

LAPORAN PRATIUM PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

Pemograman GUI 2

Disusun oleh

Nama : Muhammad Yusuf Insy

NIM : 2511532003

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

Asisten Praktikum : Aufan Taufikurrahman



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS TAHUN 2025

Daftar Isi

BAB I PENDAHULUAN	3
1.1 Latar Belakang.....	3
1.2 Tujuan	4
1.3 Manfaat.....	4
BAB II PEMBAHASAN.....	5
2.1 Langkah Kerja	
2.2 Uraian Kode Program	
BAB III KESIMPULAN	10
DAFTAR PUSTAKA	11

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi komputer mendorong kebutuhan pembuatan aplikasi yang mampu memberikan kemudahan dan interaksi yang lebih baik kepada pengguna. Salah satu bentuk aplikasi yang banyak digunakan dalam dunia pemrograman adalah aplikasi berbasis **Graphical User Interface (GUI)**. GUI memungkinkan pengguna berinteraksi dengan program melalui tombol, kotak teks, ikon, dan elemen visual lainnya tanpa harus mengetik perintah secara langsung di konsol. Dengan demikian, aplikasi berbasis GUI menjadi lebih intuitif, mudah dipahami, dan nyaman digunakan oleh berbagai kalangan.

Dalam pembelajaran pemrograman Java, pembuatan aplikasi GUI menjadi salah satu materi penting karena Java menyediakan library lengkap untuk membuat antarmuka grafis, salah satunya adalah **Java Swing**. Swing menyediakan berbagai komponen seperti JButton, JTextField, JFrame, serta berbagai event handler yang dapat mengelola interaksi pengguna. Melalui penggunaan Swing, mahasiswa dapat memahami cara kerja event-driven programming, manajemen layout, serta logika pemrosesan data di balik antarmuka visual.

Penerapan konsep GUI dalam bentuk **program kalkulator sederhana** merupakan salah satu latihan yang efektif untuk memperkuat pemahaman mengenai konsep dasar pemrograman GUI. Kalkulator adalah aplikasi yang umum digunakan dan memiliki operasi yang mudah dipahami, sehingga sangat cocok dijadikan sebagai studi kasus untuk mengimplementasikan elemen-elemen dasar seperti input data, pemrosesan operasi aritmatika, dan penampilan output.

Program kalkulator ini dibangun dengan tujuan memberikan pengalaman langsung kepada mahasiswa dalam mengembangkan aplikasi interaktif menggunakan Java. Melalui pembuatan kalkulator GUI ini, mahasiswa dapat mempelajari bagaimana menghubungkan antarmuka pengguna dengan logika program, bagaimana pengguna berinteraksi dengan komponen GUI, serta bagaimana sebuah aplikasi memproses dan menampilkan hasil secara real-time.

Dengan demikian, pembuatan program kalkulator GUI ini tidak hanya menjadi latihan teknis, tetapi juga memberikan pemahaman mendalam mengenai konsep desain aplikasi, manajemen event, serta penerapan logika perhitungan matematis dalam sebuah program visual. Hal ini diharapkan dapat meningkatkan keterampilan mahasiswa dalam membuat aplikasi berbasis GUI dan mempersiapkan mereka untuk pengembangan aplikasi yang lebih kompleks di masa mendatang.

1.2 Tujuan

1. Memahami penerapan Graphical User Interface (GUI) pada Java
2. Melatih kemampuan mahasiswa dalam membuat aplikasi berbasis event-driven,
3. Mengimplementasikan operasi aritmatika dasar

1.3 Manfaat

1. Meningkatkan kemampuan praktis dalam membangun aplikasi GU
2. Memberikan pengalaman langsung dalam menangani event dan interaksi pengguna
3. Memperkuat pemahaman konsep dasar pemrograman

BAB II PEMBAHASAN

2.1 Langkah Kerja

1. Membuka IDE dan membuat project baru bernama pekan5.
2. Membuat kelas Java: Kalkulator_2511532003
3. Menuliskan kode program sesuai instruksi praktikum.
4. Menjalankan program dengan beberapa input uji.
5. Mencatat hasil dan membuat analisis.

2.2 Uraian Kode Program

Kode program kalkulator_2511532003

1. Struktur Utama Program

Program ini merupakan turunan dari kelas JFrame, sehingga dapat menghasilkan sebuah jendela GUI. Di dalamnya terdapat beberapa komponen seperti:

- JPanel sebagai container utama.
- JTextField sebagai area tampilan input dan output angka.
- Sekumpulan JButton untuk angka (0–9), operator (+, -, *, /, %), serta tombol fungsi seperti C, Backspace, dan =.

2. Input Angka

Tiap tombol angka (0–9 dan 00) memiliki event listener yang akan menambahkan angka tersebut ke dalam textField.

Contoh untuk tombol 7:

```
String number = textField.getText() + btn7.getText();
```

```
textField.setText(number);
```

Ini memastikan pengguna dapat memasukkan lebih dari satu digit.

Cara kerjanya:

- a. Mengambil isi textField saat ini
- b. Menambahkan teks tombol (misalnya "7")
- c. Menampilkan kembali hasil gabungan ke textField

Semua tombol angka menggunakan pola penanganan event yang sama.

3. Penyimpanan Data Operasi

Ketika pengguna menekan tombol operator (+, -, *, /, %), program akan:

- Mengambil nilai pertama (first) dari textField.
- Mengosongkan kembali textField untuk input angka kedua.
- Menyimpan jenis operasi pada variabel operation.

Contoh pada operasi tambah:

```
first = Double.parseDouble(textField.getText());
```

```
textField.setText("");
```

```
operation = "+";
```

- first = angka sebelum operator
- operation = operator yang dipilih
- second = angka setelah operator (diambil nanti saat "=" ditekan)

4. Perhitungan Hasil Operasi

Saat tombol "=" ditekan:

- Program mengambil nilai kedua (second).
- Mengecek jenis operasi yang tersimpan.
- Melakukan perhitungan sesuai operator.
- Menampilkan hasil pada textField.

Contoh logika penjumlahan:

```
if (operation == "+") {
```

```
    result = first + second;
```

```
    answer = String.format("%.2f", result);
```

```
textField.setText(answer);  
  
}
```

Format hasil ditampilkan dengan dua angka di belakang koma.

5. Tombol Tambahan

- Backspace (B)
Menghapus satu karakter terakhir pada `textField` menggunakan `StringBuilder`.
- Clear (C)
Menghapus seluruh isi teks
- Titik (.)
Digunakan untuk memasukkan bilangan desimal.

6. Layout Dan tampilan GUI

Program menggunakan **layout null**, sehingga posisi setiap tombol diatur menggunakan `setBounds()`.

Misalnya:

```
btn7.setBounds(10, 133, 55, 37);
```

7. Alur Kerja Program

- Pengguna menjalankan aplikasi
- Pengguna menekan tombol angka → ditampilkan pada `textField`
- Pengguna memilih operator → nilai pertama disimpan
- Pengguna memasukkan nilai kedua
- Pengguna menekan tombol “=” → program menghitung

- Hasil ditampilkan pada textField
- Pengguna dapat melanjutkan dengan operasi lain atau menekan C untuk menghapus

BAB III KESIMPULAN

Berdasarkan hasil praktikum dan analisis terhadap program kalkulator berbasis GUI yang dibuat menggunakan Java Swing, dapat disimpulkan bahwa mahasiswa berhasil memahami dan menerapkan konsep dasar dari pemrograman antarmuka grafis. Pembuatan kalkulator ini menunjukkan bagaimana komponen-komponen GUI seperti JFrame, JPanel, JButton, dan JTextField dapat digunakan untuk membangun aplikasi yang interaktif dan mudah digunakan.

Melalui implementasi tombol angka, operator, serta fitur tambahan seperti backspace dan clear, mahasiswa dapat mempraktikkan konsep event-driven programming, di mana setiap aksi pengguna akan memicu proses tertentu pada program. Selain itu, proses penyimpanan nilai pertama, pemilihan operator, dan perhitungan hasil menunjukkan bahwa logika pemrograman dapat diintegrasikan dengan antarmuka visual secara efektif.

Pembuatan kalkulator ini juga membantu mahasiswa memahami bagaimana mengatur tata letak komponen menggunakan layout null dan memposisikan setiap elemen secara manual. Meskipun sederhana, aplikasi ini memberikan gambaran nyata tentang bagaimana sebuah program GUI bekerja, mulai dari input pengguna hingga proses produksi output.

Secara keseluruhan, praktikum ini memberikan pemahaman yang lebih mendalam tentang cara membangun aplikasi berbasis GUI di Java serta meningkatkan kemampuan mahasiswa dalam merancang antarmuka, menangani event, dan mengimplementasikan logika perhitungan ke dalam sebuah program interaktif. Aplikasi kalkulator yang dihasilkan pun dapat berfungsi dengan baik dan memenuhi tujuan pembelajaran yang telah ditetapkan.

DAFTAR PUSTAKA

Deitel, P. J., & Deitel, H. M. (2014). *Java How to Program (10th Edition)*. Pearson Education.

Schildt, H. (2018). *Java: The Complete Reference (11th Edition)*. McGraw-Hill Education.

Lemay, L., & Cadenhead, R. (2011). *Java Programming (4th Edition)*. Sams Publishing.