

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
Pembuatan Calculator GUI di Java Menggunakan Eclipse

Disusun Oleh:

Dinda Amelia

(2511531020)

Dosen Pengampu:

Dr. Wahyudi, S.T, M.T

Asisten Praktikum:

Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG 2025

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan kesempatan serta kemudahan kepada penulis untuk dapat menyelesaikan Laporan Praktikum pada mata kuliah Algoritma Pemrograman, sehingga Laporan Praktikum ini dapat dikumpulkan dengan tepat waktu. Atas rahmat dan karunianya Laporan Praktikum dapat terselesaikan dengan baik. Sholawat serta salam juga penulis sampaikan kepada baginda tercinta yaitu Nabi Muhammad SAW, yang kita nantikan syafa'atnya di akhirat nanti. Laporan Praktikum ini bertujuan untuk menambah wawasan para pembaca untuk lebih memperdalam ilmu yang ada pada makalah ini.

Laporan pratikum ini disusun sebagai bentuk penanggung jawaban atas pelaksanaan kegiatan pratikum mata kuliah algoritma pemograman yang membahas tentang Pembuatan Aplikasi Operator Aritmatika Menggunakan Desain GUI Java Eclipse. Dalam praktikum ini, penulis tidak hanya belajar tentang Penyusunan laporan ini bertujuan untuk memberikan pemahaman mengenai cara membuat aplikasi kalkulator sederhana menggunakan komponen Java Swing seperti JFrame, JLabel, JTextField, dan JButton, serta bagaimana mengimplementasikan event handling pada tombol operasi. Selain itu, laporan ini juga diharapkan dapat menjadi referensi pembelajaran bagi mahasiswa dalam memahami pembuatan aplikasi berbasis GUI.

Dalam penyusunan Laporan Praktikum ini, penulis mengalami banyak kesulitan dan penulis menyadari bahwa Laporan Praktikum ini masih jauh dari kesempurnaan. Untuk itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan pada Laporan Praktikum ini.

Padang, 19 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I: PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Manfaat Praktikum.....	2
BAB II: PEMBAHASAN	3
2.1 Dasar Teori.....	3
2.2 Langkah kerja.....	4
2.2.1 Buka eclipse	4
2.2.2 Membuat Package dan class	4
2.2.3 Mendesign GUI.....	5
2.2.4 Source Code	9
2.2.5 Outputnya.....	15
BAB III: KESIMPULAN	17
DAFTAR PUSTAKA.....	18

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi menuntut mahasiswa untuk memahami dan menguasai pemrograman, khususnya dalam pembuatan aplikasi yang interaktif dan mudah digunakan. Salah satu kemampuan penting dalam dunia pemrograman adalah memahami *Graphical User Interface* (GUI), yaitu tampilan antarmuka grafis yang mempermudah pengguna dalam mengoperasikan sebuah aplikasi tanpa harus menggunakan perintah berbasis teks. GUI banyak diterapkan dalam berbagai aplikasi modern, sehingga penguasaan konsep ini menjadi hal yang sangat penting bagi calon programmer.

Java merupakan salah satu bahasa pemrograman yang banyak digunakan dalam dunia pendidikan maupun industri karena bersifat *portable*, mudah dikembangkan, dan mendukung pembuatan aplikasi berbasis GUI melalui *library* seperti *Java Swing*. Dengan memanfaatkan lingkungan pengembangan *Eclipse IDE*, proses pembuatan aplikasi menjadi lebih cepat dan terstruktur, termasuk dalam merancang tampilan dan menambahkan *event* atau aksi pada komponen-komponen GUI.

Pembuatan aplikasi kalkulator berbasis GUI menjadi proyek yang tepat untuk memahami dasar-dasar pembuatan antarmuka, penggunaan komponen visual, serta implementasi logika pemrograman dalam bentuk yang lebih nyata. Melalui praktikum ini, mahasiswa dapat mempelajari bagaimana menghubungkan input pengguna dengan proses komputasi menggunakan event handling, sekaligus memahami proses perancangan aplikasi berbasis *desktop*.

Dengan demikian, praktikum pembuatan kalkulator GUI ini diharapkan dapat memberikan pengalaman langsung dalam mengembangkan aplikasi sederhana yang interaktif, serta memperkuat pemahaman mahasiswa mengenai pemrograman Java, pengerjaan proyek berbasis *IDE Eclipse*, dan konsep GUI secara menyeluruh.

1.2 Tujuan

1. Mempelajari cara membuat aplikasi *Java* berbasis GUI menggunakan *Eclipse*.
2. Menggunakan komponen *Swing* seperti *JFrame*, *JButton*, *JLabel*, dan *TextField*.
3. Mengimplementasikan operasi aritmatika dasar ke dalam *event* tombol (*ActionListener*).
4. Melatih kemampuan merancang tampilan aplikasi dan menghubungkannya dengan logika program.
5. Menghasilkan aplikasi kalkulator sederhana yang interaktif dan mudah digunakan.

1.3 Manfaat Praktikum

1. Mahasiswa memahami cara membuat aplikasi GUI di *Java* dengan tampilan yang lebih menarik.
2. Meningkatkan kemampuan dalam menggunakan *Eclipse* sebagai IDE profesional.
3. Menambah pemahaman tentang *event handling* seperti *ActionListener*.
4. Dapat membuat aplikasi *desktop* sederhana yang dapat dikembangkan lebih lanjut.
5. Meningkatkan keterampilan pemrograman berorientasi objek (OOP) melalui implementasi komponen GUI.

BAB II

PEMBAHASAN

2.1 Dasar Teori

1. Java Swing

Java Swing adalah *library* GUI yang digunakan untuk membuat aplikasi berbasis tampilan (*desktop app*). Komponen umum dalam *Swing* termasuk:

- *JFrame*: jendela utama aplikasi
- *JLabel*: menampilkan teks
- *TextField*: input data
- *JButton*: tombol aksi *Swing* berjalan di atas AWT dan bersifat *lightweight* sehingga fleksibel untuk membuat antarmuka.

2. Event Handling

Event Handling adalah mekanisme *Java* untuk menangani aksi pengguna, seperti menekan tombol.

Dalam kalkulator, ketika tombol (+, -, *, /) ditekan, program akan:

- mengambil *input* angka
- menghitung hasil sesuai operasi
- menampilkan *output* ke *TextField*

3. Eclipse IDE

Eclipse adalah lingkungan pengembangan *Java* yang menyediakan:

- editor code
- debug tools
- *GUI builder (Window Builder)* Digunakan untuk membuat program *Java* dengan cepat dan efisien.

4. Operator Aritmatika

Program kalkulator menggunakan operator dasar:

- + penjumlahan
- - pengurangan
- * perkalian
- / pembagian

2.2 Langkah kerja

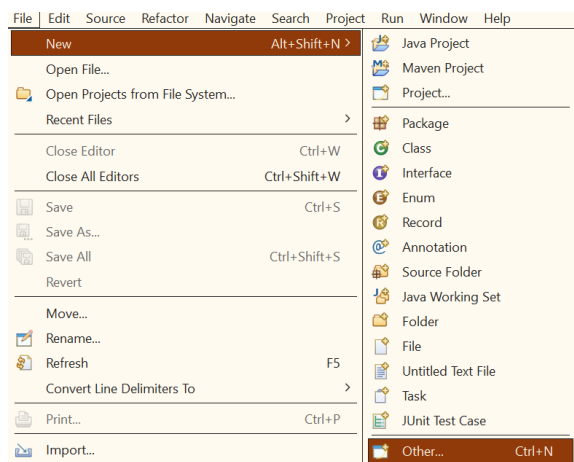
2.2.1 Buka eclipse



Gambar 2.1 Buka eclipse jalankan

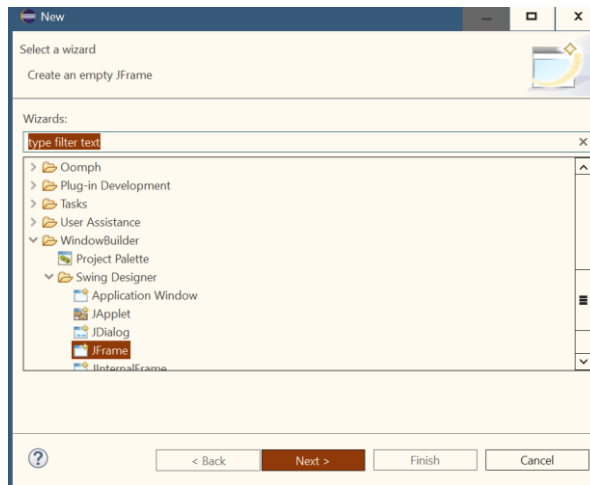
2.2.2 Membuat Package dan class

- Klik folder src
- Klik file pilih New dan Other



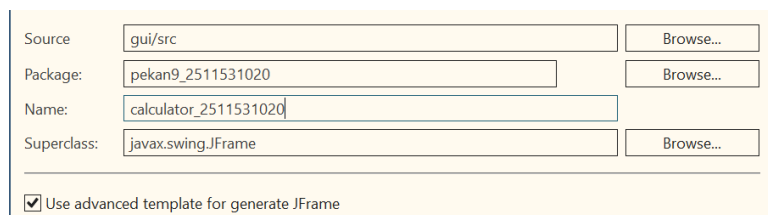
Gambar 2.2 Buka eclipse jalankan

- pilih JFrame Klik next



Gambar 2.3 Mencari JFrame

- Membuat package pekan9_2511531020 dan class calculator_2511531020

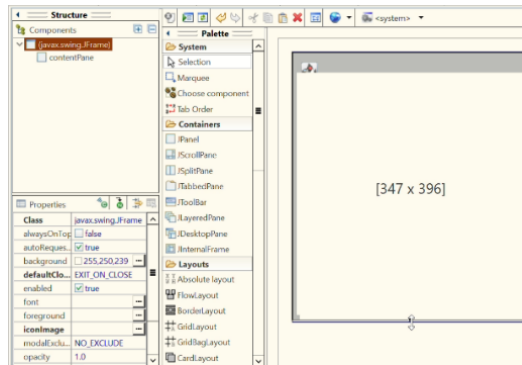


Gambar 2.4 Membuat Nama package dan class

- Klik Finish
- File form akan terbuka dengan dua mode: Source → melihat kode Design → mendesain GUI secara visual
- Pilih Design

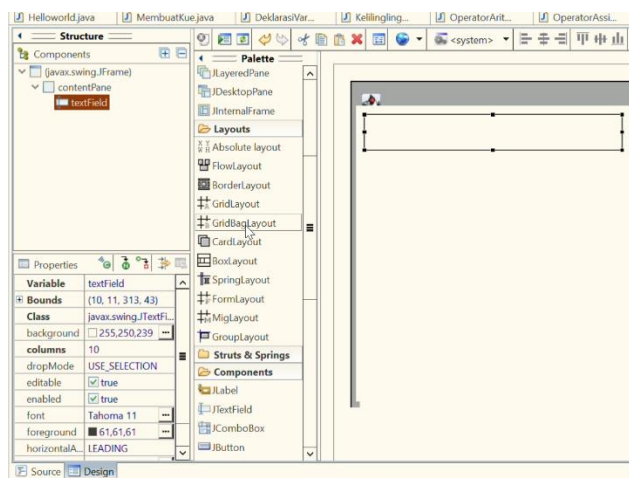
2.2.3 Mendesign GUI

- Atur ukuran size Jframenya dan klik absolute



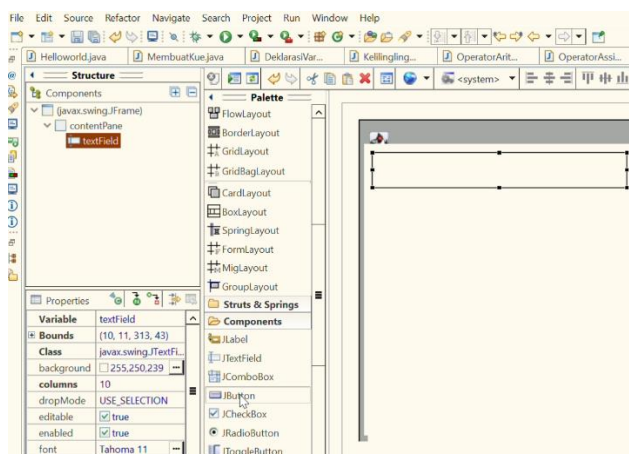
Gambar 2.5 Mendesign GUI

- Pilih JTextField atur juga ukurannya



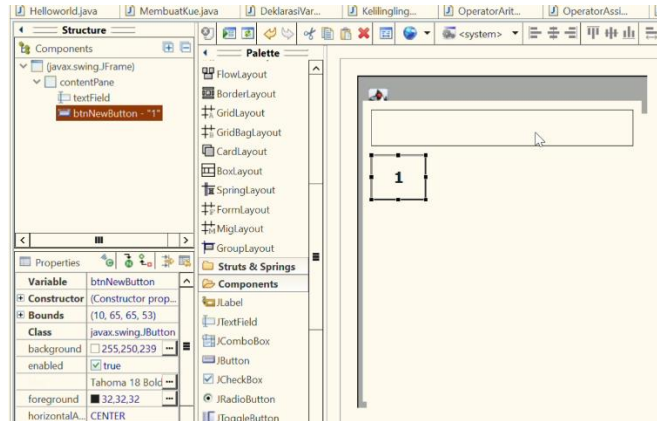
Gambar 2.6 JTextField atur juga ukurannya

- Pilih JButton



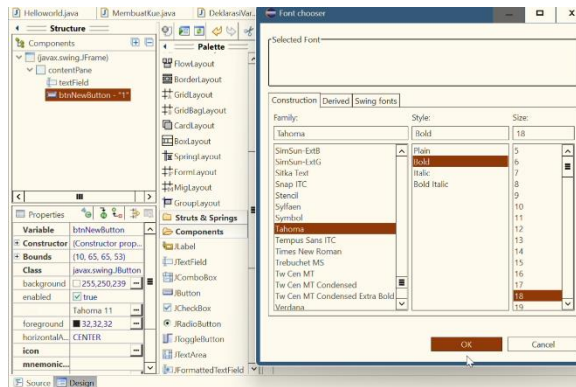
Gambar 2.7 JButton untuk membuat antarmuka.

Atur size, ubah Text → 1, dan ganti Variable → btn1



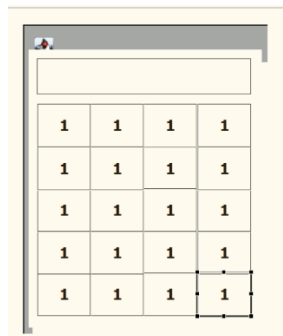
Gambar 2.8 JButton untuk membuat antarmuka.

- Atur ketebalan dari JButton



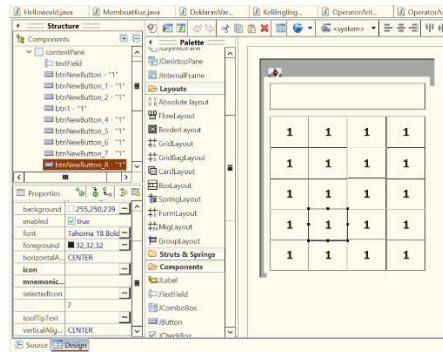
Gambar 2.9 Atur ketebalan dari JButton

- Copy paste semua JButton dengan 5 vertikal dan 4 horizontal



Gambar 2.10 Atur ketebalan dari JButton

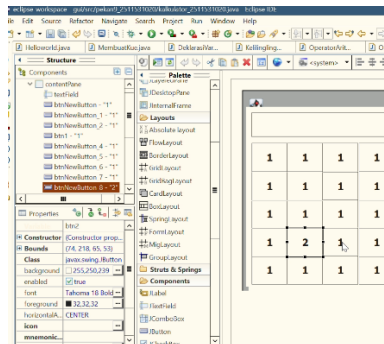
- Text 2



Gambar 2.11 Atur Jbutton Text 2

Ganti text 1 menjadi 2 ,lakukan dari 0-9, 00, dan .

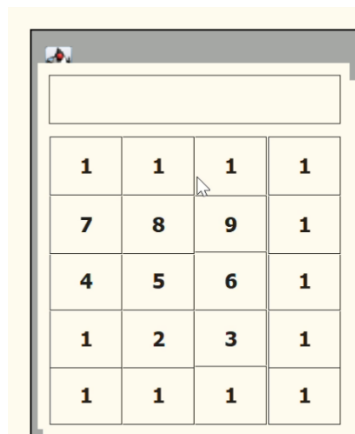
- Variable btn2



Gambar 2.12 Atur Jbutton btn2

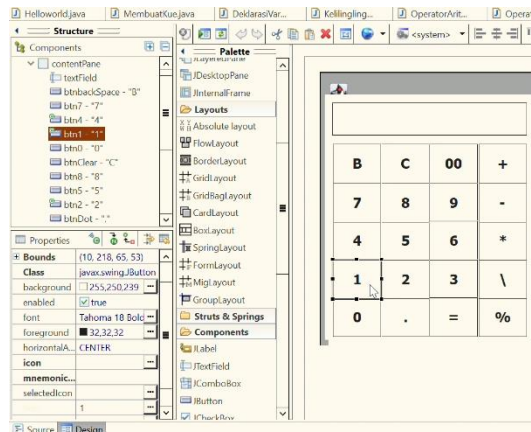
Ganti variable btn1 menjadi btn2 ,lakukan dari btn0-btn9, btnDot

- Hasil sementara dari Variable angka



Gambar 2.13 Hasil sementara dari Variable angka

- Tambahkan tombol operasi (+,-,/,%) dan (Clear & BackSpace)



Gambar 2.14 Tambahkan tombol operasi (+,-,/,%) dan (Clear & BackSpace)

Tambahkan semua elemen dengan cara yang sama dengan

Variable →backSpace dengan Text → B

Variable →Clear dengan Text → C

Variable →Plus dengan Text → +

Variable →Sub dengan Text → -

Variable →Mul dengan Text → *

Variable →Divide dengan Text → \

Variable →Percent dengan Text → %

Variable →Equal dengan Text → =

2.2.4 Source Code

- Tambahkan import untuk Swing dan event

```
import java.awt.EventQueue;
```

```
import javax.swing.*;
```

```
import javax.swing.border.EmptyBorder;
```

```
import java.awt.Font;
```

```
import java.awt.event.*;
```

Penjelasan: Import ini menyediakan komponen GUI (JFrame, JButton, JTextField) dan event handling (ActionListener).

- Definisikan state kalkulator
- Langkah: Di dalam kelas, deklarasikan:

```

17 private JPanel contentPane;
18 private JTextField textField;
19
20 double first;
21 double second;
22 double result;
23 String operation;

```

Gambar 2.15 Definisikan state kalkulator

Penjelasan: Variabel ini menyimpan angka pertama, angka kedua, hasil, dan jenis operasi yang dipilih. textField menampilkan input dan output.

- Buat main thread untuk GUI
- Langkah: Gunakan EventQueue:

```

29 public static void main(String[] args) {
30     EventQueue.invokeLater(new Runnable() {
31         public void run() {
32             try {
33                 calculator_2511531020 frame = new calculator_2511531020();
34                 frame.setVisible(true);
35             } catch (Exception e) {
36                 e.printStackTrace();
37             }
38         }
39     });
40 }

```

Gambar 2.16 Buat main thread untuk GUI

Penjelasan: Sg wajib dijalankan di Event Dispatch Thread (EDT) agar GUI responsif dan bebas race condition.

- Inisialisasi frame dan panel
- Langkah: Dalam konstruktor:

```

46 setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
47 setBounds(100, 100, 294, 400);
48 contentPane = new JPanel();
49 contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
50 setContentPane(contentPane);
51 contentPane.setLayout(null);

```

Gambar 2.17 Inisialisasi frame dan panel

Penjelasan: Menentukan ukuran jendela, membuat panel utama, dan memilih layout. Di sini digunakan koordinat absolut via `setLayout(null)`.

- Tambahkan field tampilan angka
- Langkah: Buat dan posisikan `TextField`:

```
70     textField = new JTextField();
71     textField.setFont(new Font("Tahoma", Font.BOLD, 18));
72     textField.setBounds(10, 11, 262, 65);
73     contentPane.add(textField);
74     textField.setColumns(10);
```

Gambar 2.18 Tambahkan field tampilan angka

Penjelasan: Field ini berfungsi sebagai layar kalkulator (input dan output). Font besar agar mudah dibaca.

- Tambahkan tombol utilitas (Backspace, Clear, 00)
- Langkah: Contoh Backspace:

```
JButton btnBackspace = new JButton("\uF0E7");
btnBackspace.setFont(new Font("wingdings", Font.BOLD,
18));
btnBackspace.setBounds(10, 99, 65, 51);
btnBackspace.addActionListener(e -> {
    String txt = textField.getText();
    if (txt.length() > 0) {
        textField.setText(new
StringBuilder(txt).deleteCharAt(txt.length()-1).toString());
    }
});
contentPane.add(btnBackspace);
```

Penjelasan: Backspace menghapus satu karakter terakhir. Tombol Clear “C” mengosongkan layar. Tombol “00” menambah dua nol sekaligus.

- Tambahkan tombol angka dan titik
- Langkah: Pola umum tombol angka:

```

76 JButton btn7 = new JButton("7");
77 btn7.addActionListener(new ActionListener() {
78     public void actionPerformed(ActionEvent e) {
79         String number=textField.getText()+btn7.getText();
80         textField.setText(number);
81     }
82 });
83 btn7.setFont(new Font("Tahoma", Font.BOLD, 18));
84 btn7.setBounds(10, 149, 65, 51);
85 contentPane.add(btn7);

```

Gambar 2.19 Tambahkan tombol angka dan titik

Ulangi untuk 0–9 dan “.” sesuai koordinat di kode.

Penjelasan: Setiap tombol menambahkan teks angka/titik ke textField. Ini membangun input numerik.

- Tambahkan tombol operasi (+,-,/,%)
- Langkah: Pola umum:

```

260 JButton btnPlus = new JButton("+");
261 btnPlus.addActionListener(new ActionListener() {
262     public void actionPerformed(ActionEvent e) {
263         first=Double.parseDouble(textField.getText());
264         textField.setText("");
265         operation="+";
266     }
267 });
268 btnPlus.setFont(new Font("Tahoma", Font.BOLD, 18));
269 btnPlus.setBounds(207, 99, 65, 51);
270 contentPane.add(btnPlus);

```

Gambar 2.20 Tambahkan tombol operasi (+,-,/,%)

Buat tombol lain dengan operasi masing-masing dan posisi sesuai.

Penjelasan: Saat operasi ditekan, aplikasi menyimpan angka pertama dan menunggu angka kedua. Layar dikosongkan agar pengguna memasukkan angka berikutnya.

- Tambahkan tombol sama dengan (=) untuk menghitung

- Langkah: Hitung berdasarkan operasi:

```

218 JButton btnEqual = new JButton("=");
219 btnEqual.addActionListener(new ActionListener() {
220     public void actionPerformed(ActionEvent e) {
221         String answer;
222         second=Double.parseDouble(textField.getText());
223         if(operation=="+")
224         {
225             result=first+second;
226             answer=String.format("%.2f", result);
227             textField.setText(answer);
228         }
229         else if(operation=="-")
230         {
231             result=first-second;
232             answer=String.format("%.2f", result);
233             textField.setText(answer);
234         }
235         else if(operation=="*")
236         {
237             result=first*second;
238             answer=String.format("%.2f", result);
239             textField.setText(answer);
240         }
241         else if(operation=="/")
242         {
243             result=first/second;
244             answer=String.format("%.2f", result);
245             textField.setText(answer);
246         }
247         else if(operation=="%")
248         {

```

Gambar 2.21 Tambahkan tombol sama dengan (=) untuk

```

249             result=first%second;
250             answer=String.format("%.2f", result);
251             textField.setText(answer);
252         }
253     }
254 }
255 });
256 btnEqual.setFont(new Font("Tahoma", Font.BOLD, 18));
257 btnEqual.setBounds(142, 296, 65, 51);
258 contentPane.add(btnEqual);

```

Gambar 2.22 Tambahkan tombol sama dengan (=) untuk menghitung

Penjelasan: = mengambil angka kedua, memilih operasi yang disimpan, menghitung hasil, lalu menampilkan dengan format dua desimal.

Perbaikan penting pada kode (agar bebas bug)

- Gunakan equals untuk perbandingan String
- Masalah: Kode kamu memakai operation == "+" dan sejenisnya. Di Java, ini membandingkan referensi, bukan isi string.

```

Perbaikan: if ("+".equals(operation)) { ... }
            else if ("-".equals(operation)) { ... }
            // dan seterusnya

```

- **Penjelasan:** String.equals(...) membandingkan konten string dengan benar. Menghindari bug tak terduga saat == gagal.
- Tambahkan helper untuk parsing input dengan aman
- **Langkah:** Buat metode:

```

private double parseInput(String s) {
    if (s == null || s.trim().isEmpty()) return 0.0;
    return Double.parseDouble(s);
}

```

Penjelasan: Menghindari (NumberFormatException) saat field kosong atau spasi.

- Tangani Pembagian dengan nol
- **Langkah:** Di compute:

```

private double compute(double a, double b, String op) {
    switch (op) {
        case "+": return a + b;
        case "-": return a - b;
        case "*": return a * b;
        case "/": return b == 0 ? Double.NaN : a / b;
        case "%": return b == 0 ? Double.NaN : a % b;
        default: return 0.0;
    }
}

```

Penjelasan: Menampilkan NaN saat pembagian/modulus dengan nol mencegah crash dan menjelaskan kesalahan input.

- Validasi input titik decimal
 - **Langkah:** Batasi hanya satu “.”:
- ```

btnDot.addActionListener(e -> {

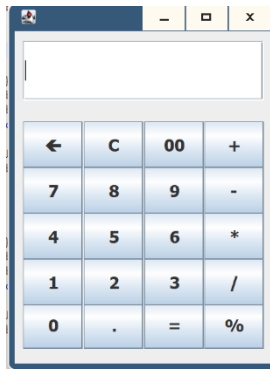
```

```
String txt = textField.getText();
if (!txt.contains(".")) textField.setText(txt + ".");
});
```

Penjelasan: Mencegah input seperti “12.3.4” yang menyebabkan error saat parsing.

- Menjalankan dan Menguji Aplikasi
- Jalankan program: - Langkah: Klik kanan class → Run As → Java Application

### 2.2.5 Outputnya



**Gambar 2.23 Outputnya**

- Uji setiap fitur
- Langkah:
  - Penjumlahan: Masukkan “7” → “+” → “8” → “=” → hasil “15.00”.

Contoh:



**Gambar 2.24 Outputnya**

- ☐ Pengurangan: “50” → “-” → “17.5” → “=” → “32.50”.
- ☐ Perkalian: “7” → “\*” → “6” → “=” → “42.00”.
- ☐ Pembagian: “9” → “/” → “4” → “=” → “2.25”.
- ☐ Modulus: “10” → “%” → “3” → “=” → “1.00”.
- ☐ Backspace: Ketik “123” → Backspace → “12”.
- ☐ Clear: Ketik angka → “C” → kosong.

Penjelasan: Pengujian sistematis memastikan semua alur bekerja dan memudahkan pelacakan bug.

## **BAB III**

### **KESIMPULAN**

Berdasarkan praktikum pembuatan Calculator GUI di Java Menggunakan Eclipse dapat di Kesimpulan dari pembuatan program kalkulator berbasis Java Swing ini menunjukkan bahwa aplikasi berhasil dibangun dengan memanfaatkan komponen GUI seperti JFrame, JPanel, JTextField, dan JButton, serta menerapkan konsep event handling melalui ActionListener. Program mampu menjalankan operasi aritmatika dasar seperti penjumlahan, pengurangan, perkalian, pembagian, dan modulus dengan hasil yang ditampilkan secara akurat dalam format dua angka di belakang koma. Selain itu, fitur tambahan seperti tombol backspace, clear, titik desimal, dan angka ganda (“00”) turut memperkaya fungsionalitas kalkulator. Penggunaan layout absolut (setLayout(null)) memberikan kontrol penuh terhadap posisi komponen, meskipun kurang fleksibel untuk tampilan dinamis. Secara keseluruhan, proyek ini berhasil memenuhi tujuan pembelajaran dengan mengintegrasikan logika perhitungan dan antarmuka grafis secara efektif, serta memberikan pengalaman praktis dalam membangun aplikasi desktop sederhana menggunakan bahasa pemrograman Java.

## DAFTAR PUSTAKA

- [1] V. Siahaan and R. H. Sianipar, *Buku Pintar JAVA GUI dengan ECLIPSE*, Balige Publishing, 2020.
- [2] Oracle, “The Java™ Tutorials – Creating a GUI With Swing,” Oracle Corporation, 2023 [online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/>
- [3] Siregar, R. F., Al-Khowarizmi, A. K., & Nasution, A. R. (2024). DESIGN OF A 4 BAND COLOR CODE RESISTOR CALCULATOR APPLICATION BASED ON THE JAVA PROGRAMMING LANGUAGE. *RELE (Rekayasa Elektrikal dan Energi): Jurnal Teknik Elektro*, 6(2), 82-87.
- [4] McAllister, A., & Feng, M. Y. (2005). Automated grading of student-designed GUI programs. *World Transactions on Engineering and Technology Education*, 4(2), 209.
- [5] GitHub, “Kalkulator GUI Java (Modern Design) by Zaakiea,” GitHub Repository, 2021. [Online]. Available: [https://github.com/zaakiea/Kalkulator\\_GUIJava](https://github.com/zaakiea/Kalkulator_GUIJava)