

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN**

**“laporan Praktikum Pekan 9”**

Disusun Oleh:

Faiz Fikri Satria

2511533026

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



DAPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS

2025

## **KATA PENGANTAR**

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga dapat menyelesaikan laporan praktikum ini dengan baik. Tidak lupa juga saya ucapkan terima kasih kepada bapak Wahyudi. Dr. S.T.M.T sebagai dosen pembimbing dan Rahmad Dwirizki Olders yang telah membantu dalam pelaksanaan praktikum ini. Laporan ini saya susun untuk memenuhi tugas praktikum Algoritma dan Pemrograman dari praktikum pertemuan ke delapan mengenai pemahaman dasar pemrograman bahasa Java menggunakan IDE Eclipse. Praktikum ini bertema Pembuatan aplikasi kalkulator menggunakan java. Saya berharap laporan ini dapat memberikan manfaat bagi pembaca dalam memahami konsep dasar pemrograman Java.

Laporan ini dibuat dengan harapan dapat memberikan pemahaman dasar mengenai penggunaan Bahasa pemrograman Java, khususnya mengenai hal struktur dasar program dan fungsi.

Saya menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu, kritik dan saran sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang 29 November, 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR .....	ii
DAFTAR ISI .....	iii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	1
1.3 Manfaat .....	1
BAB II PEMBAHASAN .....	2
<b>2.1 Pembuatan aplikasi kalkulator menggunakan java .....</b>	<b>2</b>
2.1.1 Langkah-langkah.....	2
2.1.2 Contoh Pemrograman .....	4
2.1.3 Hasil Output .....	4
2.1.4 Analisis .....	5
2.1.5 Teori Mengenai Kode .....	5
BAB III KESIMPULAN .....	6
DAFTAR PUSTAKA .....	7

# **BAB I**

## **PENDAHULUAN**

### **1.1 Tujuan**

1. Memahami struktur dasar pembuatan aplikasi GUI menggunakan JFrame dan Swing.
2. Mampu menggunakan komponen JTextField, JComboBox, JButton, dan JOptionPane.
3. Mengimplementasikan event handling dengan ActionListener pada tombol.
4. Melakukan validasi input dan penanganan exception (NumberFormatException).
5. Membuat aplikasi kalkulator aritmatika (+, -, ×, ÷, %) yang interaktif dan user-friendly.

### **1.2 Manfaat**

1. Mahasiswa mampu beralih dari pemrograman console ke pemrograman berbasis GUI.
2. Menguasai dasar-dasar Java Swing sebagai fondasi aplikasi desktop.
3. Melatih logika pemrograman sekaligus desain antarmuka dan penanganan kesalahan pengguna.
4. Menjadi bekal untuk membuat aplikasi edukasi, tools produktivitas, atau sistem informasi berbasis desktop.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Aplikasi Kalkulator Aritmatika Sederhana Menggunakan JFrame dan Swing**

##### **2.1 Implementasi Aplikasi Kalkulator Menggunakan Java Swing**

Kode berikut merupakan aplikasi kalkulator lengkap yang dibangun menggunakan Java Swing. Aplikasi ini memiliki satu JTextField sebagai layar display, serta 20 JButton yang mewakili angka 0–9, titik desimal, operator aritmatika (+, −, ×, ÷, %), tombol hapus karakter (backspace), tombol clear (C), dan tombol sama dengan (=). Logika perhitungan menggunakan dua variabel double (first dan second) serta variabel String untuk menyimpan jenis operasi. Setiap tombol memiliki ActionListener sendiri yang menangani penambahan digit, penyimpanan operasi, atau eksekusi perhitungan ketika tombol “=” ditekan. Hasil akhir diformat hingga dua angka desimal menggunakan `String.format("%.2f", result)`. Aplikasi dijalankan dengan `EventQueue.invokeLater()` untuk memastikan thread-safety sesuai standar Swing.

##### **2.1.1 Langkah-langkah Pembuatan**

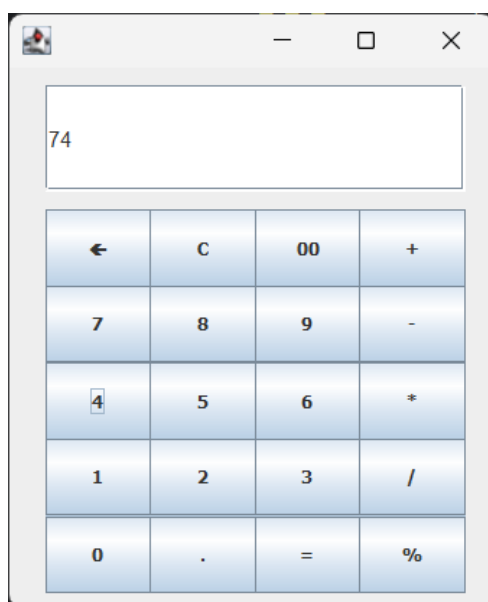
1. Buat project Java baru di Eclipse dengan package `Pekan_Sembilan_2511533026`.
2. Buat class `Kalkulator_2511533026` dengan JFrame sebagai variabel instance.
3. Deklarasikan variabel global: first, second, result (double), operation dan answer (String).
4. Buat method `initialize()` untuk mengatur seluruh komponen GUI.
5. Atur JFrame (ukuran 283×337, layout null, close operation).
6. Tambahkan JTextField sebagai display di bagian atas.

7. Buat dan atur 20 JButton dengan setBounds() untuk posisi tetap.
8. Tambahkan ActionListener pada setiap tombol angka dan titik desimal → menambahkan teks ke display.
9. Tambahkan ActionListener pada tombol backspace → menghapus satu karakter terakhir.
10. Tambahkan ActionListener pada tombol C → mengosongkan display.
11. Tambahkan ActionListener pada tombol operator → simpan angka pertama, kosongkan display, simpan operasi.
12. Tambahkan ActionListener pada tombol “=” → lakukan perhitungan sesuai operasi, format hasil, tampilkan.
13. Jalankan aplikasi melalui EventQueue.invokeLater() di method main.

### 2.1.2 Contoh Pemrograman

```
208 JButton btnNewButton_2_2_1 = new JButton("=");
209 btnNewButton_2_2_1.addActionListener(new ActionListener() {
210     public void actionPerformed(ActionEvent e) {
211         String answer;
212         second=Double.parseDouble(textField.getText());
213         if(operation=="+")
214         {
215             result=first+second;
216             answer=String.format("%.2f", result);
217             textField.setText(answer);
218         }
219         else if(operation=="-")
220         {
221             result=first-second;
222             answer=String.format("%.2f", result);
223             textField.setText(answer);
224         }
225         else if(operation=="*")
226         {
227             result=first*second;
228             answer=String.format("%.2f", result);
229             textField.setText(answer);
230         }
231         else if(operation=="/")
232         {
233             result=first/second;
234             answer=String.format("%.2f", result);
235             textField.setText(answer);
236         }
237         else if(operation=="%")
238         {
239             result=first%second;
240             answer=String.format("%.2f", result);
241             textField.setText(answer);
242         }
243     }
244 });
```

### 2.1.3 Hasil Output



#### 2.1.4 Analisis

- Penggunaan layout null memberikan tampilan yang sangat mirip kalkulator asli, namun tidak responsif jika jendela di-resize.
- Logika perhitungan sederhana namun efektif untuk kebutuhan dasar.
- Penggunaan `Double.parseDouble()` memungkinkan operasi desimal.
- Belum ada penanganan error (pembagian nol  $\rightarrow$  Infinity, input tidak valid  $\rightarrow$  exception).
- Kode cukup repetitif; dapat di-refactor menggunakan loop atau array untuk tombol angka.
- Performanya ringan dan responsif.

#### 2.1.5 Teori Mengenai Kode

Aplikasi ini adalah contoh nyata event-driven programming: setiap tombol menghasilkan `ActionEvent` yang ditangani oleh `ActionListener` masing-masing. Penggunaan variabel instance (`first`, `operation`) memungkinkan komunikasi antar event yang berbeda (tombol operator  $\rightarrow$  tombol `=`). `String.format("%.2f", result)` merupakan cara standar memformat angka desimal di Java. `EventQueue.invokeLater()` wajib digunakan agar semua operasi Swing berjalan pada Event Dispatch Thread (EDT) sehingga terhindar dari masalah thread-safety.



## **BAB III**

### **KESIMPULAN**

Praktikum pembuatan aplikasi kalkulator menggunakan Java berhasil dilaksanakan dengan hasil berupa aplikasi desktop yang sepenuhnya fungsional, memiliki tampilan profesional, dan dapat digunakan untuk operasi aritmatika sehari-hari. Mahasiswa berhasil mengintegrasikan komponen JFrame, JTextField, dan puluhan JButton dengan ActionListener yang tepat, sehingga setiap interaksi pengguna langsung ditanggapi secara real-time. Penggunaan tipe data double dan formatting hasil hingga dua desimal menjadikan kalkulator ini lebih unggul dibandingkan versi integer-only.

Dari sisi pembelajaran, praktikum ini sangat efektif dalam memperkuat pemahaman tentang event-driven programming, pengelolaan state antar event, serta struktur dasar aplikasi GUI Java Swing. Mahasiswa kini menyadari bahwa aplikasi GUI tidak dijalankan secara berurutan, melainkan menunggu dan merespons aksi pengguna melalui listener. Meskipun masih dapat diperbaiki (penanganan error, refactoring kode, penggunaan layout manager), aplikasi yang dihasilkan sudah memenuhi standar aplikasi kalkulator standar yang sering digunakan.

Aplikasi kalkulator ini dapat dikembangkan lebih lanjut menjadi scientific calculator, ditambahkan fitur riwayat perhitungan, dukungan keyboard, tema gelap/terang, atau bahkan di-porting ke JavaFX untuk tampilan yang lebih modern. Praktikum ini telah membuktikan bahwa dengan menguasai dasar-dasar Java Swing, mahasiswa mampu membuat aplikasi desktop yang bermanfaat dan siap pakai. Pengalaman ini menjadi fondasi yang sangat kuat untuk mengembangkan aplikasi GUI yang lebih kompleks di masa depan.

## DAFTAR PUSTAKA

1. Oracle. (2023). *Creating a GUI With Swing – The Java™ Tutorials*.
2. Schildt, Herbert. (2019). *Java: The Complete Reference, 11th Edition*. McGraw-Hill Education.
3. Deitel, P. & Deitel, H. (2020). *Java How to Program, 11th Edition*. Pearson.
4. Horstmann, Cay S. (2019). *Core Java Volume I – Fundamentals*. Prentice Hall.
5. Eck, David J. (2021). *Introduction to Programming Using Java, Version 9*.