

**LAPORAN PRATIKUM**  
**PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**  
**“PROJEK KALKULATOR PADA JAVA SWING GUI APPLICATION”**

**disusun Oleh:**

**AFIF RAHMAN SALEH**

**NIM 251153105**

**Dosen Pengampu: Dr. WAHYUDI, S.T, M.T**

**Asisten Pratikum: AUFAN TAUFIQURRAHAMAN**



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**

**TAHUN**

**2025**

## DAFTAR ISI

DAFTAR ISI .....	i
KATA PENGANTAR.....	ii
BAB I.....	1
PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan.....	1
1.3 Manfaat.....	2
BAB II.....	3
PEMBAHASAN .....	3
2.1 Penjelasan Langkah-Langkah .....	3
2.2 Analisis Hasil .....	6
BAB III.....	8
KESIMPULAN .....	8
3.1 Kesimpulan.....	8
DAFTAR PUSTAKA.....	9

## **KATA PENGANTAR**

Puji syukur kami panjatkan atas kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya laporan praktikum ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai bentuk penugasan dalam kegiatan praktikum mata kuliah Algoritma dan Pemograman khususnya pada materi “Projek Kalkulator Pada Java Swing Gui Application “.

Dalam penyusunan laporan ini, penulis berusaha untuk menjelaskan secara sistematis mulai dari latar belakang, tujuan, dan manfaat praktikum, hingga pembahasan kode program, analisis hasil, serta kesimpulan dan saran untuk pengembangan lebih lanjut. Materi yang dibahas diharapkan dapat menjadi referensi bagi mahasiswa atau praktikan dalam memahami dan mengimplementasikan algoritma perulangan secara efektif.

Padang, 2025

Penulis

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Saat ini User Interface (UI) sangat banyak digunakan dalam aplikasi komputer. Aplikasi yang memiliki User Interface cenderung lebih disukai oleh pengguna, karena dapat memudahkan pembuatan aplikasi tanpa menghabiskan banyak waktu menulis dalam kode. Salah satu UI pada bahasa pemrograman yang kita pelajari saat ini adalah Window Builder.

Window Builder adalah sebuah plugin untuk IDE Eclipse yang memungkinkan pengembang untuk merancang antarmuka grafis secara drag-and-drop. Window Builder menyediakan editor visual yang memungkinkan pengguna untuk menempatkan komponen GUI (seperti tombol, text field, combo box, dan label) pada form secara langsung, sambil secara otomatis menghasilkan kode Java yang sesuai di belakang layar. Hal ini sangat membantu dalam mempercepat proses pengembangan, mengurangi kesalahan penulisan kode, dan memudahkan pemahaman struktur hubungan antar komponen.

Dalam praktikum ini, Window Builder dimanfaatkan untuk membangun aplikasi kalkulator sederhana yang mampu melakukan operasi dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulo (+, -, \*, /, %). Aplikasi ini dirancang sebagai simulasi dari sistem input-output yang umum ditemukan dalam aplikasi desktop.

### **1.2 Tujuan**

1. Memahami konsep dasar pemrograman berbasis GUI menggunakan Java Swing.
2. Mampu mengaplikasikan event handling untuk menangkap aksi pengguna seperti klik tombol angka dan operator matematika.

3. Mampu mengelola input dari pengguna, melakukan operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, pembagian, dan modulus), serta menampilkan hasilnya.
4. Melatih kemampuan dalam penggunaan variabel global, kontrol alur program (if-else), dan manipulasi string dalam konteks GUI.

### **1.3 Manfaat**

1. Memperkuat pemahaman terhadap konsep pemrograman berorientasi objek.
2. Memberikan gambaran awal bagaimana aplikasi desktop dibuat dari awal sampai akhir.
3. Melatih logika pemrograman dalam menangani input, proses, dan output secara terstruktur.

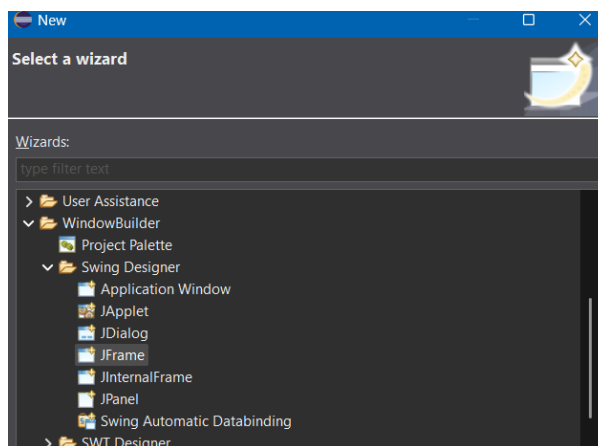
## BAB II

### PEMBAHASAN

#### 2.1 Penjelasan Langkah-Langkah

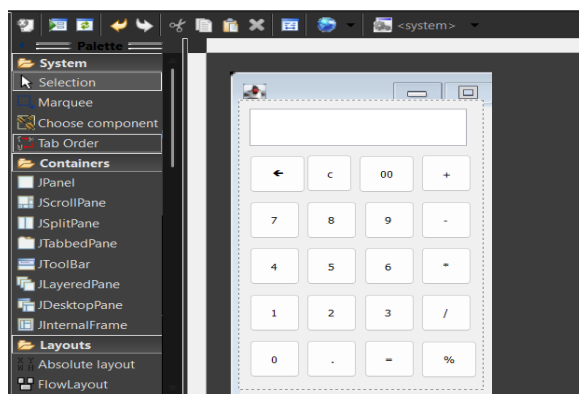
Pembuatan program kalkulator ini melibatkan beberapa tahapan penting, mulai dari pembuatan kelas JFrame, mendesain bentuk aplikasi, serta mengkodekan programnya. Berikut langkah-langkahnya :

1. Buat kelas JFrame pada windowbuilder, lalu beri nama program tersebut.



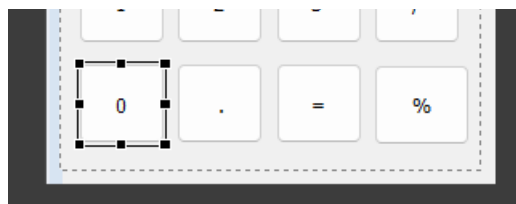
Gambar 2.1

2. Setelah itu bikin desain aplikasi kalkulator menggunakan tools seperti JTextField seperti contoh di bawah ini.



Gambar 2.2

3. Bikin beberapa variabel global untuk menyimpan nilai pertama (first), nilai kedua (second), hasil operasi (result), jenis operasi (operation), dan string hasil (answer).
4. Klik masing masing JButton sebanyak 2 kali untuk memasukkan perintah ke dalam tombol tersebut (Action Listener).
  - a. Untuk tombol angka dan titik bikin program untuk menambahkan karakter yang sudah diklik ke tesk yang sudah ada di textField seperti dibawah ini.



Gambar 2.3

```

JButton btn0 = new JButton("0");
btn0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String number = textField.getText()+btn0.getText();
        textField.setText(number);
    }
}

```

Kode program 2.1

- b. Untuk tombol C bikin perintah untuk mengosongkan seluruh isi textField.

```

JButton btnc = new JButton("C");
btnc.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
    }
}

```

- c. Untuk tombol backSpace gunakan stringBuilder untuk menghapus satu karakter terakhir dari textField.

```

btnplus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String backSpace=null;
        if (textField.getText().length()>0) {
            StringBuilder str = new StringBuilder(textField.getText());
            str.deleteCharAt(textField.getText().length()-1);
            backSpace = str.toString();
            textField.setText(backSpace);
        }
    }
});

```

- d. Untuk setiap tombol operasi seperti (+, -, <, \*, /, %) gunakan program untuk mengonversi nilai saat ini di textField menjadi double dan menyimpannya ke variabel first. Setelah itu kosongkan textField untuk input angka kedua. Lalu simpan jenis operasi ke dalam variabel operation.

```

JButton btnplus_25111531005 = new JButton("+");
btnplus_25111531005.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="+";
    }
});

```

- e. Dan untuk tombol "=" gunakan program untuk membaca nilai kedua dari textField lalu lakukan operasi aritmatika sesuai dengan nilai operation. Setelah itu tampilkan hasil dengan format dua angka di belakang koma menggunakan String.format("%.2f", result).

```

JButton btnequals_2511531005 = new JButton("=");
btnequals_2511531005.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String answer;
        second=Double.parseDouble(textField.getText());
        if(operation == "+") {
            result=first+second;
            answer=String.format("%.2f", result);
            textField.setText(answer);
        }
        else if(operation == "-")
        {
            result=first-second;
            answer=String.format("%.2f", result);
            textField.setText(answer);
        }
        else if(operation == "*")
        {
            result=first*second;
            answer=String.format("%.2f", result);
            textField.setText(answer);
        }
        else if(operation == "/")
        {
            result=first/second;
            answer=String.format("%.2f", result);
            textField.setText(answer);
        }
        else if(operation == "%")
        {
            result=first%second;
            answer=String.format("%.2f", result);
            textField.setText(answer);
        }
    }
}

```

## 2.2 Analisis Hasil

Hasil pengujian terhadap program kalkulator GUI berbasis Java menunjukkan bahwa aplikasi telah mampu menjalankan fungsi dasar perhitungan aritmetika dengan baik. Interface pengguna yang dibangun menggunakan komponen Swing seperti JTextField dan JButton memungkinkan pengguna memasukkan angka dan operator melalui klik tombol. Setiap tombol angka berfungsi sebagaimana mestinya, yaitu menambahkan digit yang sesuai ke layar input secara berurutan, sehingga pengguna dapat menyusun bilangan sesuai keinginannya. Operasi matematika seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus juga telah diimplementasikan dengan mekanisme penyimpanan nilai pertama, pemilihan operator, input nilai kedua, dan perhitungan saat tombol "=" ditekan. Hasil perhitungan ditampilkan dalam format dua angka di belakang koma, yang membuat tampilan output lebih rapi.

Contoh output aplikasi dengan contoh operasi  $7*8$ :



## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Berdasarkan pelaksanaan dan analisis terhadap praktikum pembuatan kalkulator sederhana berbasis GUI menggunakan Java Swing, dapat disimpulkan bahwa tujuan praktikum telah tercapai. Program berhasil diimplementasikan dengan memanfaatkan komponen-komponen GUI seperti JFrame, JTextField, dan JButton, serta mekanisme event handling melalui ActionListener untuk merespon interaksi pengguna. Setiap tombol angka, operator, dan fungsi kontrol (seperti hapus dan backspace) telah berfungsi sesuai dengan perannya masing-masing, memungkinkan pengguna untuk memasukkan data, memilih operasi aritmetika, dan melihat hasil perhitungan secara langsung pada layar tampilan.

Logika program yang dibangun mampu menangani operasi matematika dasar dengan hasil yang ditampilkan dalam format dua angka desimal, sehingga memberikan tampilan output yang rapi. Selain itu, penggunaan variabel global untuk menyimpan nilai pertama, nilai kedua, dan jenis operasi menunjukkan penerapan konsep pemrograman berorientasi objek dan alur kendali yang terstruktur. Meskipun demikian, program ini masih memiliki keterbatasan, terutama dalam hal validasi input dan penanganan kesalahan (error handling), seperti ketika pengguna menekan operator tanpa memasukkan angka terlebih dahulu atau melakukan pembagian dengan nol.

Secara keseluruhan, praktikum ini telah memberikan pemahaman yang mendalam mengenai pengembangan aplikasi desktop berbasis GUI di Java, khususnya dalam hal perancangan antarmuka, manajemen event, dan integrasi antara antarmuka visual dengan logika komputasi. Pengalaman ini menjadi fondasi penting bagi pengembangan aplikasi yang lebih kompleks untuk praktikum praktikum selanjutnya, sekaligus memperkuat kompetensi dalam pemrograman berorientasi objek.

## DAFTAR PUSTAKA

- [1] Oracle Corporation, *Java™ Platform, Standard Edition 8 API Specification*. Redwood City, CA: Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/>
- [2] H. M. Deitel and P. J. Deitel, *Java How to Program*, 11th ed. Hoboken, NJ: Pearson Education, 2017.
- [3] D. Liang, *Introduction to Java Programming and Data Structures: Comprehensive Version*, 12th ed. Hoboken, NJ: Pearson, 2022.
- [4] Oracle Corporation, *Creating a GUI With JFC/Swing*, The Java™ Tutorials. Redwood City, CA: Oracle, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/>
- [5] J. Eckel, *Thinking in Java*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2006.
- [6] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1995.