasi fitur validasi input dan penanganan error juga menjadi bagian penting dalam pengembangan aplikasi. Program ini memberikan contoh bagaimana menangani kesalahan seperti input

kosong, nilai yang bukan angka, atau pembagian bilangan yang tidak valid. Dengan demikian, aplikasi operator aritmatika berbasis GUI ini tidak hanya berfungsi sebagai alat bantu sederhana, tetapi juga sebagai media pembelajaran mengenai konsep dasar pemrograman, logika percabangan, serta desain antarmuka pada java.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.2.1 Membuat aplikasi sederhana yang dapat melakukan operasi aritmatika dasar, seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulo menggunakan antarmuka grafis.
- 1.2.2 Mengimplementasikan komponen-komponen java swing, seperti JTextField, JButton, JComboBox, dan JLabel dalam sebuah program GUI.
- 1.2.3 Menunjukkan contoh penerapan logika percabangan dalam menentukan operasi aritmatika berdasarkan pilihan pengguna.

1.3 Manfaat

Manfaat dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.3.1 Mempermudah proses bilangan dan memilih operator tanpa melakukan perhitungan manual.
- 1.3.2 Memberikan pengalaman langsung dalam menggunakan java swing, sehingga meningkatkan kualitas dan keandalan aplikasi.
- 1.3.3 Meningkatkan keterampilan logika dan algoritma, terutama terkait penggunaan struktur percabangan dan pengolahan data numerik.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Pada pembuatan aplikasi berbasis GUI menggunakan java, diperlukan pemahaman yang jelas mengenai konsep dan teknologi yang mendukung proses perancangannya. Dasar teori menjadi landasan penting yang menjelaskan prinsip, alat, dan mekanisme yang digunakan dalam pengembangan aplikasi. Dengan memahami teori tersebut, proses pembuatan program dapat dilakukan secara lebih terarah dan sesuai dengan kaidah pemrograman yang baik. Oleh karena itu, bagian ini membahas beberapa konsep utama yang berkaitan dengan penggunaan Window Builder, Java Swing, JFrame beserta komponennya, serta mekanisme event handling, validasi input, dan exception handling yang menjadi dasar dalam pembuatan aplikasi GUI pada program ini.

2.1.1 Window Builder

Window builder merupakan sebuah plugin pada Eclipse IDE yang digunakan untuk merancang antarmuka grafis (GUI) secara visual pada bahasa pemrograman Java. Dengan tool ini, pengembang dapat membuat tampilan aplikasi secara drag and drop tanpa harus menuliskan seluruh kode secara manual. Window builder mendukung berbagai komponen GUI seperti label, tombol, text field, panel, serta komponen swing lainnya.

Penggunaan window builder mempermudah proses desain karena setiap perubahan yang dilakukan pada tampilan akan otomatis menghasilkan kode java secara terstruktur. Hal ini membantu pengembang bekerja lebih cepat, mengurangi kesalahan penulisan kode, dan mempermudah pemahaman terhadap struktur GUI. Window builder juga menyediakan fitur pengaturan tata letak (layout), pengelolaan properti komponen,

serta integrasi langsung dengan logika program yang ditambahkan didalam kode.

2.1.2 Java Swing

Java Swing adalah library bawaan java yang digunakan untuk membuat antarmuka grafis atau GUI (Graphical User Interface). Swing menyediakan berbagai komponen visual yang dapat digunakan untuk membangun aplikasi desktop secara interaktif. Library ini bersifat lightweight, sehingga komponen yang disediakan tidak bergantung pada tampilan sistem operasi dan dapat berfungsi secara konsisten di berbagai platform.

Swing juga mendukung pengaturan tata letak, pengolahan event, serta kustomisasi tampilan komponen, sehingga memungkinkan pengembang membuat aplikasi yang lebih fleksibel dan mudah digunakan. Dengan kemampuannya yang luas, Java Swing menjadi salah satu pilihan utama dalam pengembangan GUI berbasis Java.

2.1.3 JFrame

JFrame merupakan kelas utama dalam Java Swing yang digunakan sebagai jendela utama aplikasi GUI. Pada program yang dibuat, JFrame berfungsi sebagai wadah yang menampung berbagai komponen antarmuka, seperti tombol, label, dan text field yang digunakan pengguna untuk memasukkan data serta memicu proses tertentu. Melalui JFrame, pengembang dapat mengatur ukuran jendela, tata letak komponen, judul aplikasi, serta perilaku program ketika jendela ditutup. Selain itu, JFrame menyediakan metode untuk menambahkan komponen antarmuka dan mengatur tampilannya secara visual sehingga interaksi pengguna dengan program dapat berjalan dengan baik. Dengan memanfaatkan JFrame, seluruh elemen pada aplikasi

dapat ditata secara rapi, mudah diakses, dan mampu mendukung proses kerja aplikasi secara menyeluruh.

2.1.4 Komponen JFrame

Dalam pembangunan antarmuka grafis pada java, JFrame menjadi elemen utama yang berfungsi sebagai jendela utama tempat seluruh komponen aplikasi ditempatkan. Untuk memahami struktur tampilan program, penting untuk mengetahui komponen-komponen yang digunakan di dalam JFrame.

1. JLabel

JLabel adalah komponen yang digunakan untuk menampilkan teks statis pada antarmuka. Fungsinya sebagai penanda atau keterangan bagi pengguna, seperti label untuk input “Bilangan 1”, “Bilangan 2”, “Operator”, dan “Hasil”. JLabel tidak bisa diedit oleh pengguna dan hanya berfungsi sebagai informasi visual.

2. JTextField

JTextField merupakan komponen tempat pengguna memasukkan data dalam bentuk teks. Pada program, JTextField digunakan untuk menginput bilangan pertama dan kedua, serta menampilkan hasil operasi. Komponen ini dapat dibaca oleh program untuk mengambil nilai input dari pengguna.

3. JButton

JButton adalah tombol yang dapat diklik pengguna untuk menjalankan suatu aksi. Pada program, tombol “Proses” digunakan untuk memicu perhitungan berdasarkan input yang diberikan. Event klik tombol ini ditangani melalui ActionListener.

4. JPanel

JPanel adalah wadah atau kontainer yang menampung berbagai komponen GUI lain. Dalam program ini, JPanel berfungsi sebagai area utama tempat penempatan semua komponen seperti label, tombol, text field, dan combo box. JPanel juga mendukung pengaturan warna latar dan tata letak.

5. JComboBox

JComboBox merupakan komponen dropdown yang menyediakan daftar pilihan kepada pengguna. Pada program ini, JComboBox digunakan untuk memilih operator aritmatika seperti +, −, *, /, dan %. Pilihan yang dipilih akan digunakan dalam proses perhitungan.

2.1.5 Event Handling

Event handling adalah mekanisme dalam Java Swing yang digunakan untuk menangani aksi atau interaksi yang dilakukan oleh pengguna terhadap komponen antarmuka. event handling diterapkan saat pengguna menekan tombol Proses. Aksi tersebut ditangkap oleh ActionListener, yaitu antarmuka yang bertugas merespons suatu event. Ketika tombol diklik, metode actionPerformed() akan dijalankan untuk melakukan serangkaian proses, mulai dari membaca input pengguna melalui JTextField, memeriksa operator yang dipilih pada JComboBox, hingga melakukan perhitungan aritmatika. Selain itu, event handling juga digunakan untuk menampilkan pesan peringatan atau kesalahan melalui JOptionPane apabila terjadi input tidak valid. Dengan mekanisme ini, aplikasi dapat merespons tindakan pengguna secara interaktif dan menjalankan fungsi program sesuai kebutuhan.

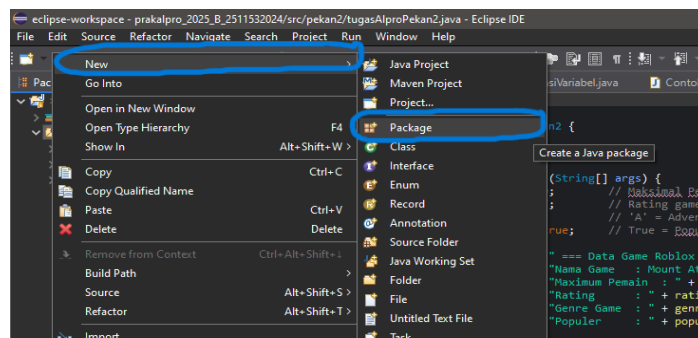
2.1.6 Validasi Input

Validasi input pada program dilakukan untuk memastikan bahwa data yang dimasukkan pengguna sudah benar sebelum dihitung. Program terlebih dahulu memeriksa apakah kolom Bilangan 1 dan Bilangan 2 kosong, dan jika kosong akan menampilkan peringatan melalui pesanPeringatan(). Selain itu, validasi juga dilakukan saat mengonversi input menjadi angka menggunakan Integer.parseInt(). Jika pengguna memasukkan karakter non-angka, program menangani kesalahan tersebut dengan menampilkan pesan melalui pesanError(). Dengan validasi ini, program dapat mencegah kesalahan perhitungan dan memastikan input yang digunakan sudah valid.

2.2 Langkah Kerja

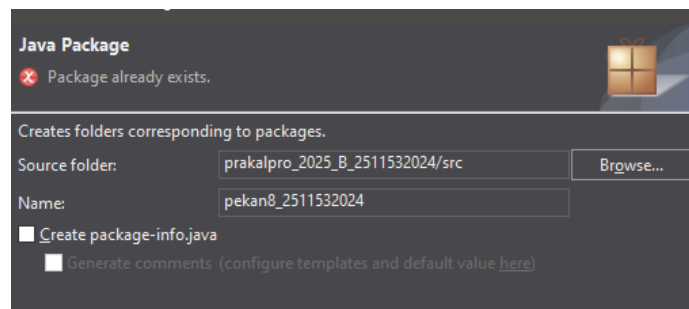
2.2.1 Pembuatan *Package* dan *Class* Pekan 8

1. Buka aplikasi eclipse, lalu buat package baru pada src klik kanan, lalu pilih “New” dan klik tulisan Package.



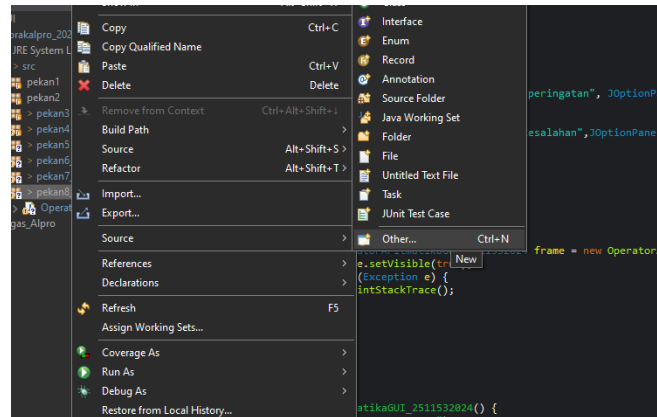
2.2.1 pembuatan package

2. Kemudian buat nama packagenya tanpa spasi, huruf kapital, karakter khusus. Beri nama “pekan8_2511532024”. Lalu klik finish.



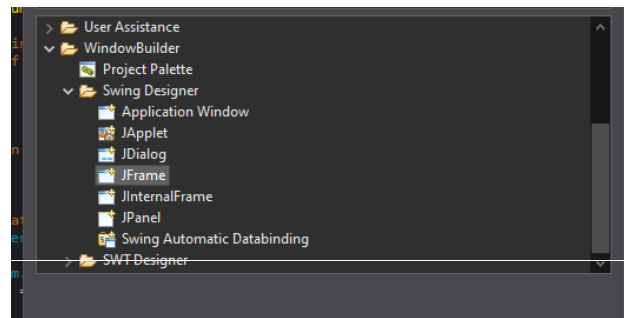
2.2.2 pembuatan package

3. Ketika package sudah jadi. Klik kanan pada bagian “pekan8_2511532024”. Lalu klik “new” dan pilih bagian

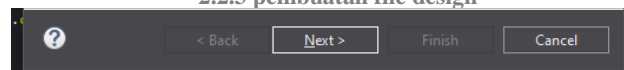


“other” untuk memulai membuat program.

4. Kemudian pilih bagian “JFrame”. Dan setelah itu klik next.

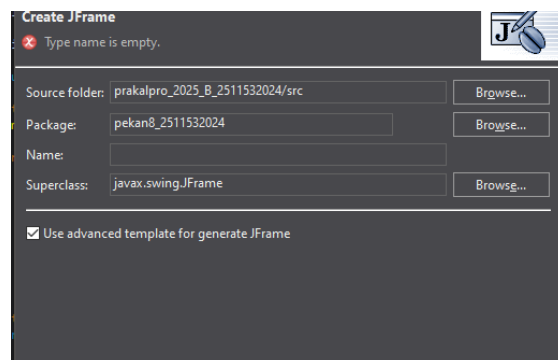


2.2.3 pembuatan file design



2.2.4 pembuatan file design

5. Setelah itu muncul Create JFrame, buat nama file dengan ketentuan nama harus capslock diawal kalimat. Lalu klik finish



2.2.5 pembuatan file design

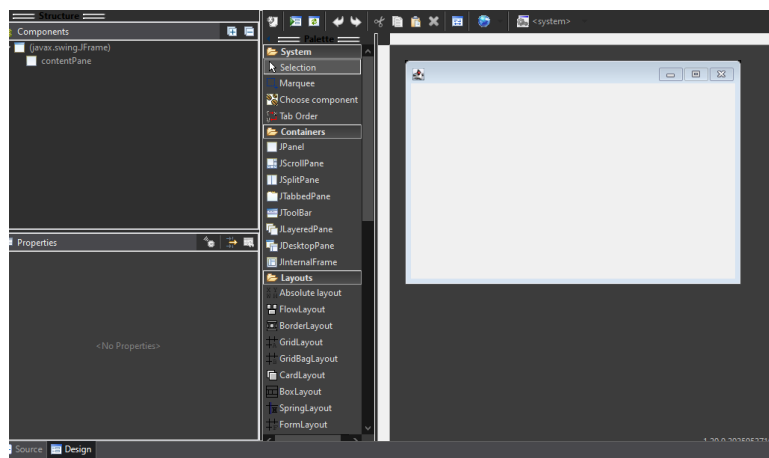
6. Setelah klik finish akan muncul source dan design.

```

1 package pekan8_2511532024;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8 import javax.swing.JLabel;
9 import javax.swing.JOptionPane;
10
11 import java.awt.Font;
12 import javax.swing.JTextField;
13 import javax.swing.JComboBox;
14 import javax.swing.DefaultComboBoxModel;
15 import javax.swing.SwingConstants;
16 import javax.swing.JButton;
17 import java.awt.event.ActionListener;
18 import java.awt.event.ActionEvent;
19 import java.awt.Color;
20
21 public class OperatorAritmatikaGUI_2511532024 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     private void pesanPeringatan(String pesan) {
30         JOptionPane.showMessageDialog(this, pesan, "peringatan", JOptionPane.INFORMATION_MESSAGE);
31     }
32     private void pesanError(String pesan) {
33         JOptionPane.showMessageDialog(this, pesan, "Kesalahan", JOptionPane.ERROR_MESSAGE);
34     }
35     /**
36      * Launch the application.
37      */

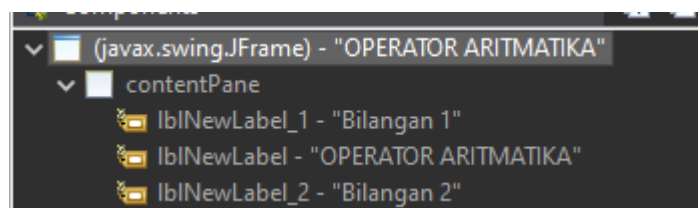
```

2.2.6 program operator aritmatika



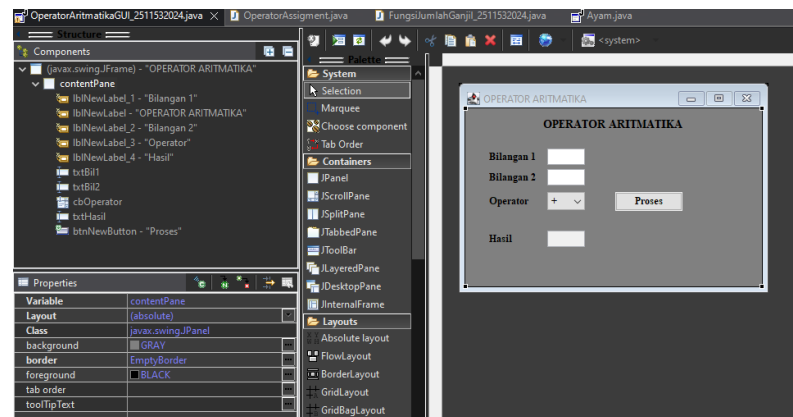
2.2.7 program operator aritmatika

7. Setelah itu kita bisa memulai mendesain tampilan program yang akan kita buat. Sebelum itu kita mengubah titlenya JFrame nya dengan nama 'OPERASI ARITMATIKA'.



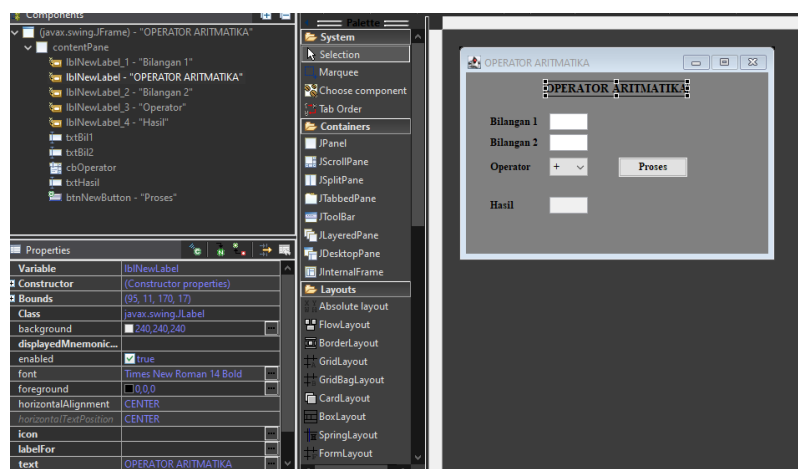
2.2.8 program operator aritmatika

8. Setelah mengubah titlenya kita bisa langsung mengatur ukuran, background atau layout pada desain program.



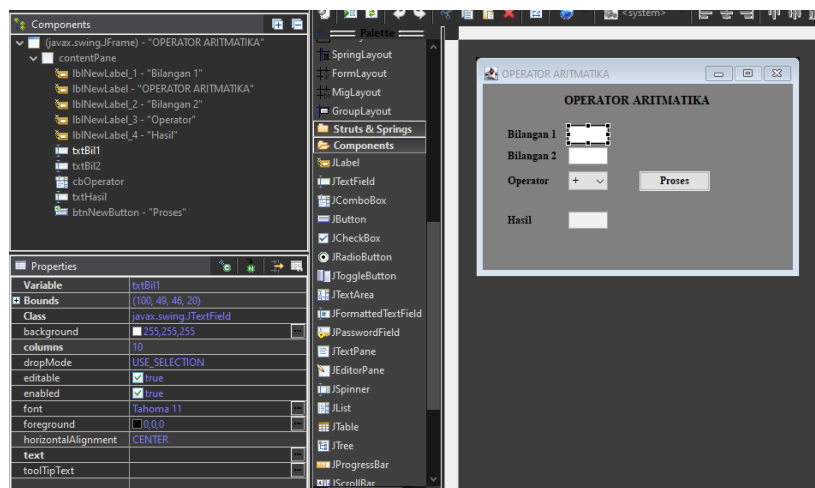
2.2.9 program operator aritmatika

9. Kemudian kita bisa mulai memasukkan komponen-komponennya seperti JLabel. Dan kita bisa langsung mengedit labelnya dengan nama “OPERASI ARITMATIKA”. Dan juga mengubah font dan size fontnya.



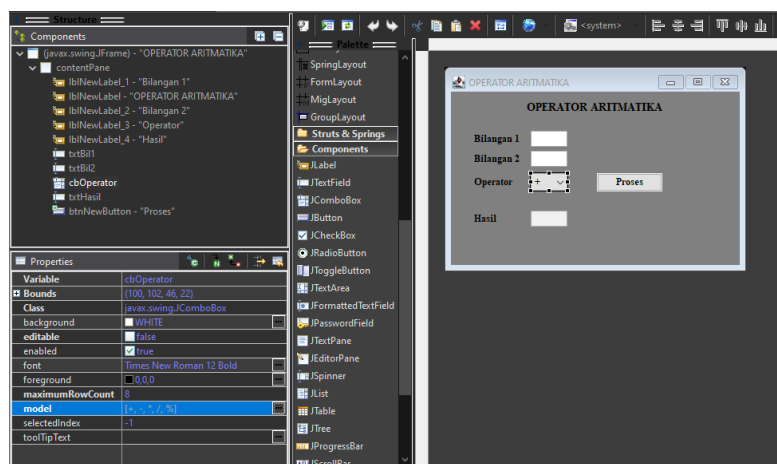
2.2.10 program operator aritmatika

10. Setelah kita bisa menambah komponen lainnya seperti JTextField dan JLabel, dan memberi nama JLabelnya Bilangan 1 dan Bilangan 2, kemudian kita mengaturnya pada jendela output dimana letaknya, besarnya. Dan kemudian mengedit dibagian JTextField propertiesnya yaitu mengubah variabelnya menjadi txtBil1 dan txtBil2.

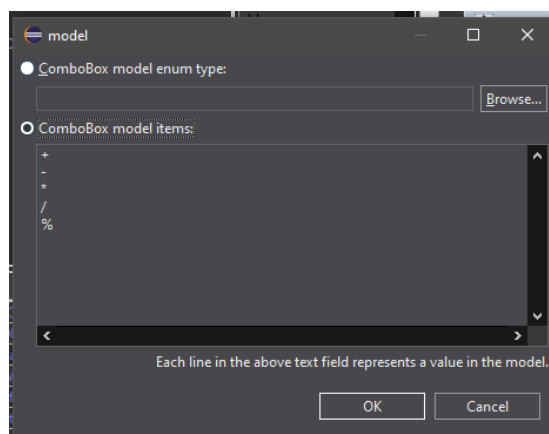


2.2.11 program operator aritmatika

11. Kemudian kita menambahkan komponen JLabel dan JComboBox untuk bisa memilih beberapa pilihan. Kemudian mengubah variabelnya menjadi cbOperator dan menambahkan pilihanya pada opsi model di propertiesnya.

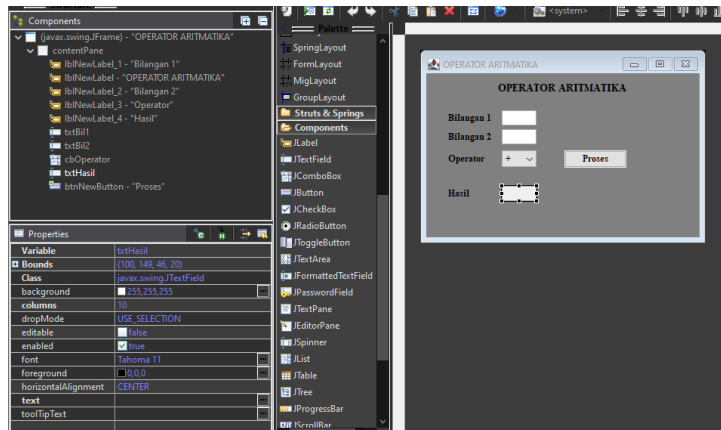


2.2.12 program operator aritmatika



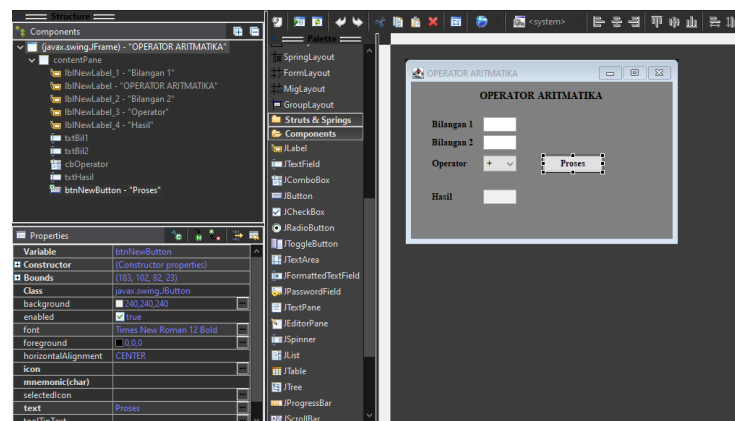
2.2.13 program operator aritmatika

12. Selanjutnya kita menambahkan komponen JLabel dan JTextField untuk bagian hasil dan mengubah variabel JTextFieldnya menjadi txthasil.



2.2.14 program opearator aritmatika

13. Kemudian kita menambahkan komponen JButton untuk memproses program yang akan kita buat sehingga bisa menghasilkan output yang diinginkan.



2.2.15 program operator aritmatika

14. Setelah selesai mengatur semuanya menjadi rapi, kita pindah ke bagian source untuk mulai membuat kode program agar programnya bekerja dengan benar.

```

JButton btnNewButton = new JButton("Proses");
btnNewButton.setFont(new Font("Times New Roman", Font.BOLD, 12));
btnNewButton.addActionListener(new ActionListener() {
    int hasil;
    public void actionPerformed(ActionEvent e) {
        if (txtBil1.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 1 harus diisi");
        } else if (txtBil2.getText().trim().isEmpty()) {
            pesanPeringatan("Bilangan 1 harus diisi");
        } else {
            try {
                int a = Integer.parseInt(txtBil1.getText());
                int b = Integer.parseInt(txtBil2.getText());
                int c = cbOperator.getSelectedIndex();
                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");
            }
        }
    }
}

```

2.2.16 program operator aritmatika

15. Kemudian kita juga menambahkan kode program agar ketika terjadi kesalahan akan muncul peringatan pada output programnya nanti.

```

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.

```

2.2.17 program operator aritmatika

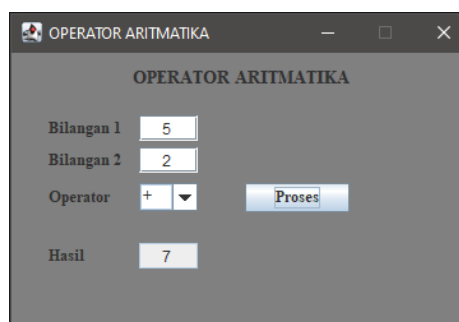
```

if (txtBil1.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 1 harus diisi");
} else if (txtBil2.getText().trim().isEmpty()) {
    pesanPeringatan("Bilangan 1 harus diisi");
} else

```

2.2.18 program operator aritmatika

16. Setelah selesai semua, klik run untuk menjalankan program yg telah dibuat.



2.2.19 output program

The screenshot shows a window titled "OPERATOR ARITMATIKA". Inside, there are four input fields: "Bilangan 1" with the value 9, "Bilangan 2" with the value 5, "Operator" with a dropdown menu showing "-", and "Hasil" with the value 4. A "Proses" button is located to the right of the operator dropdown.

2.2.20 output program

The screenshot shows a window titled "OPERATOR ARITMATIKA". Inside, there are four input fields: "Bilangan 1" with the value 2, "Bilangan 2" with the value 5, "Operator" with a dropdown menu showing "*", and "Hasil" with the value 10. A "Proses" button is located to the right of the operator dropdown.

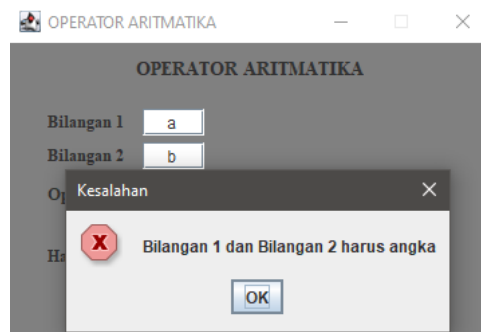
2.2.21 output program

The screenshot shows a window titled "OPERATOR ARITMATIKA". Inside, there are four input fields: "Bilangan 1" with the value 15, "Bilangan 2" with the value 5, "Operator" with a dropdown menu showing "/", and "Hasil" with the value 3. A "Proses" button is located to the right of the operator dropdown.

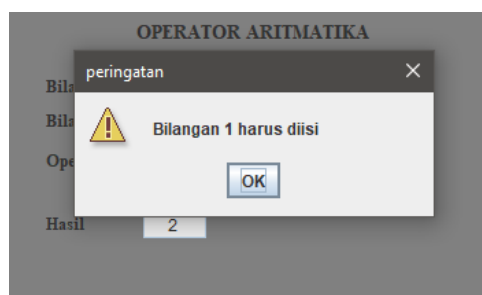
2.2.22 output program

The screenshot shows a window titled "OPERATOR ARITMATIKA". Inside, there are four input fields: "Bilangan 1" with the value 20, "Bilangan 2" with the value 6, "Operator" with a dropdown menu showing "%", and "Hasil" with the value 2. A "Proses" button is located to the right of the operator dropdown.

2.2.23 output program



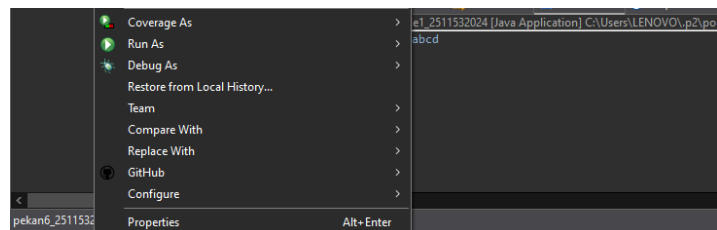
2.2.24 output program error



2.2.25 output program error

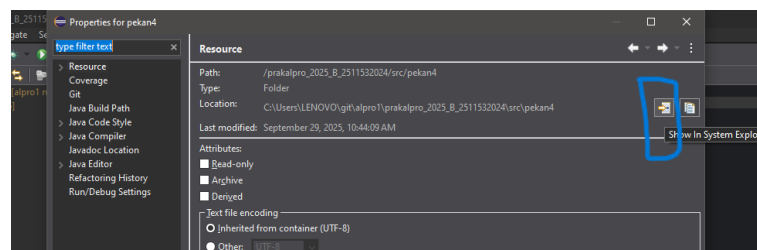
2.2.2 Commit dan Push ke GitHub

1. Klik kanan pada bagian “pekan8_2511532024”. Kemudian pilih bagian “properties”.



2.2.26 proses commit dan push ke GitHub

2. Setelah itu keluar opsi properties. Kemudian klik “Show In System Explorer”.



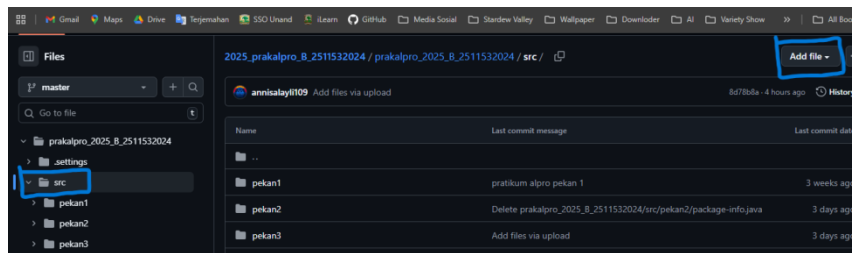
2.2.27 proses commit dan push ke github

3. Setelah itu akan terbuka folder file tersebut di file explorer.

pekan6_2511532024	11/7/2025 7:48 PM	File folder
pekan7_2511532024	11/16/2025 8:13 AM	File folder
pekan8_2511532024	11/20/2025 10:34 PM	File folder

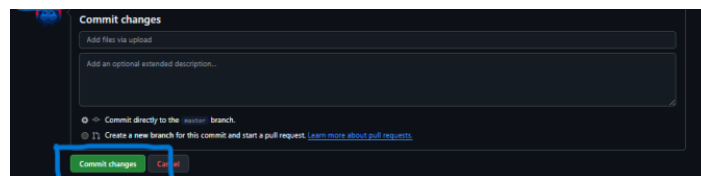
2.2.28 proses commit dan push ke GitHub

4. Kemudian buka akun GitHub dan buka pada bagian src. Dan klik pada bagian “add file”. Dan pilih “upload file”.



2.2.29 proses commit dan push ke GitHub

5. Setelah itu tarik file “pekan8_2511532024” dari file explorer ke bagian upload file dan kemudian tunggu loading file diunggah ke GitHub. Dan setelah itu klik “Commit changes”.



2.2.30 proses commit dan push ke GitHub

6. Kemudian cek akun GitHub untuk melihat apakah file yang diupload sudah masuk atau belum. Jika sudah maka upload sudah selesai.

pekan6_2511532024	Add file
pekan7_2511532024	Add file
pekan8_2511532024	Add file

2.2.31 proses commit dan push ke GitHub

BAB III

PENUTUP

3.1 Kesimpulan

Aplikasi yang dikembangkan berhasil menerapkan konsep dasar antarmuka grafis berbasis Java dengan cukup baik. Penggunaan komponen seperti JLabel, JTextField, JButton, JComboBox, dan JPanel menunjukkan pemahaman yang tepat terhadap struktur dan fungsi elemen GUI dalam Java Swing. Proses event handling mampu menangani setiap aksi pengguna secara efektif, khususnya saat tombol dipilih untuk menjalankan perhitungan. Selain itu, penerapan validasi input dan exception handling membuat aplikasi lebih aman dan stabil, karena mampu mencegah error akibat input kosong ataupun data yang tidak sesuai. Secara keseluruhan, aplikasi ini dapat menjalankan operasi aritmatika dengan benar dan memberikan respon yang sesuai terhadap interaksi pengguna, sehingga layak dijadikan contoh dasar pembuatan aplikasi GUI sederhana.

3.2 Saran

Aplikasi dapat ditingkatkan dengan menambahkan pemeriksaan khusus, seperti larangan membagi angka dengan nol, serta menyediakan tombol reset untuk mempermudah pengguna menghapus input. Selain itu, penerapan layout manager dapat membuat tampilan lebih rapi dan adaptif terhadap berbagai ukuran jendela.

DAFTAR PUSTAKA

- [1] Oracle, “*Trail: Creating a GUI With JFC/Swing (The Java™ Tutorials)*,” Oracle Documentation. <https://docs.oracle.com/javase/tutorial/uiswing/>
- [2] Oracle, “*Class JFrame*,” Java Platform SE Documentation. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JFrame.html>
- [3] Oracle, “*How to Write an Action Listener*,” Java Tutorials. <https://docs.oracle.com/javase/tutorial/uiswing/events/actionlistener.html>
- [4] Oracle, “*Class JOptionPane*,” Java Platform SE Documentation. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html>
- [5] Oracle, “*Class JPasswordField*,” Java Platform SE Documentation. <https://docs.oracle.com/javase/8/docs/api/javax/swing/JPasswordField.html>