

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 penulis dapat menyelesaikan laporan praktikum dengan judul “Pemrograman GUI” ini tepat waktu.

Laporan ini disusun sebagai salah satu syarat untuk memenuhi tugas mata kuliah Praktikum Algoritma dan Pemrograman. Laporan ini bertujuan untuk menguji pemahaman dan implementasi *Graphical User Interface* (GUI) menggunakan pustaka *Swing* dan AWT untuk membangun sebuah aplikasi kalkulator sederhana.

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, 27 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 Analisis Kode Program	3
BAB III KESIMPULAN	
3.1 Kesimpulan	6
3.2 Saran	6
DAFTAR PUSTAKA	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Saat ini, interaksi antara pengguna dan komputer tidak lagi hanya bergantung pada CLI. Pengguna kini menuntut antarmuka yang lebih intuitif, visual, dan mudah digunakan. Untuk memenuhi tuntutan tersebut, GUI kini digunakan sebagai standar dalam pengembangan aplikasi *desktop*. GUI memungkinkan pengguna untuk berinteraksi dengan perangkat lunak melalui elemen visual seperti jendela, ikon, tombol, dan menu, yang membuat pengoperasian program menjadi lebih efisien.

Bahasa pemrograman Java menyediakan dukungan yang kuat pada pengembangan GUI melalui pustaka AWT (*Abstract Window Toolkit*) dan *Swing*. Secara khusus, *Swing* menawarkan komponen yang lebih ringan dan fleksibel untuk membangun antarmuka yang kompleks. Praktikum ini berfokus pada penerapan konsep GUI tersebut dalam studi kasus pembuatan aplikasi kalkulator.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini adalah:

- 1) Memahami konsep dasar dan struktur komponen GUI dalam bahasa pemrograman Java (JFrame, JPanel, JButton, JTextField).
- 2) Mampu merancang antarmuka pengguna (*User Interface*) menggunakan pustaka *javax.swing* dan *java.awt*.
- 3) Mengimplementasikan *Event Handling* menggunakan *ActionListener* untuk merespons interaksi pengguna.

1.3 Manfaat Praktikum

Manfaat yang diperoleh dari praktikum ini adalah:

- 1) Mahasiswa mampu beralih dari konsep pemrograman berbasis teks ke pemrograman berbasis visual.
- 2) Meningkatkan pemahaman tentang alur kerja program berbasis *event-driven*.
- 3) Sebagai landasan untuk mengembangkan aplikasi *desktop* yang lebih kompleks dan ramah pengguna di masa depan.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Graphical User Interface atau yang disingkat dengan GUI adalah antarmuka pada sistem operasi atau komputer yang menggunakan menu grafis, ikon, dan *mouse* untuk berinteraksi dengan pengguna, menggantikan metode perintah teks manual. Dalam Java, GUI dibangun di atas kelas-kelas dasar yang disediakan oleh JRE (*Java Runtime Environment*).

Salah satu pustaka utama yang digunakan adalah *Java Swing*, yang merupakan bagian dari *Java Foundation Classes* (JFC) yang digunakan untuk membuat aplikasi berbasis jendela. Berbeda dengan AWT yang bersifat *heavyweight* karena bergantung pada platform sistem operasi, komponen *Swing* ditulis sepenuhnya dalam bahasa Java sehingga memiliki karakteristik *platform-independent* dan *lightweight*. Pustaka ini menyediakan berbagai kelas utama seperti *JFrame* sebagai wadah utama atau jendela aplikasi, *JButton* yaitu komponen tombol yang dapat di klik dan menghasilkan aksi, kemudian *JTextField* yang merupakan area untuk menginputkan dan menampilkan satu baris teks, serta *JPanel* untuk mengelompokkan komponen lain.

Agar komponen-komponen visual tersebut dapat berfungsi secara interaktif, diperlukan mekanisme *Event Handling*. Dalam bahasa Java, mekanisme ini diimplementasikan dengan menambahkan *listener* pada komponen, di mana *interface* yang paling umum digunakan adalah *ActionListener* yang memiliki metode *actionPerformed* untuk menangani respons terhadap aksi pengguna, seperti saat klik pada tombol.

2.2 Langkah Praktikum

Langkah-langkah yang dilakukan untuk membuat aplikasi Kalkulator adalah sebagai berikut:

- 1) Menyiapkan perangkat yang telah dilengkapi dengan JDK dan sebuah IDE seperti Eclipse.
- 2) Membuka IDE dan membuat sebuah *project* Java baru bernama *pekan9_2511532023* dan kelas dengan nama *Kalkulator_2511532023*.
- 3) Mengimpor pustaka yang diperlukan, yaitu *java.awt* untuk *font* dan *event* dan *java.swing* untuk komponen visual.
- 4) Mendeklarasikan komponen GUI utama seperti *JFrame* dan *JTextField*, serta variabel logika (*first*, *second*, *result*, *operation*).
- 5) Menginisialisasi *JFrame* dengan ukuran tertentu dan layout absolut untuk penempatan komponen secara manual.
- 6) Membuat dan menempatkan tombol-tombol angka (0-9), tombol operasi (+, -, *, /, %), tombol *equal* (=), tombol *clear* (C), dan tombol *backspace*, lalu mengubah variabel masing-masing tombol sesuai nama yang sudah dibuat.

- 7) Menambahkan *ActionListener* pada setiap tombol untuk mendefinisikan logika yang terjadi saat tombol ditekan.
- 8) Menjalankan aplikasi melalui *main method* menggunakan *EventQueue.invokeLater* untuk keamanan *thread*.

2.3 Analisis Kode Program

Berikut adalah analisis dari kode program *Kalkulator_2511532023.java* yang telah diimplementasikan:

A. Deklarasi dan Inisialisai Variabel

```
1. public class Kalkulator_2511532023 {
2.
3.     private JFrame frame;
4.     private JTextField textField;
5.
6.     double first;
7.     double second;
8.     double result;
9.     String operation;
10.    String answer;
11.    // ...
12. }
```

Kode Program 2.1

Pada bagian ini, program mendefenisikan komponen utama GUI. Variabel *first*, *second*, dan *result* bertipe *double* digunakan untuk menyimpan angka yang dioperasikan, memungkinkan perhitungan bilangan desimal. Variabel *operation* digunakan untuk menyimpan jenis operator matematika yang dipilih pengguna, dan variabel *answer* untuk menyimpan jawaban dari hasil kalkulasi yang diinputkan pengguna.

B. Pengaturan Jendela Utama (JFrame)

```
1.     private void initialize() {
2.         frame = new JFrame();
3.         frame.setBounds(100, 100, 308, 383);
4.         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
5.         frame.getContentPane().setLayout(null);
```

Kode Program 2.2

Objek frame diinisiliasi sebagai jendela utama aplikasi. Metode *setLayout(null)* menunjukkan pengguna *Absolute Layout*, di mana posisi setiap komponen (tombol dan layar) ditentukan secara manual menggunakan koordinat x dan y. Meskipun fleksibel, pendekatan ini mengharuskan presisi dalam menentukan posisi *setBounds*.

C. Logika Input Angka

```
1.         JButton btn7 = new JButton("7");
2.         btn7.addActionListener(new ActionListener() {
3.             public void actionPerformed(ActionEvent
4. e) {
5.                 String number=
6. textField.getText() + btn7.getText();
```

```

5.         textField.setText(number);
6.     }
7. });

```

Kode Program 2.3

Setiap tombol angka seperti angka “7” memiliki *ActionListener*. Ketika diklik, program akan mengambil teks yang sudah ada di layar *textField.getText()*, menggabungkannya dengan angka baru, lalu menapilkannya kembali. Ini memungkinkan pengguna memasukkan angka multidigit.

D. Logika Operasi Aritmatika

```

1. JButton btnPlus_2511532023 = new JButton("+");
2.     btnPlus_2511532023.addActionListener(new
3.     ActionListener() {
4.         public void actionPerformed(ActionEvent e)
5.         {
6.             first=
7.             Double.parseDouble(textField.getText());
8.             textField.setText("");
9.             operation= "+";
10.        }
11.    });

```

Kode Program 2.4

Pada saat tombol operator seperti “+” ditekan, program akan melakukan tiga hal: pertama mengambil angka yang ada di layar dan mengonversinya menjadi tipe *double*, lalu menyimpannya di variabel *first*, kedua mengosongkan layar agar pengguna dapat memasukkan angka kedua, terakhir menyimpan simbol operator ke dalam variabel *operation* untuk digunakan saat perhitungan nanti.

E. Logika Kalkulasi

```

1. btnEqual.setFont(new Font("Tahoma", Font.BOLD, 18));
2.     btnEqual.setBounds(146, 280, 63, 50);
3.     frame.getContentPane().add(btnEqual);
4.
5.     JButton btnSub_2511532023 = new JButton("-");
6.     btnSub_2511532023.addActionListener(new
7.     ActionListener() {
8.         public void actionPerformed(ActionEvent
9.         e) {
10.             first=
11.             Double.parseDouble(textField.getText());
12.             textField.setText("");
13.             operation= "-";
14.        }
15.    });

```

Kode Program 2.5

Ini adalah inti logika kalkulator. Program akan mengambil angka kedua, kemudian menggunakan struktur kontrol *if-else* untuk mengecek operator yang disimpan sebelumnya. Setelah operasi matematika dilakukan, hasilnya diformat menjadi *string* dengan

dua angka di belakang koma (%.2f) dan ditampilkan kembali ke layar.

F. Fitur *Backspace*

Fitur backspace diimplementasikan menggunakan kelas *StringBuilder*. Program menghapus karakter terakhir dari string yang ada di layar menggunakan *deleteCharAt(length - 1)* jika panjang teks lebih dari 0.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilaksanakan, dapat disimpulkan bahwa pemrograman *Graphical User Interface* (GUI) menggunakan Java *Swing* sangat efektif dalam membangun aplikasi interaktif dengan memanfaatkan komponen-komponen visual, seperti *JFrame* sebagai wadah utama dan *JButton* sebagai elemen pemicu aksi. Dalam konsep pemrograman berbasis *event-driven*, penerapan *ActionListener* menjadi elemen yang sangat krusial untuk menangkap dan merespons setiap interaksi pengguna secara dinamis. Selain itu, logika pemrograman dasar, termasuk percabangan *if-else* dan operasi aritmatika, terbukti dapat diintegrasikan secara mulus ke dalam antarmuka visual untuk menghasilkan aplikasi fungsional seperti kalkulator. Ditemukan pula bahwa penggunaan tipe data *double* memberikan tingkat presisi yang jauh lebih baik untuk operasi pembagian dan penanganan bilangan desimal dibandingkan dengan penggunaan tipe data *integer*.

3.2 Saran

Demi pengembangan aplikasi yang lebih baik di masa mendatang, terdapat beberapa saran teknis yang dapat diterapkan. Pertama, disarankan untuk beralih menggunakan *Layout Manager* seperti *GridLayout* atau *BorderLayout* menggantikan *Absolute Layout (null)* agar tata letak aplikasi tetap rapi dan responsif ketika ukuran jendela diubah. Kedua, sangat disarankan untuk mengimplementasikan mekanisme penanganan kesalahan (*Exception Handling*) melalui blok *try-catch* guna mengantisipasi input yang tidak valid atau kesalahan matematis fatal, sehingga mencegah aplikasi berhenti secara mendadak (*crash*). Terakhir, untuk memastikan akurasi logika, proses perbandingan data bertipe string sebaiknya selalu menggunakan metode *.equals()* alih-alih operator *==*, mengingat sifat objek *string* dalam bahasa pemrograman Java.

DAFTAR PUSTAKA

- [1] B. Hartono, *Pemrograman Java untuk Pemula*. Semarang: Yayasan Prima Agus Teknik, 2022.
- [2] Oracle, "Java Swing GUI," [Daring]. Tersedia pada: <https://www.guru99.com/id/java-swing-gui.html>. [Diakses: 20-Nov-2025].
- [3] Newtum, "GUI Programming in Java," [Daring]. Tersedia pada: <https://blog.newtum.com/gui-programming-in-java/>. [Diakses: 27-Nov-2025].
- [4] Dev.to, "Cara Membuat Aplikasi GUI Menggunakan Java Swing," [Daring]. Tersedia pada: <https://dev.to/blogbebasapaaja/cara-membuat-aplikasi-gui-menggunakan-java-swing-4gc8>. [Diakses: 27-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: 20-Nov-2025].