

**LAPORAN PRAKTIKUM ALGORITMA  
PEMROGRAMAN**

**“PENERAPAN GRAPHICAL USER INTERFACE (GUI)  
DALAM PEMROGRAMAN JAVA”**

**PEMBUATAN KALKULATOR**

**DISUSUN OLEH:**

**NAYSHA APRILIA RIZKI**

**2511531014**

**DOSEN PENGAMPU: Dr. WAHYUDI, S.T,M.T  
ASISTEN PRAKTIKUM: MUHAMMAD ZAKI AL  
HAFIZ**



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

**2025**

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Laporan praktikum ini dapat diselesaikan dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas dalam rangka memenuhi tugas yang diberikan oleh dosen dan juga asisten praktikum.

Padang, 30 November 2025

Naysha Aprilia Rizki



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk di bidang pendidikan, industri, dan komunikasi. Salah satu pilar utama dalam perkembangan teknologi tersebut adalah **pemrograman komputer**. Di antara banyak bahasa pemrograman yang ada, **Java** merupakan salah satu bahasa yang sangat populer dan banyak digunakan di berbagai belahan dunia.

Java adalah bahasa pemrograman tingkat tinggi yang bersifat **object-oriented**, dikembangkan oleh Sun Microsystems pada tahun 1995, dan kini dikelola oleh Oracle. Keunggulan utama Java terletak pada kemampuannya untuk berjalan di berbagai platform (platform-independent) berkat konsep **Write Once, Run Anywhere (WORA)** yang dimilikinya. Hal ini memungkinkan program yang dibuat dalam Java untuk dijalankan di berbagai sistem operasi tanpa perlu diubah ulang.

Selain itu, Java memiliki sintaks yang relatif mudah dipahami bagi pemula, dokumentasi yang lengkap, dan komunitas pengguna yang besar, sehingga sangat mendukung proses belajar dan pengembangan perangkat lunak. Java juga menjadi bahasa utama dalam pengembangan berbagai aplikasi, seperti aplikasi desktop, mobile (Android), web, hingga sistem enterprise berskala besar.

Dengan melihat pentingnya Java dalam dunia pemrograman dan aplikasinya yang luas, sangatlah relevan untuk mempelajari dasar-dasar bahasa pemrograman Java, baik dari segi teori maupun praktik. Oleh karena itu, laporan ini disusun untuk memberikan pemahaman yang lebih mendalam tentang bahasa Java, struktur dasarnya, serta penerapan konsep pemrograman berorientasi objek dalam bahasa tersebut

## **1.2 Tujuan**

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik

## **1.3 Manfaat Praktikum**

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bahan awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.
6. Memberikan pengalaman langsung dalam merancang dan mengimplementasikan komponen GUI pada aplikasi atau program Java.

## BAB II

### PEMBAHASAN

#### **2.1 Pengertian Graphical User Interface (GUI)**

Dalam bahasa pemrograman Java, Graphical User Interface (GUI) merupakan antarmuka yang memungkinkan user berinteraksi dengan aplikasi/ program melalui elemen visual seperti tombol, jendela, ikon, dan text box. Dalam pemrograman Java, GUI biasanya dibuat menggunakan Java Swing, yaitu library yang menyediakan berbagai komponen untuk menyusun tampilan aplikasi. Dengan adanya GUI, pengguna dapat melakukan input dan navigasi menggunakan komponen visual dengan cara merancang dan mengimplementasikan GUI menggunakan Java Swing serta memanfaatkan WindowBuilder, yaitu plugin visual editor untuk membuat tampilan i secara drag-and-drop di Eclipse. Pemahaman terhadap JFrame, komponen GUI, serta mekanisme event handling menjadi dasar penting dalam membangun aplikasi /program Java yang interaktif.

##### ***1. WindowBuilder***

WindowBuilder adalah sebuah GUI designer untuk Java yang tersedia sebagai plugin pada IDE Eclipse. Dengan WindowBuilder, dapat merancang tampilan GUI secara visual menggunakan metode drag-and-drop, tanpa harus menulis semua kode secara manual. WindowBuilder mendukung pengembangan antarmuka menggunakan Swing, SWT, maupun GWT.

Berikut beberapa fungsi dari WindowBuilder:

1. Sinkronisasi otomatis antara editor visual dan kode.
2. Editor visual untuk mendesain layout.
3. Mempercepat proses pembuatan antarmuka Java 6

##### ***2. Java Swing***

Java Swing adalah library GUI yang termasuk dalam Java Foundation Classes (JFC). Swing digunakan untuk membuat aplikasi desktop dengan tampilan yang dapat dikustomisasi. Library ini menyediakan banyak komponen GUI, mulai dari komponen dasar seperti tombol hingga komponen kompleks seperti tabel dan dialogs.

Keunggulan Java Swing:

1. Bersifat lightweight (tidak bergantung pada sistem operasi).
2. Memiliki banyak komponen bawaan.
3. Memungkinkan pembuatan layout yang fleksibel.

### ***3. Container J-Frame***

JFrame adalah kelas utama dalam swing yang digunakan untuk membuat jendela aplikasi. JFrame berfungsi sebagai wadah (container) utama yang menampung berbagai komponen GUI seperti tombol, label, dan text field.

### ***4. Component GUI Java Swing***

Komponen GUI adalah elemen visual yang digunakan pengguna untuk berinteraksi dengan program. Beberapa komponen dasar yang digunakan pada praktikum ini antara lain:

- 1. JLabel** Digunakan untuk menampilkan teks atau informasi statis. Fungsi: memberi petunjuk, judul, atau deskripsi input.
- 2. JTextField** Komponen input untuk menerima masukan berupa teks dari pengguna. Biasanya digunakan untuk formulir, input nilai, atau data sederhana.
- 3. JButton** Tombol yang dapat ditekan pengguna untuk melakukan aksi tertentu. Umumnya digunakan bersama ActionListener.
- 4. JPanel** Wadah tambahan yang digunakan untuk menata komponen secara rapi berdasarkan layout tertentu. Membantu modularisasi tampilan aplikasi.
- 5. JComboBox** Drop-down menu untuk memilih salah satu opsi dari daftar yang dibuat.
- 6. JCheckBox & JRadioButton** Komponen input pilihan:
  - i. **JCheckBox** (pilihan bisa lebih dari satu)
  - ii. **JRadioButton** (pilihan tunggal dalam satu grup)

### ***5. Event Handling***

Event handling adalah mekanisme untuk menangani aksi pengguna, seperti menekan tombol, mengetik teks, atau memilih opsi. Dalam Java Swing, event handling dilakukan menggunakan listener, seperti:

- i. **ActionListener**, menangani aksi tombol

**ii.Listener**, menangani input keyboard.

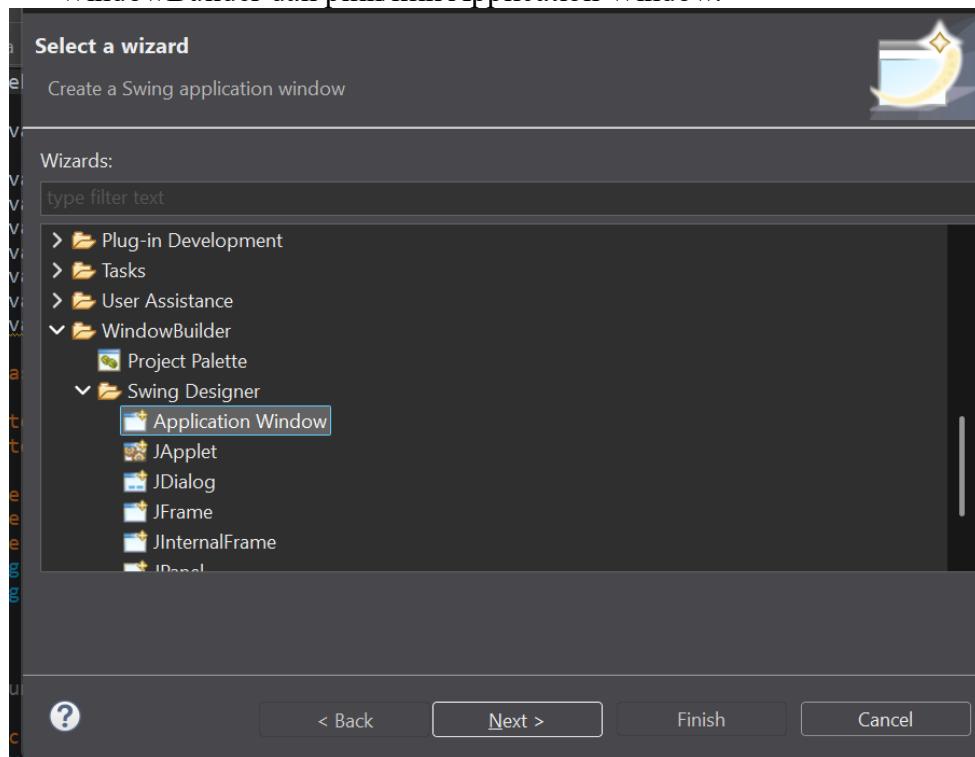
**iii.Listener**, menangani klik mouse.

Event handling memastikan aplikasi merespon interaksi pengguna dengan benar dan dinamis.

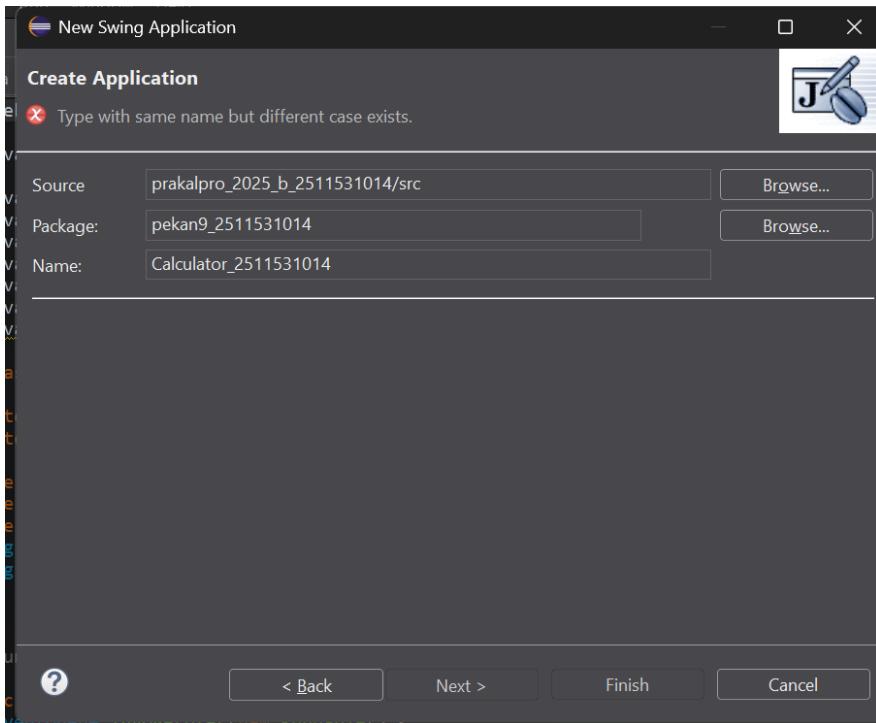
## 2,2 Praktikum Pekan 8

### 2.1.1 Membuat Package dan Class baru

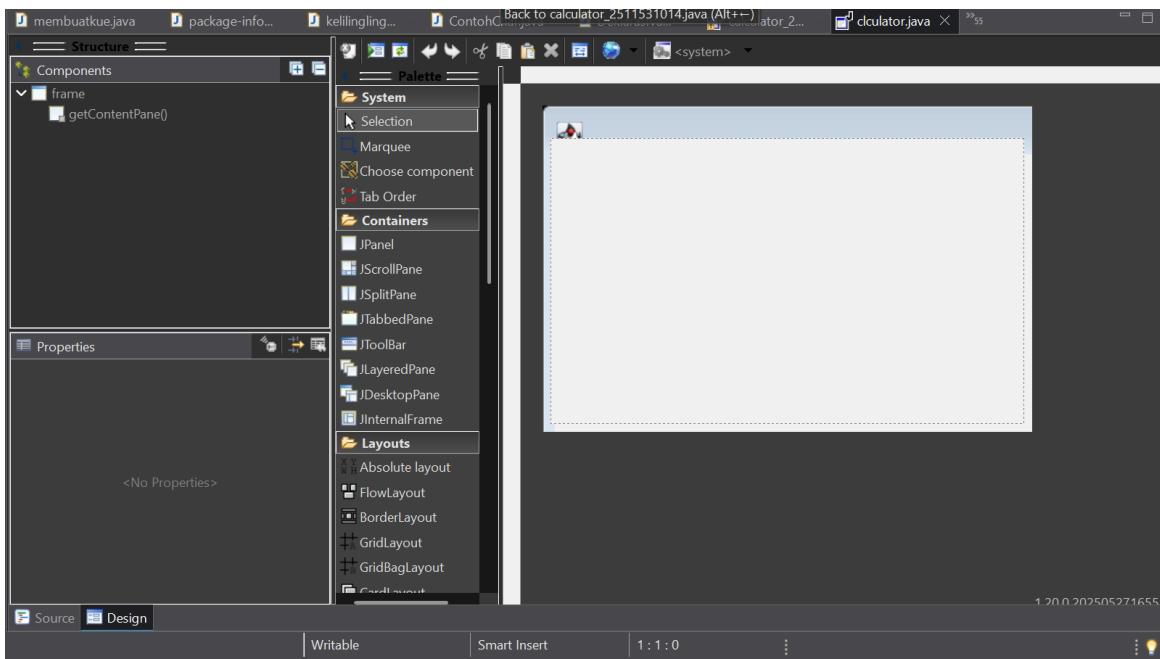
- Pertama buka apl Eclipse IDE for Java Developers, lalu klik kanan klik other pilih WindowBuilder dan pilih/klik Application Window.



Setelah tampilan ini lanjutkan dengan membuat nama package “pekan9\_NIM” dan buat nama class “Calculator\_NIM.” Dan klik “Finish”. Maka akan kelas baru, dengan tab “Source” dan “Design”.



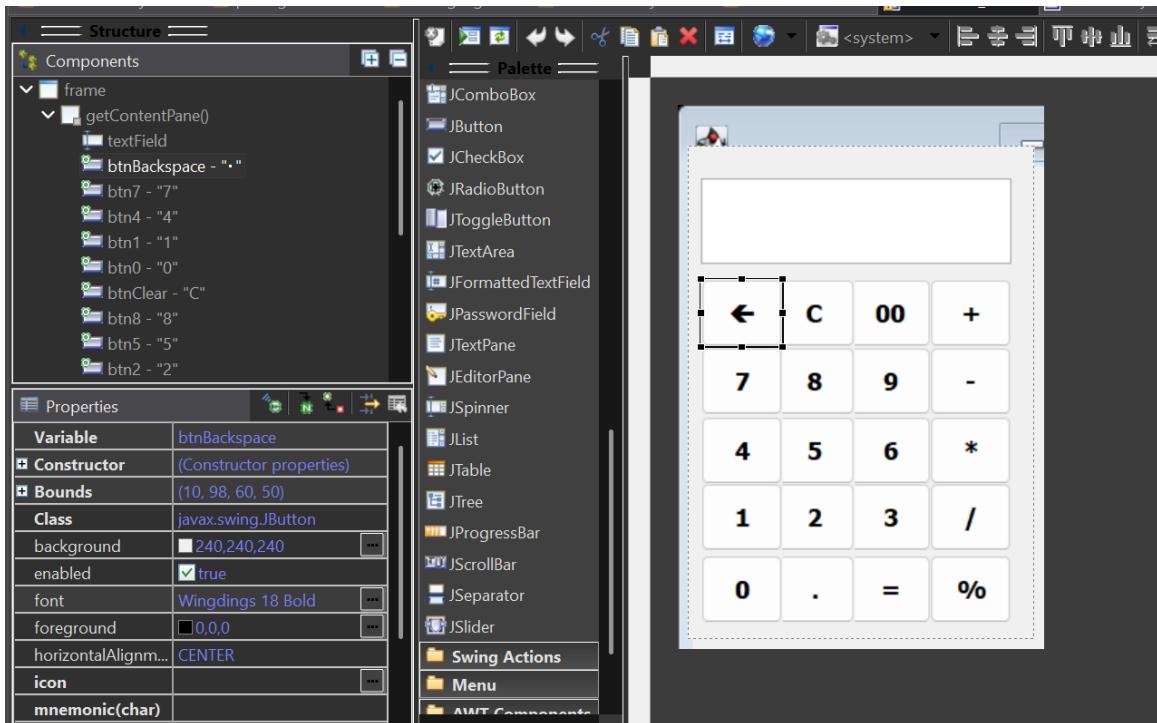
```
1 package pekan9_2511531014;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6
7 public class clculator {
8
9     private JFrame frame;
10
11    /**
12     * Launch the application.
13     */
14    public static void main(String[] args) {
15        EventQueue.invokeLater(new Runnable() {
16            public void run() {
17                try {
18                    clculator window = new clculator();
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 clculator() {
31        initialize();
32    }
33}
```



### 2.1.2 Membuat Design GUI (Calculator)

1. User dapat mulai membuat desain tampilan antar muka program dengan drag and drop. Pertama ubah layout menjadi **Absolut Layout** lalu tambahkan **JTextField**, drag and drop ke panel atur sizenya.
2. Menambahkan Komponen GUI dengan menambahkan **JButton**, drag and drop ke panel. Maka akan muncul text box “New Button” tambahkan JButton sebanyak 4 kesamping dan 5 kebawah, dengan ukuran yang dapat disesuaikan
3. Mulai mengatur property dari tiap komponen, seperti memilih font, alignment, warna dan lainnya. Misalkan disini saya mengedit component “New Button” menjadi angka yang diperlukan di dalam kalkulator.

Akan menghasilkan sebuah Design seperti gambar dibawah



### 2.1.3 Membuat Syntax

Mulai menambahkan Kode Program.

1. Mendeklarasikan Kelas Utama dari Kalkulator dan Variabel yang digunakan

```
public class calculator_2511531014 {

    private JFrame frame;
    private JTextField textField;

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

2. Menambahkan Syntax untuk JButton

1) Button Backspace

```
63         JButton btnBackspace = new JButton("\uF0E7");
64         btnBackspace.addActionListener(new ActionListener() {
65             public void actionPerformed(ActionEvent e) {
66                 String backSpace=null;
67                 if(textField.getText().length()>0)
68                 {
69                     StringBuilder str=new StringBuilder(textField.getText());
70                     str.deleteCharAt(textField.getText().length()-1);
71                     backSpace=str.toString();
72                     textField.setText(backSpace);
73                 }
74             });
75     };
```

2) Button Clear (C/AC)

```
JButton btnClear = new JButton("C");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
    }
});
btnClear.setFont(new Font("Tahoma", Font.BOLD, 18));
btnClear.setBounds(63, 98, 60, 50);
frame.getContentPane().add(btnClear);
```

3) Button Angka (00, 0-9)

```
JButton btn0 = new JButton("0");
btn0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String number=textField.getText()+btn0.getText();
        textField.setText(number);
    }
});
btn0.setFont(new Font("Tahoma", Font.BOLD, 18));
btn0.setBounds(10, 301, 60, 50);
frame.getContentPane().add(btn0);
```

4) Button Operator (+, -, /, %)

```
public void actionPerformed(ActionEvent arg0) {
    first = Double.parseDouble(textField.getText());
    textField.setText("");
    operation = "+";
}
});
btnPlus.setFont(new Font("Tahoma", Font.BOLD, 18));
btnPlus.setBounds(178, 98, 60, 50);
frame.getContentPane().add(btnPlus);
}
```

5) Button Hasil (=)

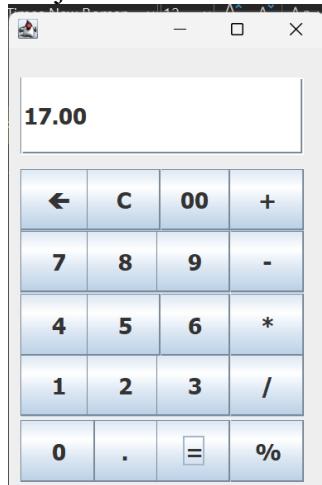
```

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);
        }
    }
})

```

Dari Syntax dan Design diatas dapat dihasilkan output seperti berikut

- 1) Penjumlahan dari 8+9



- 2) Pengurangan dari 13-4

- 3) Perkalian dari 3\*7
- 4) Pembagian dari 27/9
- 5) Modulus dari 9%5

- 1) Penjumlahan dari 8+9

- 2) Pengurangan 13-4

A screenshot of a Windows-style calculator application. The display shows "17.00". The keyboard layout is standard, with numeric keys 0-9, arithmetic operators (+, -, \*, /), and function keys like C, %, and =.

3) Perkalian dari  $3*7$

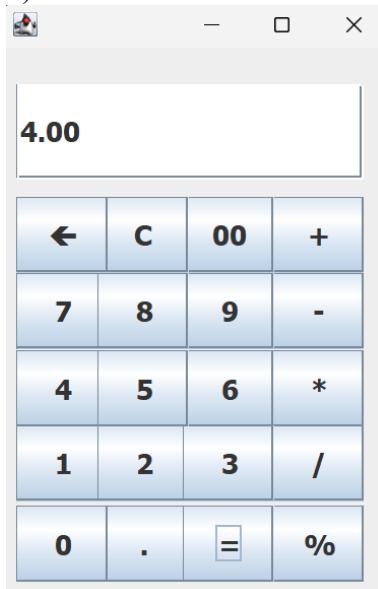
A screenshot of a Windows-style calculator application. The display shows "21.00". The keyboard layout is standard, with numeric keys 0-9, arithmetic operators (+, -, \*, /), and function keys like C, %, and =.

A screenshot of a Windows-style calculator application. The display shows "9.00". The keyboard layout is standard, with numeric keys 0-9, arithmetic operators (+, -, \*, /), and function keys like C, %, and =.

4) Pembagian  $27/9$

A screenshot of a Windows-style calculator application. The display shows "3.00". The keyboard layout is standard, with numeric keys 0-9, arithmetic operators (+, -, \*, /), and function keys like C, %, and =.

5) Modulus 9%5



## **BAB III**

### **PENUTUP**

#### **3.1. Kesimpulan**

Dari seluruh rangkaian praktikum pembuatan Calculator berbasis GUI ini, saya dapat menyimpulkan bahwa penerapan/pemakaian Components antarmuka seperti JTextField, JButton sangat membantu dalam membangun antarmuka program yang interaktif. Program berjalan dengan baik, di mana user dapat memasukkan angka, memilih operasi (+, -, \*, /, dan %), dan memperoleh hasil perhitungan secara otomatis. Setiap tombol berhasil memicu event yang sesuai, termasuk fitur pendukung seperti Clear dan Backspace.

Secara keseluruhan Praktikum kali ini sangat membantu dan mempermudah user dalam memahami dan membuat program berupa kalkulator yang sangat berguna dan sederhana.

#### **3.2. Saran**

Saran untuk praktikum selanjutnya adalah agar mahasiswa lebih teliti dalam memahami penulisan component yang tepat sesuai dengan kebutuhan program. Selain itu, penting untuk membedakan dengan benar antara tipe data String dan numerik agar tidak terjadi kesalahan logika pada proses perhitungan maupun tampilan output. Dan diharapkan agar mahasiswa lebih teliti dalam memahami syntax agar meminimalisir kesalahan penulisan kode.

## **DAFTAR PUSTAKA**

Sumber Daring (website):

[1] Tech Projects, “GUI Calculator in JAVA using eclipse ide,” YouTube, Tersedia pada: <https://www.youtube.com/watch?v=mZVD5OssHQM>.

[2] Eclipse Foundation, “WindowBuilder – Eclipse Project.” Tersedia pada: <https://eclipse.dev/windowbuilder/>.

[3] GeeksforGeeks, “Introduction to Java Swing.” [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/java/introduction-to-java-swing/>. [