

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN

“Aplikasi Kalkulator”

disusun Oleh:

Muhammad Yasin Habiburrahman

2511532016

Dosen Pengampu:

Dr. Wahyudi. S.T.M.T

Asisten Pratikum:

Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan tugas kelompok ini dengan baik. Shalawat serta salam semoga senantiasa tercurah kepada junjungan kita Nabi Muhammad SAW, keluarga, sahabat, serta para pengikutnya hingga akhir zaman.

Tugas kelompok ini kami susun dengan tujuan untuk memenuhi salah satu tugas mata kuliah Algoritma dan Pemrograman, dengan tema “*Aplikasi Kalkulator*”, yang mencakup bagaimana cara membuat aplikasi kalkulator menggunakan IDE Eclipse.

Kami menyadari bahwa penyusunan tugas ini masih jauh dari sempurna, baik dari segi materi maupun penyajiannya. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun dari semua pihak demi kesempurnaan tugas kami di masa mendatang.

Akhir kata, kami mengucapkan terima kasih kepada dosen pembimbing yang telah memberikan arahan, serta kepada seluruh anggota kelompok yang telah bekerja sama dengan baik sehingga tugas ini dapat terselesaikan tepat waktu. Semoga tugas ini dapat bermanfaat bagi kami khususnya, dan bagi pembaca pada umumnya.

Padang, 1 Desember

Muhammad Yasin Habiburrahman

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat Praktikum.....	1
BAB II PEMBAHASAN.....	3
2.1 Langkah-Langkah Awal.....	3
2.2 Pembuatan dan Pengkodean Struktur	4
2.3 Logika Tombol Kalkulator.....	7
2.4 Logika Aritmatika Aplikasi Kalkulator	8
BAB III KESIMPULAN.....	10
DAFTAR PUSTAKA.....	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era modern ini, penggunaan alat hitung digital sudah menjadi bagian dari kehidupan sehari-hari, baik untuk keperluan sekolah, kuliah, maupun pekerjaan. Sebagai calon programmer, membuat sendiri sebuah kalkulator bukan hanya soal menghitung — melainkan juga latihan untuk menerapkan logika, pemrosesan input, serta desain antarmuka pengguna (*GUI*). Dengan membuat kalkulator berbasis GUI menggunakan *Java Swing*, mahasiswa dapat memahami bagaimana elemen visual, interaksi pengguna, dan logika pemrograman bersatu untuk menghasilkan program yang berfungsi nyata. Tutorial seperti dalam video “GUI Calculator in JAVA using eclipse ide | Tech Projects” memberi panduan konkret untuk membangun aplikasi kalkulator sederhana secara praktis. Praktikum ini menjadi jembatan antara konsep teoritis di kelas (variabel, input/output, pengondisian, *event handling*) dan aplikasi praktis yang dapat langsung digunakan — sekaligus memberi pengalaman membangun software desktop sederhana.

1.2 Tujuan

- Membuat program kalkulator berbasis GUI menggunakan bahasa *Java* dan pustaka *Swing*.
- Menerapkan logika aritmatika dasar (penjumlahan, pengurangan, perkalian, pembagian, modulo) dan menangani kasus khusus (misalnya pembagian/modulo dengan nol).
- Melatih kemampuan penanganan input pengguna, konversi data, dan validasi input agar program aman dan tidak error.
- Mengintegrasikan desain antarmuka, interaksi pengguna, dan logika backend dalam satu aplikasi utuh.
- Meningkatkan pemahaman terhadap konsep *event-driven programming* dan *error handling* pada aplikasi GUI.

1.3 Manfaat Praktikum

- Memahami cara membuat aplikasi GUI sederhana yang interaktif dan user-friendly.
- Mempelajari struktur program yang lebih kompleks: penggabungan antara logika pemrograman dan antarmuka grafis.
- Terbiasa menangani input dari pengguna, termasuk validasi dan penanganan kesalahan (misalnya input bukan angka, pembagian dengan nol).

- Memperkuat konsep dasar pemrograman: variabel, operator, percabangan, konversi data, serta manajemen alur program.
- Mendapatkan pengalaman nyata membuat program desktop sehingga membangun kepercayaan diri dan mempersiapkan diri untuk proyek yang lebih rumit di masa depan.

BAB II PEMBAHASAN

2.1 Langkah-Langkah Awal

Pada Pertemuan kali ini, kami diminta untuk membuat aplikasi kalkulator dengan mengikuti langkah-langkah dari video tutorial YouTube.

Pertama-tama, kita diminta untuk membuat proyek baru melalui File → New → Jawa Project di IDE Eclipse. Kemudian di bagian *package explorer*, klik kanan folder proyek yang baru saja dibuat, klik New → Other → WindowBuilder → SwingDesigner → Application Window. Kemudian, file tersebut diberi nama, misalnya “Calculator”.

Maka, secara otomatis, tampilan pada bagian *Source* akan menjadi seperti berikut:

```

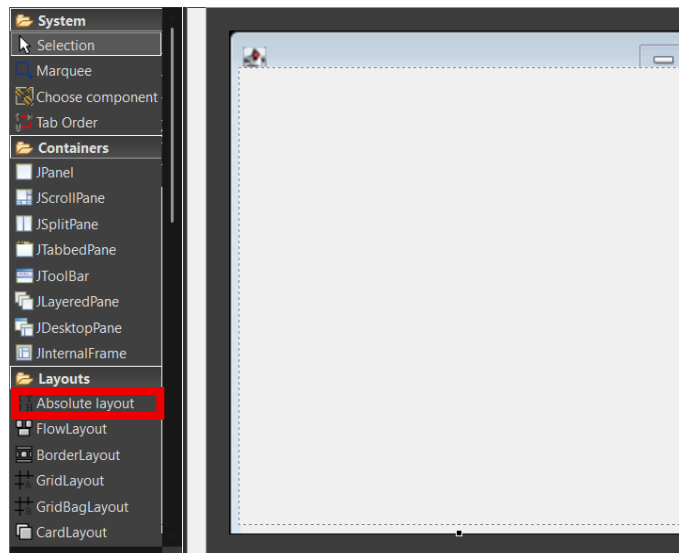
1  package pekan9_2511532016;
2
3  import java.awt.EventQueue;
4
5  import javax.swing.JFrame;
6
7  public class Calculator {
8
9      private JFrame frame;
10
11     /**
12      * Launch the application.
13      */
14     public static void main(String[] args) {
15         EventQueue.invokeLater(new Runnable() {
16             public void run() {
17                 try {
18                     Calculator window = new Calculator();
19                     window.frame.setVisible(true);
20                 } catch (Exception e) {
21                     e.printStackTrace();
22                 }
23             }
24         });
25     }
26
27     /**
28      * Create the application.
29      */
30     public Calculator() {
31         initialize();
32     }
33
34     /**
35      * Initialize the contents of the frame.
36      */
37     private void initialize() {
38         frame = new JFrame();
39         frame.setBounds(100, 100, 450, 300);
40         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
41     }
42
43 }

```

Gambar 2.1 – Kode Program Awal dari Aplikasi Kalkulator

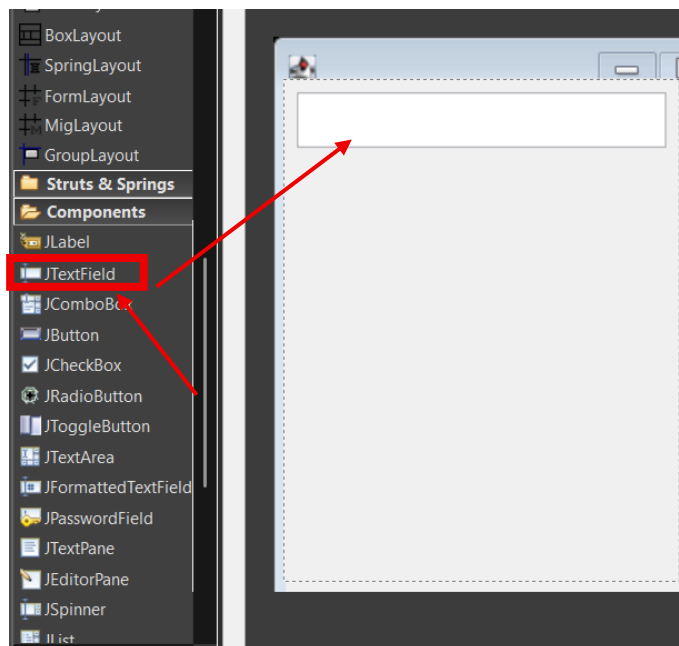
2.2 Pembuatan dan Pengkodean Struktur

Tahap selanjutnya, pergi ke bagian *Design* untuk memulai. Atur ukuran panel kalkulator sesuai yang diinginkan. Di bagian *Layouts*, pilih “Absolute Layout”. Absolute layout digunakan ketika ingin menempatkan komponen GUI secara manual dengan koordinat x, y, lebar, tinggi yang kamu tentukan sendiri menggunakan `setBounds()`. Tarik absolute layout ke panel.



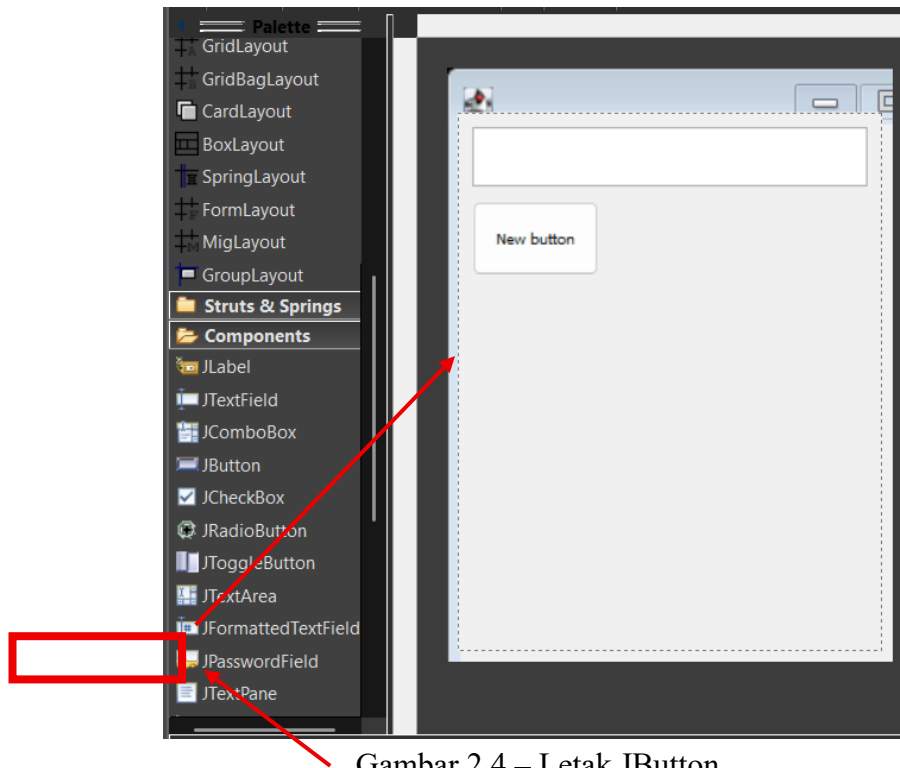
Gambar 2.2 – Letak Absolute Layout

Selanjutnya, tambahkan `JTextField` di bagian *Components*. Tarik ke panel, kemudian atur posisi dan ukuran sesuai dengan keinginan.



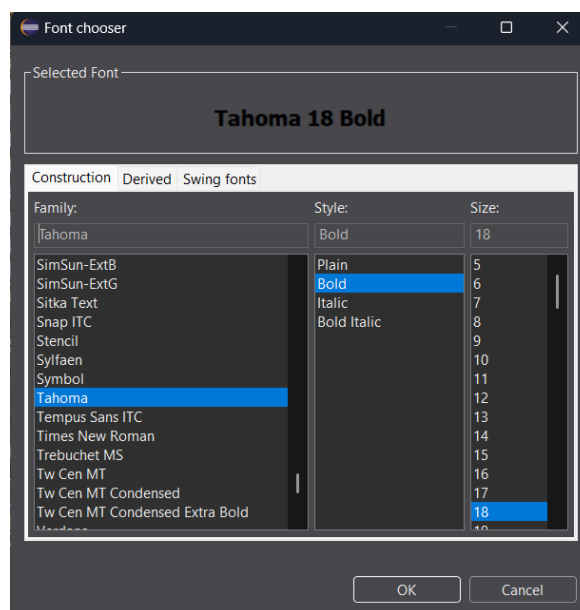
Gambar 2.3 – Letak JTextField

Kemudian, kita perlu menambahkan tombol-tombol angka, operasi dan beberapa karakter lainnya, seperti 00 dan koma (.). Untuk itu, kita perlu menggunakan JButton.



Gambar 2.4 – Letak JButton

Selanjutnya, tekan tombol pada panel, dan akan muncul bagian “Properties”. Bagian ini mengatur tampilan serta tombol dan teks pada tombol. Pada kasus ini, semua tulisan di tombol akan memiliki font “Tahoma”, berukuran 18, dan memiliki style **bold**.

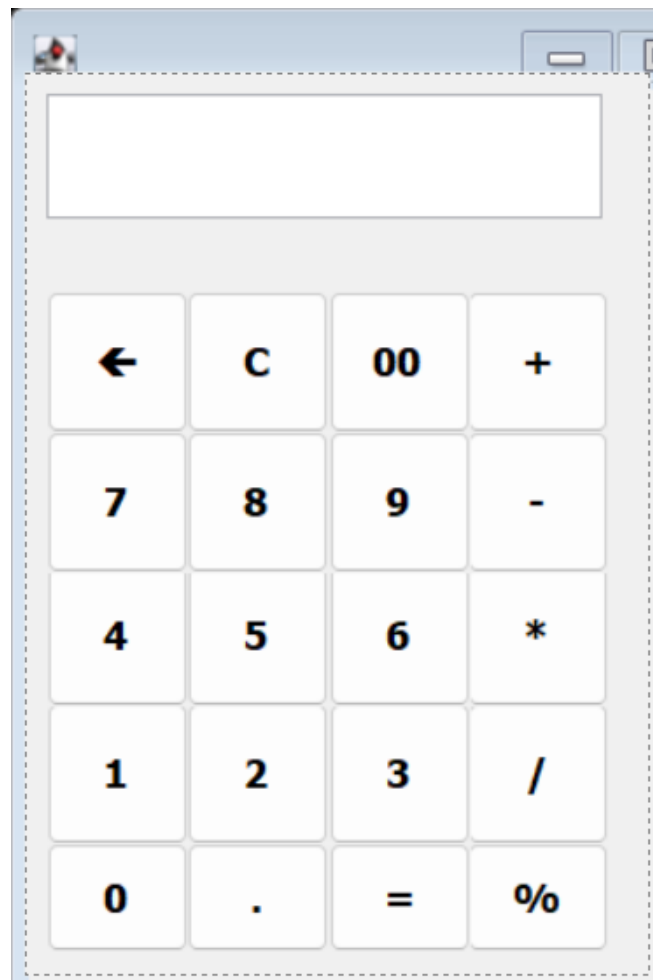


Gambar 2.5 – Tampilan Layar Font

Properties	
Variable	btnNewButton
Constructor	(Constructor properties)
Bounds	(10, 60, 85, 50)
Class	javax.swing.JButton
background	240,240,240
enabled	<input checked="" type="checkbox"/> true
font	Tahoma 10
foreground	0,0,0
horizontalAlignment	CENTER
icon	
mnemonic(char)	
selectedIcon	
text	New button
toolTipText	
verticalAlignment	CENTER

Gambar 2.6 – Tampilan Menu Properties

Kita membutuhkan 20 tombol pada kalkulator ini. Tambahkan tombol-tombol lagi mengikuti langkah sebelumnya. Kemudian, berikan tulisan seperti gambar berikut.



Gambar 2.7 – Tampilan Akhir Aplikasi Kalkulator

Pada setiap tombol, berikan nama berdasarkan fungsi tombol di bagian “Variable” pada menu “Properties”. Misalkan, btn1, btn2, btn3, dst. Terkhusus untuk tombol hapus, digunakan nama variabel “\uF0E7” dan font “Wingdings” untuk mendapatkan tanda “←”.

2.3 Logika Tombol Kalkulator

Kalkulator tidak dapat berjalan tanpa adanya logika. Kita perlu menambahkan kode program agar kalkulator dapat berjalan sesuai dengan keinginan. Tekan dua kali salah satu tombol, misalkan tombol 1, maka akan langsung otomatis dibawa ke bagian “Source” dan potongan program untuk tombol tersebut.

```

 JButton btn1 = new JButton("1");
 btn1.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
     }
 }

```

2.8 – Potongan Program Tombol

Pada kelas actionPerformed, tambahkan kode program agar menjadi seperti berikut.

```

 JButton btn1 = new JButton("1");
 btn1.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
         String number = textField.getText()+btn1.getText();
         textField.setText(number);
     }
 }

```

2.9 – Potongan Program Tombol Lanjutan

Kode program itu akan menampilkan angka yang bersesuaian ke text field utama. Untuk tombol lainnya, kode programnya harus sesuai dengan variabelnya. Pada contoh potongan di atas variabel yang dipakai adalah “btn1”. Kode program lain, kecuali untuk tombol operator, harus menyesuaikan dengan variabel yang bersesuaian agar tidak terjadi error. Setelah ditambahkan, kalkulator akan mendapatkan fungsi untuk menambahkan angka pada text field utama.

Untuk menghapus angka saat meng-input, digunakan program berikut.

```

 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 2.10 – Kode Program Hapus Angka

Pertama-tama, Variabel backSpace disiapkan untuk menyimpan teks setelah karakter terakhir dihapus. Lalu, program memastikan bahwa panjang teks lebih dari 0. Jika kosong, backspace tidak dilakukan agar tidak terjadi error. Kemudian, StringBuilder digunakan karena bisa memodifikasi teks secara langsung.

`deleteCharAt()` menghapus karakter pada indeks tertentu. Indeks `length - 1` berarti karakter paling akhir. Kemudian, Hasil dari `StringBuilder` diubah ke dalam bentuk string biasa agar bisa ditampilkan ke text field. Terakhir, teks yang sudah dihapus karakter terakhirnya ditampilkan kembali ke layar.

Untuk menghapus semua angka, atau mengulang lagi dari awal, dibutuhkan fungsi `clear` yang ditandai dengan tombol “C”. Sederhananya, kita hanya perlu menambahkan “`textField.setText(null)`” pada `actionPerformed`.

```

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

```

Gambar 2.11 – Kode Program Tombol Clear

2.4 Logika Aritmatika Aplikasi Kalkulator

Sebuah kalkulator memerlukan logika aritmatika agar berjalan sesuai dengan fungsinya. Untuk itu, beberapa kode program perlu ditambahkan.

Kita perlu membuat beberapa variabel global terlebih dahulu, yaitu:

`first; double` ← Untuk menyimpan angka pertama

`second; double` ← Untuk menyimpan angka kedua

`result; double` ← Untuk menyimpan hasil operasi

`operation; String` ← Untuk menyimpan operator

`answer; String` ← Untuk menyimpan hasil operasi dalam bentuk String

Untuk setiap method `actionPerformed` tombol operator, misalkan tanda tambah (+), ditambahkan kode program berikut.

```

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

```

Gambar 2.12 – Contoh Kode Program Operator Aritmatika

Variabel `first` akan menyimpan isi text field yang berisi angka yang telah dimasukkan. Kemudian angka tersebut diubah dari String menjadi Double, dan disimpan ke variabel `first` sebagai angka pertama untuk operasi penjumlahan.

Selanjutnya, text field utama dikosongkan, siap untuk menerima input kedua dari pengguna. Lakukan hal yang sama untuk tombol operator aritmatika lainnya dengan menyesuaikan simbol aritmatikanya!

Tanda sama dengan digunakan dalam aritmatika sebagai simbol untuk menyatakan hasil dari suatu perhitungan matematika. Oleh karena itu, dibutuhkan kode program yang sesuai dengan kaidah matematika. Kode program untuk tombol “=” adalah sebagai berikut.

```

JButton btnEquals = new JButton("=");
btnEquals.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);
        }
    }
}

```

Gambar 2.13 – Kode Program Tombol “=”

Pertama-tama, angka pada text field diambil, diubah dari string menjadi double, kemudian nilainya disimpan dalam variabel “second”.

Kemudian, dilakukan percabangan dengan mengecek operator yang digunakan oleh pengguna. Hasil operasi disimpan dalam variabel “result”, yang kemudian akan diubah dalam bentuk String dengan nilai dua angka di belakang koma dan disimpan dalam variabel “answer”. Terakhir, nilai akan ditampilkan pada text field dari variabel answer”.

BAB III

KESIMPULAN

Pada praktikum minggu kesembilan ini, mahasiswa berhasil memahami dan menerapkan pembuatan aplikasi kalkulator berbasis *Graphical User Interface* menggunakan bahasa Java dan pustaka *Swing*. Melalui pembuatan tombol angka, operator, pengelolaan input pengguna, serta logika perhitungan, mahasiswa dapat melihat secara langsung bagaimana sebuah aplikasi yang sederhana namun fungsional dapat dibangun dari konsep dasar pemrograman yang telah dipelajari sebelumnya. Selain itu, penggunaan metode seperti konversi string ke angka, penyimpanan nilai sementara, pemilihan operasi, dan eksekusi perhitungan melalui tombol “=” memperkuat pemahaman terhadap *event-driven programming*. Proses ini juga melibatkan penanganan kesalahan, misalnya ketika pengguna melakukan input tidak valid atau mencoba melakukan operasi pembagian dengan nol, sehingga mahasiswa semakin memahami pentingnya validasi data dalam pengembangan aplikasi interaktif. Secara keseluruhan, praktikum ini memberikan pengalaman nyata dalam mengembangkan aplikasi GUI yang terstruktur, meningkatkan kemampuan logika pemrograman, serta menanamkan dasar-dasar pembuatan perangkat lunak yang dapat dikembangkan ke tahap yang lebih kompleks di masa mendatang.

DAFTAR PUSTAKA

[1] Tech Projects. (2020, April 3). *GUI Calculator in JAVA using eclipse ide | Tech Projects* [Video]. YouTube. <https://www.youtube.com/watch?v=mZVD5OssHQM>