

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
DESAIN GUI KALKULATOR

Disusun Oleh:

Khairun Nisa

2511532015

Dosen Pengampu:

DR. WAHYUDI, S.T, M.T

Asisten Pratikum:

Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
2025

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas terselesaikannya laporan praktikum Algoritma dan Pemrograman ini. Laporan ini disusun sebagai dokumentasi hasil kegiatan praktikum yang telah dilaksanakan pada 25 November 2025, dengan pembahasan desain GUI kalkulator.

Penulis mengucapkan terima kasih kepada Bapak Dosen serta asisten laboratorium yang telah membimbing selama praktikum berlangsung. Semoga laporan ini dapat bermanfaat bagi pembaca dan menjadi referensi untuk perkembangan ilmu pengetahuan di bidang pemrograman.

Padang, 28 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum	1
1.3 Manfaat Pratikum	2
BAB II PEMBAHASAN	3
2.1 Desain GUI Kalkulator.....	3
2.1.1 Langkah-Langkah Pembuatan Desain GUI Kalkulator ..	3
2.1.2 Kode Program Desain GUI Kalkulator.....	5
2.1.3 Penjelasan Output Kalkulator	8
BAB III KESIMPULAN	10
3.1 Kesimpulan	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam perkembangan teknologi informasi saat ini, perangkat lunak tidak hanya dituntut untuk membuat logika yang akurat, tetapi juga antarmuka yang berbasis konsol, meskipun mampu menjalankan fungsi komputasi dengan benar, sering kali kurang intuitif dan tidak menarik bagi pengguna awam. Oleh karena itu, GUI (*Graphic User Interface*) menjadi solusi utama dalam merancang aplikasi yang *user-friendly*, visual, dan responsif.

Java menyediakan dukungan kuat untuk pengembangan aplikasi desktop berbasis GUI melalui pusatka seperti Swing dan JavaFX. Dengan bantuan Integrated Development Environment (IDE) seperti Eclipse, khususnya melalui plugin WindowBuilder, mahasiswa dapat merancang tampilan aplikasi secara visual tanpa harus menulis semua kode antarmuka secara manual. Hal ini memungkinkan focus lebih besar pada integritas antara desain visual dan logika pemrograman.

Praktikum Algoritma dan Pemrograman ini dilaksanakan untuk mengenalkan mahasiswa pada proses pembuatan aplikasi kalkulator sederhana berbasis GUI menggunakan Java di Eclipse. Aplkasi ini tidak hanya menampilkan komponen seperti tombol angka, operator aritmatika, dan layar hasil, tetapi juga menerapkan algoritma dasar untuk menangani input pengguna, memvalidasi datam serta melakukan perhitungan matematis secara dinamis.

1.2 Tujuan Praktikum

1. Memahami konsep dasar GUI dalam pemrograman Java menggunakan pustaka Swing,
2. Mampu merancang antarmuka aplikasi secara visual dengan bantuan plugin WindowBuilder di Eclipse.
3. Melatih kemampuan mengintegrasikan desain tampilan GUI dengan logika perograman melalui event handling.

4. Menerapkan algoritma dasar untuk operasi aritmatika
5. Menghasilkan aplikasi kalkulator sederhana yang fungsional, responsive dan mudah digunakan oleh pengguna awam.

1.3 Manfaat Praktikum

1. Memberikan pengalaman nyata dalam membuat program yang tidak hanya berjalan di konsol, tetapi juga memiliki tampilan visual seperti aplikasi sehari-hari.
2. Memperkuat pemahaman bahwa pemrograman yang baik tidak hanya tentang logika, tetapi juga tentang kemudahan penggunaan.
3. Melatih keterampilan menggabungkan desain antarmuka dan logika program dalam satu kesatuan yang utuh.
4. Menjadi fondasi awal untuk pengembangan aplikasi desktop yang lebih kompleks di mata kuliah lanjutan.
5. Meningkatkan kepercayaan diri dalam menggunakan IDE professional (Eclipse) dan alat bantu visual (WindowBuilder) yang sering digunakan di dunia industry.

BAB II

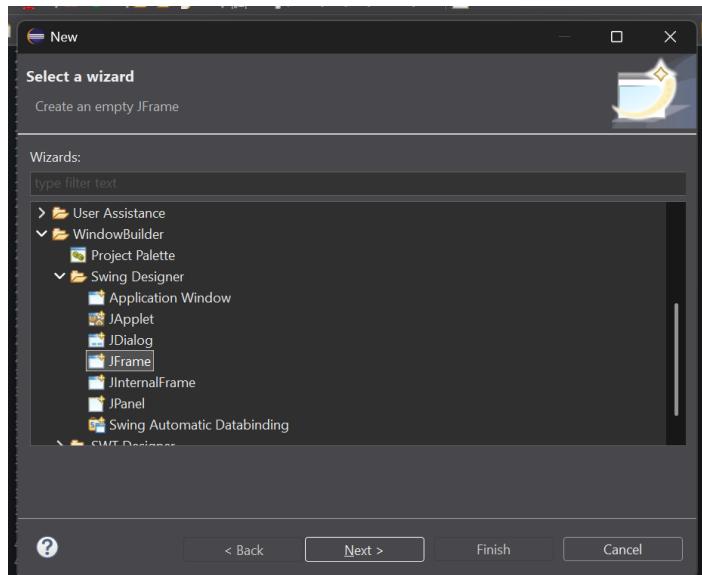
PEMBAHASAN

2.1 Desain GUI Kalkulator

Praktikum pada minggu ini merupakan penyempurnaan dari aplikasi desktop operator aritmatika yang dibuat sebelumnya. Jika pada praktikum sebelumnya pengguna hanya dapat memasukkan dua bilangan dan memilih salah satu operator untuk dihitung sekali jalan, maka pada praktikum kali ini dikembangkan sebuah kalkulator GUI yang lebih interaktif dan fungsional. Perbedaan utama terletak pada penambahan fitur Clear © untuk mengatur ulang layar, serta Backspace untuk menghapus digit terakhir secara langsung. Hal ini menuntut desain logika program yang lebih dinamis, tidak hanya dalam proses perhitungan, tetapi juga dalam pengelolaan input pengguna secara bertahap. Dengan demikian, program tidak lagi bersifat statis, melainkan mampu merespons berbagai aksi pengguna secara real-time, sehingga memberikan pengalaman yang lebih mirip dengan aplikasi nyata. Desain antarmuka dibuat dengan menggunakan WindowBuilder di Eclipse dengan komponen Swing seperti JFrame, JTextField, dan JButton, serta tata letak yang intuitif agar mudah digunakan oleh pengguna awam.

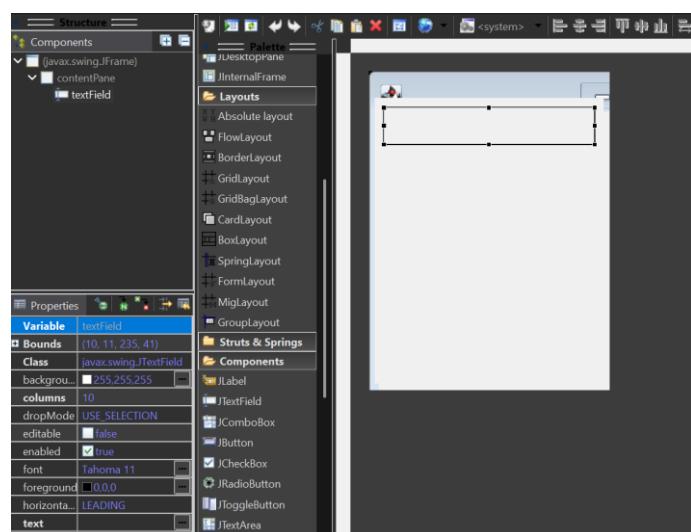
2.1.1 Langkah-Langkah Pembuatan Desain GUI Kalkulator

1. Buat package baru dengan nama ‘Pekan9_2511532015’ dan di dalam package buat kelas Java baru dengan tipe JFrame WindowBuilder dan diberi nama ‘Kalkulator_2511532015’ lalu finish.



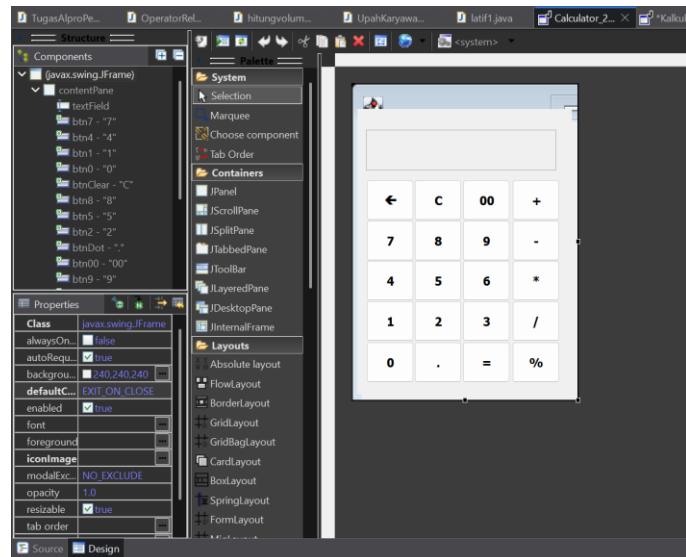
Gambar 2.1: Pembuatan kelas Java dengan tipe JFrame

2. Pembuatan tampilan visual nya dilakukan menggunakan pendekatan drag-and-dorp pada komponen-komponen Swing. Pertama, atur ukuran jendela utama supaya pas untuk komponen yang dibutuhkan untuk membuat kalkulator.
3. Komponen pertama yang dibutuhkan adalah JTextField sebagai layar kalkulator, ditempatkan di bagian atas sesuaikan ukurannya, property editable diatur ke false agar pengguna tidak bisa mengetik langsungm meainkan hanya menginput melalui tombol.



Gambar 2.2: Penambahan JTextField

4. Dibawah JTextField, disusun 20 tombol fungsional menggunakan JButton, memuat tombol angka (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) serta tombol 00 untuk efisiensi input, tombol operator (+, -, *, /, %), serta tombol kendali C (Clear) untuk mengosongkan layar, tombol sama dengan (=) untuk menampilkan hasil, titik desimal (.), Backspace (dengan ikon panah kiri menggunakan font Wingdings; \uF0E7). Tombol-tombol tersebut disusun hingga membentuk kalkulator, setiap font diberi ukuran 14 dan dibuat bold agar mudah dibaca. Untuk tombol Backspace digunakan font khusus Wingdings agar menampilkan ikon panah yang intuitif.



Gambar 2.3: Tampilan desain setelah ditambahkan komponen sehingga menyerupai kalkulator

2.1.2 Kode Program Desain GUI Kalkulator

- Setiap tombol angka (0-9), tombol titik (.), dan tombol 00 diberi ActionListener agar bisa menambahkan angkanya ke layar kalkulator saat diklik. Dengan cara menambahkan kode program berikut

```
 JButton btn1 = new JButton("1");
btn1.setFont(new Font("Tahoma", Font.BOLD, 14));
btn1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String number=textField.getText()+btn1.getText();
        textField.setText(number);
    }
});
btn1.setBounds(10, 227, 56, 48);
contentPane.add(btn1);
```

Kode Program 2.1: Kode program untuk menambahkan aksi

2. Setiap tombol operator aritmatika (+, -, *, /, dan %) tambahkan kode program seperti berikut, untuk beda operator tinggal ganti saja bagian operationnya.

```
 JButton btnPlus = new JButton("+");
btnPlus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first= Double.parseDouble(textField.getText());
        textField.setText("");
        operation="+";
    }
});
btnPlus.setFont(new Font("Tahoma", Font.BOLD, 14));
btnPlus.setBounds(183, 84, 56, 48);
contentPane.add(btnPlus);
```

Kode Program 2.2: Kode program untuk operator aritmatika

Kode program ini berfungsi untuk menangkap input pengguna saat tombol operator diklik, sehingga program menyimpan angka pertama dari layar, dan mengosongkan layar untuk angka kedua, dan mencatat bahwa operasi yang akan dilakukan adalah misal: penjumlahan, sehingga nanti saat tombol '=' ditekan, program sudah siap menghitung dan siap untuk ditampilkan jawabannya.

3. Selanjutnya tambahkan kode berikut yang berfungsi untuk mengeksekusi perhitungan aritmatika saat tombol '=' ditekan

```

JButton btnSamadengan = new JButton("=");
btnSamadengan.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);
        }
    }
});
btnSamadengan.setFont(new Font("Tahoma", Font.BOLD, 14));
btnSamadengan.setBounds(124, 276, 56, 48);
contentPane.add(btnSamadengan);

```

Kode Program 2.3: Kode program untuk mengeksekusi perhitungan matematika

Ketika pengguna menekan tombol ‘=’, program akan membaca angka kedua dari layar, lalu melakukan operasi aritmatika sesuai dengan operator yang sebelumnya dipilih. Hasil perhitungan kemudian diformat menjadi dua angka di belakang koma dan ditampilkan kembali ke layar. Dengan begitu tombol ‘=’ tidak hanya menutup input, tetapi juga memberikan hasil akhir dari operasi yang diminta.

4. Tambahkan kode program berikut yang berfungsi untuk menghapus seluruh isi layar kalkulator saat tombol ‘C’ diklik

```

btn0.setBounds(10, 276, 56, 48);
contentPane.add(btn0);

JButton btnClear = new JButton("C");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
    }
});
btnClear.setFont(new Font("Tahoma", Font.BOLD, 14));
btnClear.setBounds(67, 84, 56, 48);
contentPane.add(btnClear);

```

Kode Program 2.4: Kode program untuk menghapus seluruh isi layar kalkulator

5. Terakhir, tambahkan kode program berikut yang berfungsi untuk menghapus satu karakter terakhir dari layar kalkulator saat tombol Backspace diklik

```

 JButton btnBackSpace = new JButton("\u26D7");
 btnBackSpace.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);
        }
    }
});
btnBackSpace.setFont(new Font("Wingdings", Font.BOLD, 14));
btnBackSpace.setBounds(10, 84, 56, 48);
contentPane.add(btnBackSpace);

```

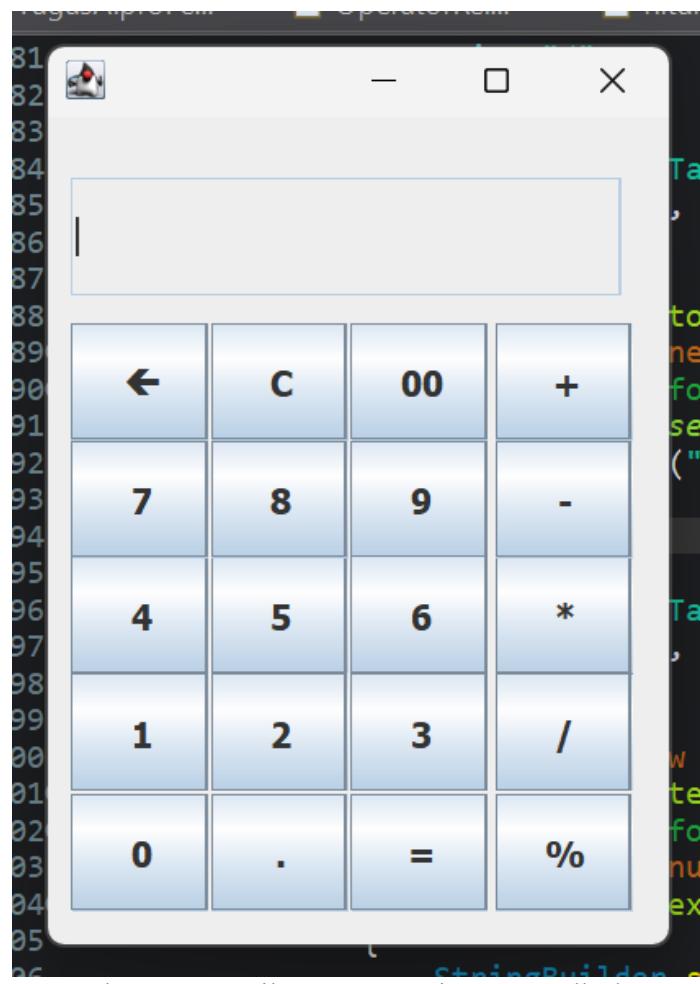
Kode Program 2.5: Kode program Backspace

Ketika pengguna menekan tombol Backspace, program pertama-tama memeriksa apakah ada teks di layar (Panjang teks > 0). Jika iya, ia membuat Salinan teks tersebut dalam bentuk StringBuilder, lalu menghapus satu karakter terakhir dengan metode deleteCharAt(). Hasilnya teks tanpa karakter terakhir kemudian ditampilkan kembali ke layar. Dengan car aini pengguna bisa memperbaiki kesalahan ketik secara cepat dan mudah, tanpa harus menghapus seluruh input. Tombol ini sangat berguna dalam pengalaman pengguna karena meniru backspace pada kalkulator fisik atau keyboard.

2.1.3 Penjelasan Output Kalkulator

Setelah kalkulator dijalankan, tampilan awal menampilkan layar kosong di bagian atas dan deret tombol angka operator di bawahnya. Pengguna dapat melakukan perhitungan dengan menekan tombol angka dan memilih operator yang diinginkan lalu menekan sama dengan (=) untuk menampilkan hasilnya, hasil perhitungan akan berbentuk dua angka dibelakang koma.

Jika pengguna melakukan kesalahan ketik, pengguna bisa menekan tombol backspace untuk menghapus satu digit terakhir, sedangkan jika ingin memulai ulang dari awal, pengguna bisa menekan tombol Clear (C), dan layar akan langsung kosong.



Gambar 2.4: Hasil Ouput Desain GUI Kalkulator

BAB III

KESIMPULAN

3.1 Kesimpulan

Praktikum ini berhasil menghasilkan sebuah aplikasi kalkulator desktop berbasis GUI menggunakan bahasa pemrograman Java dan IDE Eclipse dengan bantuan plugin WindowBuilder. Aplikasi ini tidak hanya mampu melakukan operasi aritmatika dasar, tetapi juga dilengkapi tombol interaktif seperti tombol angka, titik decimal, Clear, dan backspace, sehingga menyerupai kalkulator digital yang digunakan sehari-hari.

Melalui praktikum ini mahasiswa mampu memahami cara menggabungkan desain antarmuka dengan logika pemrograman melalui event handling, serta mengelola input pengguna secara dinamis dan resposif. Dengan demikian, praktikum ini tidak hanya melatih keterampilan teknis dalam pembuatan aplikasi desktop, tetapi juga memperkuat pemahaman bahwa program yang baik harus menggabungkan fungsi, estetika, dan kemudahan penggunaan dalam satu kesatuan yang utuh.