

LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN  
PEMROGRAMAN GUI 2

Disusun Oleh:

Nama: Rifki Maulana

NIM: 2511533007

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

Asisten Praktikum: Jovantri Immanuel Gulo



DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
TAHUN 2025

## **KATA PENGANTAR**

Puji syukur kehadiran Tuhan Yang Maha Esa karena dengan rahmat-Nya laporan praktikum ini dapat diselesaikan. Laporan ini disusun sebagai bagian dari pelaksanaan mata kuliah Praktikum **Algoritma dan Pemrograman** untuk memenuhi tugas dan memperdalam pemahaman praktis terhadap materi yang dibahas.

Terima kasih disampaikan kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah memberikan bimbingan, bantuan, dan kerja sama selama kegiatan praktikum berlangsung. Penulis menyadari laporan ini masih jauh dari sempurna, oleh karena itu kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang.

Semoga laporan ini bermanfaat bagi penulis dan pembaca.

**Padang, 27 November 2025**

**Rifki Maulana**

## DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN.....	3
2.1 Uraian Kode Program.....	3
2.2 Langkah Kerja.....	8
2.3 Analisis hasil.....	8
BAB 3 KESIMPULAN.....	10
DAFTAR PUSTAKA.....	11

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pemrograman merupakan keterampilan dasar yang penting dalam bidang informatika. GUI (Graphical User Interface) merupakan bagian penting dalam pengembangan aplikasi modern karena menyediakan cara yang lebih simple dan mudah bagi pengguna untuk berinteraksi dengan program dibandingkan dengan antarmuka berbasis teks. Pada Java, salah satu framework GUI yang sering digunakan adalah Swing, yang menyediakan komponen-komponen seperti JFrame, JPanel, JLabel, JTextField, dan JButton untuk membangun antarmuka pengguna yang interaktif. Menurut Deitel & Deitel (2017), Java Swing menawarkan fleksibilitas tinggi dalam merancang aplikasi desktop dengan komponen yang dapat disesuaikan sesuai kebutuhan. Selain itu, alat bantu visual seperti WindowBuilder mempermudah perancangan layout dan pengaturan komponen secara visual sehingga proses pembuatan GUI menjadi lebih cepat, efisien, dan mudah dipahami terutama bagi pemula.

Praktikum ini menggunakan contoh program kalkulator sederhana yang dibuat dengan WindowBuilder untuk mendemonstrasikan implementasi GUI dalam aplikasi nyata. Program kalkulator tersebut menampilkan tombol-tombol angka (0-9), operator aritmatika (+, -, \*, /, %), fungsi khusus seperti clear dan backspace, serta area display untuk menampilkan input dan hasil perhitungan. Selain implementasi operasi matematis dasar, program juga memperlihatkan aspek penting lain dalam pengembangan GUI seperti event handling menggunakan ActionListener dan pengelolaan state aplikasi melalui variabel instance. Horstmann (2019) menekankan bahwa pemahaman tentang event-driven programming sangat penting dalam pengembangan aplikasi GUI karena memungkinkan aplikasi merespons aksi pengguna secara dinamis dan tepat.

## **1.2 Tujuan Praktikum**

Praktikum ini memiliki tujuan sebagai berikut:

- a) Memahami konsep dasar pemrograman GUI dengan Java.
- b) Menggunakan komponen GUI (misalnya JTextField, JComboBox, JButton, JLabel) dan mengatur layout sederhana.
- c) Menggunakan logika pemrograman untuk membuat program kalkulator sederhana

## **1.3 Manfaat Praktikum**

Praktikum ini memberikan manfaat penting bagi mahasiswa dalam memperoleh pengalaman secara langsung merancang aplikasi berbasis Graphical User Interface (GUI) menggunakan Java Swing dan WindowBuilder. Praktikum ini memberikan pemahaman mengenai cara kerja komponen-komponen GUI, penerapan event-driven programming, serta implementasi logika aplikasi yang terintegrasi dengan antarmuka pengguna sehingga aplikasi dapat merespons interaksi pengguna secara efektif. Mahasiswa juga dilatih untuk mengorganisir kode program dengan baik, mengelola state aplikasi menggunakan variabel instance, dan memahami alur kerja aplikasi dari input hingga output. Kemampuan ini sangat bermanfaat sebagai dasar dalam pengembangan aplikasi desktop yang lebih kompleks pada dunia kerja, serta meningkatkan keterampilan mahasiswa dalam menghasilkan perangkat lunak yang interaktif, fungsional, dan ramah pengguna sesuai dengan standar industri.

## BAB 2

### PEMBAHASAN

#### 2.1 Uraian Kode Program

```
package Pekan9_2511533007;

import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.Font;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;

public class kalkulator_2511533007 {

    private JFrame frame;
    private JTextField textField;

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

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    kalkulator_2511533007 window = new
kalkulator_2511533007();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the application.
     */
    public kalkulator_2511533007() {
        initialize();
    }

    /**
     * Initialize the contents of the frame.
     */
    private void initialize() {
        frame = new JFrame();
        frame.setBounds(100, 100, 321, 426);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.getContentPane().setLayout(null);

        textField = new JTextField();
        textField.setFont(new Font("Times New Roman", Font.BOLD, 18));
        textField.setBounds(10, 11, 272, 48);
        frame.getContentPane().add(textField);
    }
}
```

```

textField.setColumns(10);

JButton btnBackspace = new JButton("\uF0E7");
btnBackspace.setFont(new Font("Wingdings", Font.BOLD, 14));
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.setBounds(10, 83, 69, 55);
frame.getContentPane().add(btnBackspace);

JButton btn7_2511533007 = new JButton("7");
btn7_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String
number=textField.getText()+btn7_2511533007.getText();
        textField.setText(number);
    }
});
btn7_2511533007.setBounds(10, 135, 69, 55);
frame.getContentPane().add(btn7_2511533007);

JButton btn4_2511533007 = new JButton("4");
btn4_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String
number=textField.getText()+btn4_2511533007.getText();
        textField.setText(number);
    }
});
btn4_2511533007.setBounds(10, 187, 69, 55);
frame.getContentPane().add(btn4_2511533007);

JButton btn1_2511533007 = new JButton("1");
btn1_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String
number=textField.getText()+btn1_2511533007.getText();
        textField.setText(number);
    }
});
btn1_2511533007.setBounds(10, 239, 69, 55);
frame.getContentPane().add(btn1_2511533007);

JButton btn0_2511533007 = new JButton("0");
btn0_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String
number=textField.getText()+btn0_2511533007.getText();
        textField.setText(number);
    }
});
btn0_2511533007.setBounds(10, 291, 69, 55);
frame.getContentPane().add(btn0_2511533007);

JButton btnC = new JButton("C");
btnC.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
    }
});
btnC.setBounds(78, 83, 69, 55);
frame.getContentPane().add(btnC);

```

```

        JButton btn8_2511533007 = new JButton("8");
        btn8_2511533007.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String
number=textField.getText()+btn8_2511533007.getText();
                textField.setText(number);
            }
        });
        btn8_2511533007.setBounds(78, 135, 69, 55);
        frame.getContentPane().add(btn8_2511533007);

        JButton btn5_2511533007 = new JButton("5");
        btn5_2511533007.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String
number=textField.getText()+btn5_2511533007.getText();
                textField.setText(number);
            }
        });
        btn5_2511533007.setBounds(78, 187, 69, 55);
        frame.getContentPane().add(btn5_2511533007);

        JButton btn2_2511533007 = new JButton("2");
        btn2_2511533007.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String
number=textField.getText()+btn2_2511533007.getText();
                textField.setText(number);
            }
        });
        btn2_2511533007.setBounds(78, 239, 69, 55);
        frame.getContentPane().add(btn2_2511533007);

        JButton btnDot_2511533007 = new JButton(".");
        btnDot_2511533007.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String
number=textField.getText()+btnDot_2511533007.getText();
                textField.setText(number);
            }
        });
        btnDot_2511533007.setBounds(78, 291, 69, 55);
        frame.getContentPane().add(btnDot_2511533007);

        JButton btn00 = new JButton("00");
        btn00.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String number=textField.getText()+btn00.getText();
                textField.setText(number);
            }
        });
        btn00.setBounds(147, 83, 69, 55);
        frame.getContentPane().add(btn00);

        JButton btn9_2511533007 = new JButton("9");
        btn9_2511533007.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String
number=textField.getText()+btn9_2511533007.getText();
                textField.setText(number);
            }
        });
        btn9_2511533007.setBounds(147, 135, 69, 55);
        frame.getContentPane().add(btn9_2511533007);

        JButton btn6_2511533007 = new JButton("6");
        btn6_2511533007.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String
number=textField.getText()+btn6_2511533007.getText();

```



```

        textField.setText(number);
    }
});
btn6_2511533007.setBounds(147, 187, 69, 55);
frame.getContentPane().add(btn6_2511533007);

JButton btn3_2511533007 = new JButton("3");
btn3_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String
number=textField.getText()+btn3_2511533007.getText();
        textField.setText(number);
    }
});
btn3_2511533007.setBounds(147, 239, 69, 55);
frame.getContentPane().add(btn3_2511533007);

JButton btnEqual_2511533007 = new JButton("=");
btnEqual_2511533007.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_2511533007.setBounds(147, 291, 69, 55);
frame.getContentPane().add(btnEqual_2511533007);

JButton btnPlus_2511533007 = new JButton("+");
btnPlus_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="+";
    }
});
btnPlus_2511533007.setBounds(213, 83, 69, 55);
frame.getContentPane().add(btnPlus_2511533007);

JButton btnMinus_2511533007 = new JButton("-");
btnMinus_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="-";
    }
});
btnMinus_2511533007.setBounds(213, 135, 69, 55);

```

```

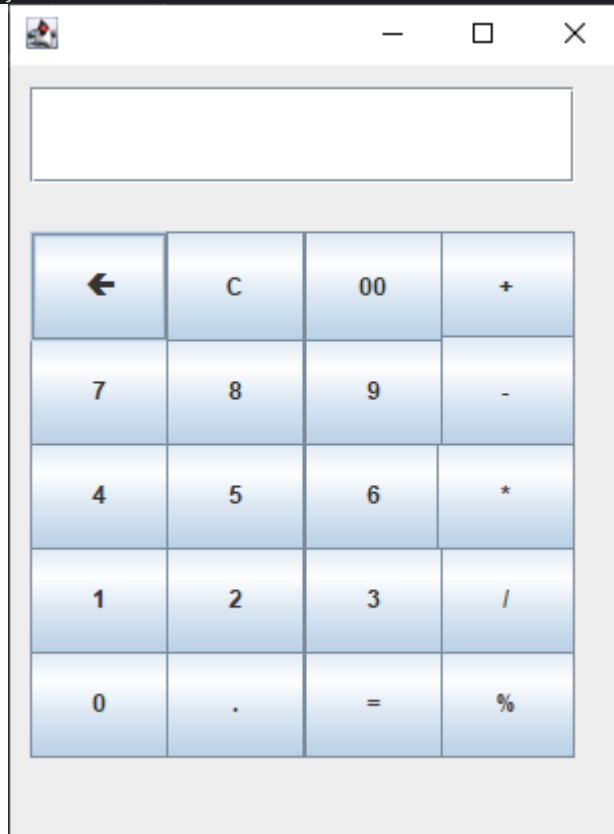
frame.getContentPane().add(btnMinus_2511533007);

JButton btnMul_2511533007 = new JButton("*");
btnMul_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="*";
    }
});
btnMul_2511533007.setBounds(213, 187, 69, 55);
frame.getContentPane().add(btnMul_2511533007);

JButton btnDiv_2511533007 = new JButton("/");
btnDiv_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="/";
    }
});
btnDiv_2511533007.setBounds(213, 239, 69, 55);
frame.getContentPane().add(btnDiv_2511533007);

JButton btnPercent_2511533007 = new JButton("%");
btnPercent_2511533007.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="%";
    }
});
btnPercent_2511533007.setBounds(213, 291, 69, 55);
frame.getContentPane().add(btnPercent_2511533007);
}
}

```



Kode Program 2.1 kalkulator\_25115330007

## **2.2 Langkah Kerja**

Langkah-langkah menyusun program:

Program OperatorAritmatikaGUI\_2511533007

- a) Mulai
- b) Buka package, kemudian cari other
- c) Setelah membuka other klik JFrame
- d) Kemudian cari design
- e) Atur layout menjadi absolut layout
- f) Cari JButton dan buatlah variabel beserta text pada button sesuai kalkulator yang bisa anda lihat
- g) Buat JTextField untuk hasil aritmatika
- h) Buat operasinya menggunakan if else statement
- i) selesai

## **2.3 Analisis Hasil**

Analisis Kode Program Kalkulator\_2511533007

Program kalkulator ini dibuat menggunakan Java Swing pada windowbuilder. Menurut Deitel & Deitel (2017), Java Swing menyediakan komponen-komponen GUI yang lebih fleksibel dan ringan dibandingkan dengan AWT karena menggunakan pendekatan Model-View-Controller (MVC). Program mengimplementasikan pola Event-Driven Programming di mana setiap tombol memiliki ActionListener yang merespons interaksi user. Komponen utama terdiri dari JTextField sebagai display dan 20 tombol JButton untuk angka (0-9), operator (+, -, \*, /, %), serta fungsi khusus seperti clear, backspace, dan =.

Program menggunakan variabel instance untuk menyimpan state perhitungan yaitu first dan second untuk operand, operation untuk operator, dan result untuk hasil. Menurut Sierra & Bates (2005), penggunaan variabel instance untuk menyimpan state adalah praktik yang baik dalam pemrograman berorientasi

objek. Alur kerja dimulai ketika pengguna memasukkan angka pertama, memilih operator yang menyimpan angka ke variabel `first`, memasukkan angka kedua, dan menekan tombol sama dengan untuk menampilkan hasil dengan format dua desimal menggunakan `String.format("%.2f", result)`. Program juga dilengkapi fitur `backspace` menggunakan untuk menghapus karakter terakhir dan fungsi `clear` untuk mengosongkan `textField`, sehingga meningkatkan interaktivitas dan kemudahan penggunaan bagi pengguna.

Meskipun program ini dapat berfungsi, terdapat beberapa kekurangan yang perlu diperbaiki. Pertama, program menggunakan operator `==` untuk membandingkan string pada operator, padahal Bloch (2018) menyatakan seharusnya menggunakan method `equals()` karena `==` membandingkan referensi objek bukan nilai string. Kedua, program tidak memiliki exception handling untuk menangani `NumberFormatException` ketika input tidak valid, padahal Horstmann (2019) menekankan pentingnya exception handling untuk meningkatkan robustness aplikasi. Ketiga, tidak ada validasi untuk mencegah pembagian dengan nol. Secara keseluruhan, program berhasil mengimplementasikan operasi aritmatika dasar dengan GUI yang interaktif, namun memerlukan perbaikan pada penanganan error dan validasi input untuk meningkatkan kualitas program.

### **BAB III**

### **KESIMPULAN**

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa pembuatan program kalkulator menggunakan Java Swing memberikan pemahaman mengenai cara membuat antarmuka grafis dan menghubungkannya dengan logika pemrograman melalui event handling. Program kalkulator yang dibuat mampu menjalankan operasi aritmatika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus berdasarkan input pengguna melalui tombol-tombol yang tersedia. Program menggunakan komponen GUI seperti JFrame, JTextField, dan JButton yang diatur dengan absolute positioning, serta mengimplementasikan ActionListener untuk menangani event ketika pengguna menekan tombol. Fitur tambahan seperti backspace, clear, dan format output dengan dua angka desimal juga berhasil diimplementasikan untuk meningkatkan fungsionalitas dan kemudahan penggunaan aplikasi. Meskipun demikian, terdapat beberapa aspek yang perlu diperbaiki seperti validasi input untuk mencegah kesalahan penggunaan, penanganan exception menggunakan try-catch untuk menangani NumberFormatException, penggunaan method equals() untuk perbandingan string, serta validasi pembagian dengan nol agar aplikasi lebih bagus dan dapat menangani berbagai kondisi error dengan baik. Dengan demikian, praktikum ini berhasil menunjukkan bagaimana pembuatan aplikasi kalkulator sederhana, pengolahan data input, implementasi event-driven programming, serta komponen GUI dapat diintegrasikan untuk menghasilkan aplikasi desktop yang interaktif dan fungsional, sekaligus memberikan pemahaman tentang pentingnya validasi dan error handling dalam pengembangan perangkat lunak yang bagus.

## DAFTAR PUSTAKA

- [1] P. Deitel dan H. Deitel, *Java How to Program, Early Objects*, 11th ed. Boston, MA: Pearson, 2017
- [2] C. S. Horstmann, *Core Java Volume I – Fundamentals*, 11th ed. Upper Saddle River, NJ: Prentice Hall, 2019
- [3] K. Sierra dan B. Bates, *Head First Java*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2005
- [4] J. Bloch, *Effective Java*, 3rd ed. Boston, MA: Addison-Wesley Professional, 2018