

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

PEMROGRAMAN GUI

PEMROGRAMAN JAVA

disusun Oleh:

Nama: Amiratul Fadhilah

NIM: 2511532023

Dosen Pengampu: Dr. Wahyudi, S.T, M.T

Asisten Praktikum: Jovantri Immanuel Gulo



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Puji syukur ke hadirat Allah Yang Maha Esa atas rahmat dan karunia-Nya, sehingga laporan praktikum dengan judul “Pemrograman GUI” ini dapat diselesaikan dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk memenuhi tugas mata kuliah Praktikum Algoritma dan Pemrograman. Tujuan utama dari penyusunan laporan ini adalah untuk menguji pemahaman dan implementasi pustaka Java *Swing* dalam membangun aplikasi operator aritmatika sederhana. Pembahasannya mencakup pada penggunaan komponen dasar GUI seperti JFrame, JTextField, JComboBox, dan sebagainya.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan, baik dari segi materi maupun

analisis. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat serta pemahaman yang lebih mendalam bagi pembaca.

Padang, 20 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	1
BAB II PEMBAHASAN	
2.1 Dasar Teori	2
2.2 Langkah Praktikum	2
2.3 Kode Program.....	2
BAB III KESIMPULAN	
3.1 Kesimpulan.....	6
3.2 Saran.....	6
DAFTAR PUSTAKA	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak, aspek interaksi antara pengguna dan komputer sangatlah penting. Sebelum adanya GUI (*Graphical User Interface*), program umumnya berbasis teks atau CLI (*Command Line Interface*). Metode ini kurang efisien dan tidak ramah bagi pengguna awam karena mengharuskan pengguna menghafal sintaks dan perintah tertentu.

Adanya Java sebagai bahasa pemrograman, menyediakan fasilitas lengkap untuk membangun aplikasi berbasis GUI melalui pustaka AWT (*Abstract Window Toolkit*) dan *Swing*. *Swing* menawarkan komponen yang lebih ringan dan fleksibel dibanding sebelumnya. Praktikum ini difokuskan pada pemanfaatan komponen *Swing* seperti *JFrame*, *JPanel*, dan *JButton* untuk membuat aplikasi operator aritmatika yang menarik, menggantikan metode input berbasis konsol yang kaku.

1.2 Tujuan Praktikum

Adapun tujuan dari praktikum ini adalah:

- 1) Memahami konsep dasar pemrograman berbasis antarmuka grafis (GUI) di Java.
- 2) Mampu mengimplementasikan kelas-kelas dari paket `javax.swing` dan `java.awt`.
- 3) Memahami mekanisme *Event Handling* pada komponen button.
- 4) Menerapkan logika validasi input dan penanganan kesalahan dalam aplikasi GUI.

1.3 Manfaat Praktikum

- 1) Mahasiswa mampu membuat aplikasi dekstop yang memiliki antarmuka visual.
- 2) Meningkatkan pemahaman tentang logika pemrograman yang dijalankan berdasarkan interaksi pengguna.
- 3) Sebagai dasar untuk pengembangan aplikasi berskala lebih besar yang membutuhkan interaksi pengguna yang kompleks.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Graphical User Interface adalah antarmuka yang memungkinkan pengguna berinteraksi dengan perangkat elektronik melalui ikon grafis dan indikator visual, bukan antarmuka berbasis teks. Pada Java, pembuatan GUI umumnya menggunakan pustaka *Swing* yang merupakan bagian dari Java yang sering digunakan untuk membangun antarmuka pengguna. *Foundation Classes (JFC)*.

Dalam pemrograman Java *Swing*, terdapat beberapa komponen fundamental yang sering digunakan untuk membangun antarmuka pengguna. *JFrame* berfungsi sebagai wadah utama atau kerangka jendela aplikasi yang menampung komponen lainnya. Di dalamnya, *JLabel* digunakan sebagai area untuk menampilkan teks pendek atau gambar yang bersifat statis dan tidak dapat disunting oleh pengguna. Untuk keperluan input data, *TextField* menyediakan fasilitas bagi pengguna untuk mengedit satu baris teks. Interaksi eksekusi program difasilitasi oleh *JButton*, yaitu tombol yang memicu aksi tertentu ketika diklik. Selain itu, untuk komunikasi interaktif seperti memberikan informasi atau meminta masukan sederhana, digunakan *JOptionPane* yang merupakan kotak dialog standar.

2.2 Langkah Praktikum

- 1) Membuka aplikasi IDE (*Integrated Development Environment*) seperti Eclipse.
- 2) Membuat *project* baru dan *package* dengan nama pekan8_2511532023.
- 3) Membuat *class* baru bernama *OperatorAritmatikaGUI_2511532023* yang mewarisi (*extends*) kelas *JFrame*.
- 4) Menuliskan kode program untuk mendesain antarmuka (label, text field, combo box, button) dan logika aritmatika.
- 5) Menjalankan program (*Run*) dan melakukan uji coba perhitungan serta validasi kesalahan.

2.3 Kode Program

- 1) *Import Library* dan Deklarasi Kelas

```
1. package pekan8_2511532023;  
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 java.awt.Font;
```

```

10. import javax.swing.JTextField;
11. import javax.swing.JComboBox;
12. import javax.swing.JButton;
13. import javax.swing.DefaultComboBoxModel;
14. import javax.swing.JOptionPane;
15. import java.awt.event.ActionListener;
16. import java.awt.event.ActionEvent;
17.
18. public class OperatorAritmatikaGUI_2511532023 extends JFrame {
19.     private static final long serialVersionUID = 1L;
20.     private JPanel contentPane;
21.     private JTextField txtBil1;
22.     private JTextField txtBil2;
23.     private JTextField txtHasil;...
24. }

```

Kode Program 2.1

Program diawali dengan mengimpor pustaka yang diperlukan. Kelas utama didefinisikan sebagai turunan dari JFrame, yang menjadikannya sebuah jendela aplikasi. Variabel komponen dideklarasikan di tingkat kelas agar dapat diakses oleh seluruh metode internal.

2) Metode *Helper*

```

1.     private void pesanPeringatan(String pesan) {
2.         JOptionPane.showMessageDialog(this, pesan,
3. "Peringatan", JOptionPane.WARNING_MESSAGE);}
4.     private void pesanError(String pesan) {
5.         JOptionPane.showMessageDialog(this, pesan,
6. "Kesalahan", JOptionPane.ERROR_MESSAGE);}

```

Kode Program 2.2

Metode *helper* dibuat untuk membungkus fungsi JOptionPane. Hal ini bertujuan agar kode lebih bersih dan modular saat menampilkan pesan peringatan atau kesalahan kepada pengguna..

3) Kontruktor dan Tata Letak

Di dalam konstruktor, properti jendela diatur menggunakan setBounds untuk posisi dan ukuran. Tata letak diatur ke *null* (*Absolute Layout*), yang memungkinkan penempatan komponen secara manual menggunakan koordinat X dan Y. Komponen input seperti txtBil1 dan txtBil2 diinisialisasi, serta JComboBox diisi dengan model operator aritmatika (+, -, *, /, %) .

D. Logika Utama dan *Event Handling*

```

1. Proses.addActionListener(new ActionListener() {
2.     int hasil;
3.     public void actionPerformed(ActionEvent e) {

```

```

4.         if(txtBil1.getText().trim().isEmpty())
{pesanPeringatan ("Silahkan Input Bilangan 1");
5.         }else if (txtBil2.getText().trim().isEmpty())
{pesanPeringatan("Silahkan Input Bilangan 2");
6.         }else
7.         if(txtBil2.getText().trim().startsWith("0"))
{pesanPeringatan("Bilangan 2 tidak boleh angka 0");
8.         }
9.         else {
10.            try {
11.                int a = Integer.parseInt(txtBil1.getText());
12.                int b = Integer.parseInt(txtBil2.getText());
13.                int c = cbOperator.getSelectedIndex();
14.
15.                if (c==0) {hasil = a+b;}
16.                if (c==1) {hasil = a-b;}
17.                if (c==2) {hasil = a*b;}
18.                if (c==3) {hasil = a/b;}
19.                if (c==4) {hasil = a%b;}
20.                txtHasil.setText(String.valueOf(hasil));
21.            } catch (NumberFormatException ex) {pesanError
("Bilangan 1 dan Bilangan 2 harus Angka");
22.            }
23.        }
24.    }
25.    });
26. }
27. }

```

Kode Program 2.3

Alur logika program diawali dengan validasi input menggunakan metode `.isEmpty()` untuk memastikan kotak isian tidak kosong, serta pengecekan khusus pada bilangan kedua guna mengantisipasi kesalahan pembagian dengan nol. Data masukan yang bertipe String kemudian dikonversi menjadi integer menggunakan `Integer.parseInt()`, yang dilindungi oleh mekanisme *Exception Handling* (blok `try-catch`) untuk menangani potensi `NumberFormatException` akibat input non-numerik. Setelah data tervalidasi, program menentukan jenis operasi aritmatika berdasarkan indeks yang dipilih pada `JComboBox` dan menampilkan hasil perhitungan kembali ke antarmuka pengguna.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum pembuatan aplikasi "Operator Aritmatika GUI", dapat disimpulkan bahwa implementasi antarmuka grafis menggunakan *Java Swing* menawarkan kemudahan pengoperasian yang signifikan bagi pengguna dibandingkan antarmuka berbasis teks (CLI). Secara teknis, kelas *JFrame* memegang peranan vital sebagai kerangka utama aplikasi, yang didukung oleh komponen seperti *JTextField* dan *JButton* untuk memfasilitasi proses input dan output. Selain itu, stabilitas program sangat bergantung pada penerapan penanganan eksepsi (*Exception Handling*) yang tepat guna mengantisipasi data masukan yang tidak valid, serta pemanfaatan *JOptionPane* yang terbukti efektif dalam menyampaikan notifikasi kesalahan atau peringatan secara visual kepada pengguna.

3.2 Saran

Demi pengembangan aplikasi yang lebih baik di masa mendatang, terdapat beberapa hal yang perlu diperhatikan. Penggunaan tata letak absolut (*Absolute Layout*) melalui `contentPane.setLayout(null)` sebaiknya dihindari pada aplikasi yang lebih kompleks karena rentan menyebabkan tampilan yang tidak proporsional saat ukuran jendela diubah; sebagai alternatif, disarankan untuk memanfaatkan *Layout Manager* yang lebih fleksibel seperti *FlowLayout* atau *GridLayout*. Selanjutnya, logika validasi terkait pembagian dengan nol sebaiknya diterapkan secara spesifik hanya pada saat operator pembagian atau modulo dipilih, alih-alih digeneralisasi untuk seluruh operasi. Terakhir, penerapan konvensi penamaan variabel komponen yang konsisten dan deskriptif sangat dianjurkan untuk memudahkan proses pemeliharaan kode program.

DAFTAR PUSTAKA

- [1] B. Hartono, Pemrograman Java untuk Pemula. Semarang: Yayasan Prima Agus Teknik, 2022.
- [2] Oracle, "The Java Tutorials: Creating a GUI with JFC/Swing," [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/uiswing/>. [Diakses: 20-Nov-2025].
- [3] J. Ismail, "Laporan Praktikum 8 Java GUI," 2019. [Daring]. Tersedia pada: <https://juniismail.blogspot.com/2019/03/laporan-praktikum-8-java-gui.html>. [Diakses: 20-Nov-2025].
- [4] Guru99, "Java Swing GUI: Programming Examples," [Daring]. Tersedia pada: <https://www.guru99.com/id/java-swing-gui.html>. [Diakses: 20-Nov-2025].
- [5] Scribd, "Modul 6 Graphical User Interface," [Daring]. Tersedia pada: <https://www.scribd.com/document/868224240/MODUL-6-GRAPHICAL-USER-INTERFACE-1>. [Diakses: 19-Nov-2025].