

LAPORAN PRAKTIKUM

PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

“MEMBUAT KALKULATOR INTERAKTIF DENGAN MENGGUNAKAN
ECLIPSE IDE DAN BAHASA PEMROGRAMAN JAVA”

disusun oleh

Hafizh Habibullah

2511531002

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

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TAHUN 2025

KATA PENGANTAR

Puji syukur kita panjatkan ke hadirat Allah SWT. karena atas rahmat dan karunia-Nya laporan praktikum Algoritma dan Pemrograman ini dapat diselesaikan dengan baik.

Laporan ini disusun untuk memenuhi tugas praktikum pada mata kuliah Algoritma dan Pemrograman dengan topik Membuat Kalkulator Interaktif di *Eclipse IDE* dengan Bahasa Pemrograman Java. Melalui laporan ini, penulis berusaha menjelaskan hasil praktikum yang meliputi penjelasan mengenai Objek yang digunakan, cara kerja, serta hasil output dari program.

Penulis sangat mengharapkan kritik dan saran yang membangun agar dapat menjadi bahan perbaikan di kemudian hari. Semoga laporan ini dapat memberikan manfaat bagi penulis maupun pembaca.

Padang, 29 November 2025

Hafizh Habibullah

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR LAMPIRAN	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat.....	1
BAB 2 PEMBAHASAN.....	2
2.1 Membuat <i>Layout</i> Kalkulator.....	2
2.2 Membuat <i>Program</i> untuk Kalkulator	4
2.3 Penerapan Logika Dasar Program	6
2.4 Hasil Output dari Aplikasi	7
BAB III KESIMPULAN.....	8
3.1 Kesimpulan.....	8
3.2 Saran Pengembangan.....	8
DAFTAR PUSTAKA	9

DAFTAR LAMPIRAN

Gambar 2. 1 - textField utama.....	2
Gambar 2. 2 - Button untuk kalkulator	2
Gambar 2. 3 - Bentuk Layout setelah memasukkan semua tombol.....	3
Gambar 2. 4 - Layout Kalkulator setelah setiap tombol disesuaikan.....	3
Gambar 2. 5 - Kode Program untuk menampilkan Input dari tombol Angka.....	4
Gambar 2. 6 - Jenis variabel yang akan digunakan.....	4
Gambar 2. 7 - Kode Program untuk menampilkan Input dari tombol Operasi.....	5
Gambar 2. 8 - Kode Program untuk menghapus isi textField.....	5
Gambar 2. 9 - - Kode Program untuk menghapus isi textField satu-persatu	6
Gambar 2. 10 - Kode Program untuk menentukan jenis operasi	6
Gambar 2. 11 - Hasil Output dari Kalkulator jika dijalankan	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan perangkat lunak mendorong kebutuhan aplikasi sederhana yang stabil dan mudah digunakan, salah satunya kalkulator. Pembuatan kalkulator dapat memberi pemahaman tentang alur kerja antarmuka, pemrosesan input, hingga pengelolaan logika dasar perhitungan. Pendekatan belajar berbasis video tutorial dipakai karena lebih mudah dipahami secara visual dan memungkinkan *programmer* untuk mengikuti proses pengembangan secara langsung, dari tahap desain hingga implementasi. Praktikum ini juga menjadi dasar untuk melanjutkan ke pengembangan aplikasi yang lebih kompleks.

1.2 Tujuan

Adapun tujuan dari praktikum ini adalah:

1. Menerapkan dasar-dasar pembuatan *interface* aplikasi.
2. Memahami logika pemrosesan operasi aritmetika.
3. Melatih penerapan struktur program yang rapi dan modular.

1.3 Manfaat

Manfaat dari Praktikum ini adalah:

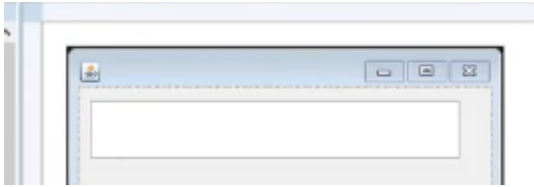
1. Menambah pemahaman tentang proses pengembangan aplikasi sederhana.
2. Meningkatkan keterampilan dalam memadukan *interface*, *event handling*, dan logika program.
3. Menjadi pengalaman dasar sebelum membuat aplikasi dengan fitur yang lebih lengkap.

BAB 2

PEMBAHASAN

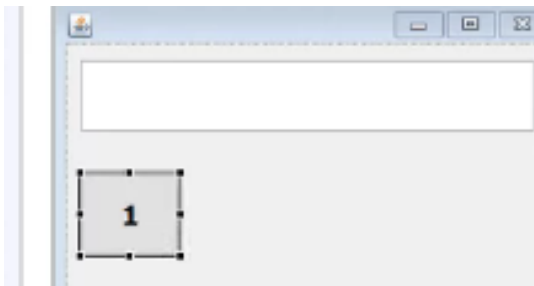
2.1 Membuat *Layout* Kalkulator

Pertama, buatlah *class* dengan menggunakan **WindowBuilder** untuk program kalkulator-nya. Lalu, masuk ke bagian *Design* dan dimulai dengan merubah ukuran *Frame* sesuai yang diinginkan. Agar nantinya komponen yang dimasukkan dapat dipindah-pindah posisinya, ubah mode *Layout* menjadi “Absolute”. Setelah itu, Masukkan komponen *textField* yang nantinya dijadikan bagian yang menampilkan angka yang diinputkan.



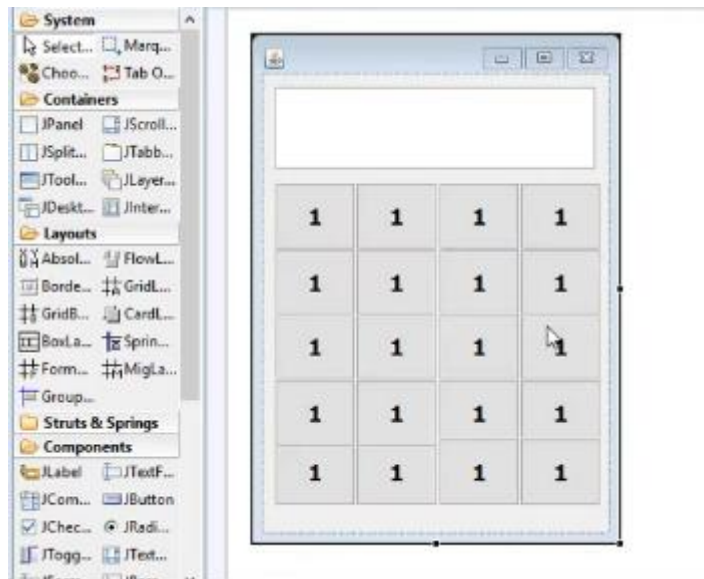
Gambar 2. 1 - textField utama

Setelah itu, tambahkan komponen *Button* yang nantinya digunakan untuk menampilkan angka dan jenis operasi yang dapat digunakan oleh pengguna. Setelah meletakkan *Button* di posisi yang sesuai, atur ukuran teks dan *Font*-nya di bagian *Properties*, lalu ke *font*. Ubah teks menjadi angka 1.



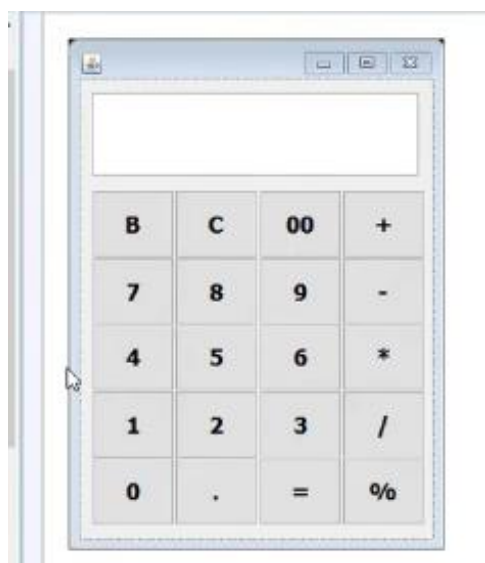
Gambar 2. 2 - Button untuk kalkulator

Setelah itu, *copy* dan *paste* button yang sudah dimasukkan sampai *Button* berjumlah 20 tombol (dengan ukuran 4 kesamping dan 5 kebawah). Ini dilakukan supaya setiap tombol dapat dijadikan angka atau operasi yang berbeda-beda yang dapat digunakan nantinya.



Gambar 2. 3 - Bentuk Layout setelah memasukkan semua tombol

Lalu, ubah semua tombol menjadi angka dan operasi yang sesuai. Angka 0 sampai 9, simbol penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), modulo (%), dan sama dengan (=) untuk menampilkan hasil perhitungan. Tambahkan juga tombol untuk memasukkan titik (.) untuk bilangan desimal, digit 00, menghapus angka yang dimasukkan (B), dan tombol untuk membersihkan papan *textField* (C). Lalu, ubah juga ukuran teks untuk bagian *textField* sesuai dengan ukuran teks tombol-tombol dibawahnya. *Layout* kalkulator akan terlihat seperti ini:



Gambar 2. 4 - Layout Kalkulator setelah setiap tombol disesuaikan

2.2 Membuat *Program* untuk Kalkulator

Pertama, klik dua kali salah satu tombol yang terdapat di *Layout* kalkulator, yang nantinya akan membawa *programmer* ke *Source code* untuk bagian tombol yang diklik tadi. Contohnya, *double-click* tombol 1. Program akan menambahkan program yang menjalankan suatu *action* jika pengguna memberikan *input* dari menekan tombolnya. Setelah itu, masukkan kode berikut agar tombol dapat berfungsi dengan benar:

```
 JButton btn1 = new JButton("1");
 btn1.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
         String number=textField.getText()+btn1.getText();
         textField.setText(number);
     }
 });
 btn1.setFont(new Font("Formula1 Display Bold", Font.PLAIN, 18));
 btn1.setBounds(10, 263, 64, 53);
 frame.getContentPane().add(btn1);
```

Gambar 2. 5 - Kode Program untuk menampilkan Input dari tombol Angka

Setelah itu, lakukan hal yang sama dengan tombol-tombol lainnya. Hanya saja, untuk setiap tombol, ubah **btn1** pada bagian "**btn1.getText()**" menjadi variabel masing-masing tombol angka. Jika dijalankan, dan ditekan tombolnya, Program akan menampilkan angka yang dimasukkan ke bagian *textField*.

```
public class kalkulator_2511531002 {

    private JFrame frame;
    private JTextField textField;

    double first;
    double second;
    double result;
    String operation;
    String answer;
```

Gambar 2. 6 - Jenis variabel yang akan digunakan

Sebelum membuat kode untuk setiap jenis operasi, buat jenis-jenis variabel yang akan digunakan terlebih dahulu. Gunakan jenis *double* untuk variabel-variabel seperti input pertama (**first**), kedua (**second**), dan hasil (**result**). Masukkan juga variabel operasi (**operation**) dan jawaban (**answer**) dengan jenis *String*. Setelah itu,

masukkan kode seperti dibawah. Lalu, lakukan hal yang sama dengan simbol-simbol lainnya. Ubah **operation** menjadi operasi bilangan yang bersangkutan.

```
JButton btnMul = new JButton("*");
btnMul.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="*";
    }
});
btnMul.setFont(new Font("Formula1 Display Bold", Font.PLAIN, 18));
btnMul.setBounds(202, 211, 64, 53);
frame.getContentPane().add(btnMul);
```

Gambar 2. 7 - Kode Program untuk menampilkan Input dari tombol Operasi

Untuk beberapa bagian khusus, seperti *Clear* dan *Backspace*, ada program khususnya. Untuk *Clear*, karena hasil output yang diinginkan adalah menghapus isi dari *textField*-nya, masukkan kode yang akan mengubah *textField* menjadi kosong (*null*).

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

Gambar 2. 8 - Kode Program untuk menghapus isi textField

Sedangkan untuk bagian *Backspace*, ubah output “**B**” menjadi “\Uf0E7”. Ini adalah jenis kode yang akan menampilkan simbol tanda panah (←). Setelah itu, masukkan kode yang menghapus isi *textField* satu persatu, seperti dibawah:

```

JButton btnBack = new JButton("\uF0E7");
btnBack.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);
        }
    }
});

```

Gambar 2. 9 - - Kode Program untuk menghapus isi textField satu-persatu

2.3 Penerapan Logika Dasar Program

Antarmuka Grafis memang sudah dibuat sebelumnya, namun Operasi yang terdapat didalamnya masih belum bisa dijalankan. Hal ini karena program masih belum memiliki Kode yang berisikan logika yang dapat menjalankan Operasi.

```

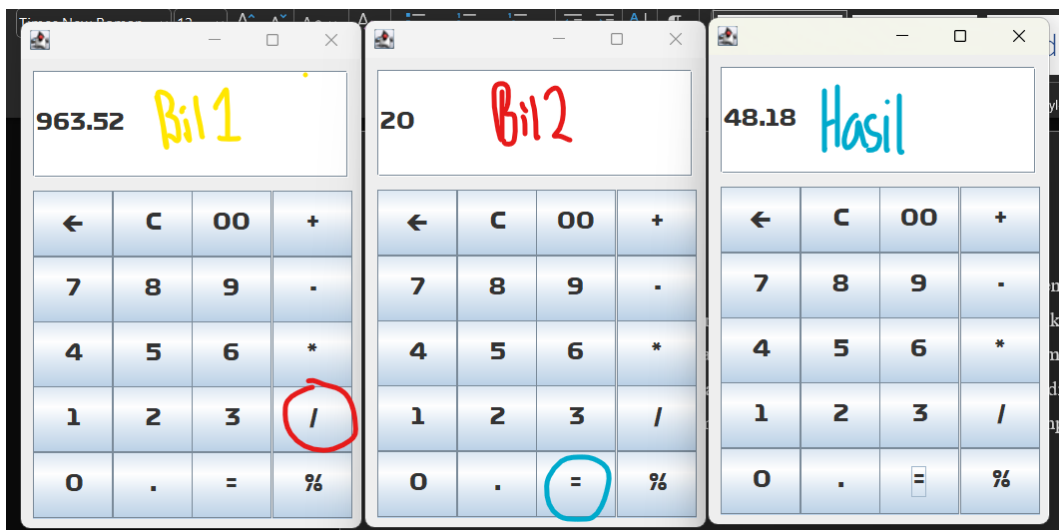
JButton btnEqual = new JButton("=");
btnEqual.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String answer;
        second = Double.parseDouble(textField.getText());
        if (operation == "+") {
            result = first + second;
            answer = 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);
        }
    }
});
btnEqual.setFont(new Font("Formula1 Display Bold", Font.PLAIN, 18));
btnEqual.setBounds(137, 316, 64, 53);
frame.getContentPane().add(btnEqual);

```

Gambar 2. 10 - Kode Program untuk menentukan jenis operasi

Dengan kode program di atas, program dapat menampilkan hasil perhitungan sesuai dengan logika yang ditemukan setelah diperiksa. Contoh, jika operasi yang digunakan adalah “+”, maka program akan menjumlahkan bilangan pertama (**first**) dan bilangan kedua (**second**) dan menyimpannya di variabel **result**. Variabel tersebut kemudian diubah menjadi **String** dan disimpan ke variable **answer**.

2.4 Hasil Output dari Aplikasi



Gambar 2. 11 - Hasil Output dari Kalkulator jika dijalankan

Berikut adalah contoh Hasil Ouput dari Kalkulator jika dijalankan. Pada contoh, dilakukan perhitungan **963,52** dibagi dengan **20**. Setelah menekan tombol **sama dengan (=)**, *textField* di atas akan menampilkan hasil operasi bilangan tersebut, yaitu **48,18**.

BAB III

KESIMPULAN

3.1 Kesimpulan

Praktikum pembuatan kalkulator berhasil menunjukkan bagaimana *interface*, logika perhitungan, dan pengelolaan input dapat digabungkan dalam satu program yang fungsional. Proses mengikuti tutorial video dari *YouTube* membantu memahami alur kerja pengembangan secara bertahap, mulai dari desain tampilan hingga implementasi kode. Praktikum ini menjadi dasar penting untuk memahami struktur aplikasi dan cara menerapkan logika komputasi secara terprogram.

3.2 Saran Pengembangan

Untuk pengembangan praktikum di masa mendatang, beberapa hal berikut dapat dipertimbangkan:

1. Menyediakan modul pendamping yang lebih ringkas agar alur belajar tidak sepenuhnya bergantung pada tutorial video.
2. Menambahkan sesi penjelasan teori dasar sebelum praktik dimulai supaya peserta memahami alasan di balik setiap langkah implementasi.
3. Memberikan contoh variasi proyek kecil lain agar peserta dapat membandingkan struktur dan pendekatan pemrograman yang berbeda.
4. Menambahkan evaluasi mandiri setelah praktikum untuk memastikan setiap langkah telah dipahami dengan baik.
5. Mengadakan sesi umpan balik atau diskusi terbuka untuk menampung kendala yang dialami selama proses pengerjaan.

DAFTAR PUSTAKA

- [1] Oracle, *The Java™ Tutorials*. Oracle Docs, 2024. [Online]. Available: <https://docs.oracle.com/javase/>
- [2] Oracle Corporation, “Creating a GUI With Swing,” *Oracle Java Documentation*, 2025. [Online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/>
- [3] Tech Projects, GUI Calculator in JAVA using Eclipse IDE. (April 3, 2020). Accessed: Nov. 29, 2025. [Online Video]. Available: <https://www.youtube.com/watch?v=mZVD5OssHQM>