

LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA PEMROGRAMAN
MEMBUAT CALKULATOR GUI



Disusun Oleh:
Gina Ramadhani
Nim: 2511533014

Dosen Pengampu: DR. Wahyudi, S.T, M.T
Asisten Pratikum: Rahmad Dwi Rizki Olders

DAPERTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “Membuat Calkulator GUI” pada Bahasa Pemrograman Java ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggung jawaban kegiatan praktikum serta sebagai sarana untuk memperdalam pemahaman mengenai penerapan konsep pemrograman berorientasi objek dan pembuatan antarmuka grafis (GUI) dalam bahasa pemrograman Java.

Melalui praktikum ini, penulis memperoleh pengalaman langsung dalam merancang, mengimplementasikan, dan menguji sebuah aplikasi kalkulator sederhana berbasis GUI. Kegiatan ini diharapkan dapat meningkatkan kemampuan mahasiswa dalam memanfaatkan pustaka *Java Swing/AWT*, memahami struktur program, serta mengembangkan aplikasi interaktif yang fungsional.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang. Penulis juga mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah memberikan bimbingan dan dukungan selama proses praktikum hingga penyusunan laporan ini.

Padang, 28 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB II	3
PEMBAHASAN	3
2.1 Dasar Teori.....	3
2.2 Langkah Kerja.....	5
BAB III.....	15
PENUTUP.....	15
3.1 Kesimpulan	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi menuntut kemampuan dalam membangun aplikasi yang tidak hanya berfungsi dengan baik, tetapi juga memiliki antarmuka yang interaktif dan mudah digunakan. Salah satu aspek penting dalam pengembangan perangkat lunak adalah pembuatan *Graphical User Interface (GUI)*, yang memungkinkan pengguna berinteraksi dengan aplikasi melalui tampilan grafis yang intuitif.

Bahasa pemrograman Java menyediakan berbagai *library* seperti *Swing* dan *AWT* yang memudahkan pengembang dalam merancang dan mengimplementasikan aplikasi berbasis *GUI*. Melalui penerapan konsep pemrograman berorientasi objek dan pemanfaatan komponen *GUI*, mahasiswa dapat memahami alur kerja sebuah aplikasi dan meningkatkan kemampuan *problem solving* dalam pengembangan perangkat lunak.

Pembuatan aplikasi kalkulator berbasis *GUI* menjadi salah satu contoh sederhana namun efektif untuk mempelajari dasar-dasar pembuatan antarmuka pengguna, pengelolaan *event (event handling)*, serta logika perhitungan dalam sebuah aplikasi. Praktikum ini memberikan kesempatan bagi mahasiswa untuk menerapkan teori yang telah dipelajari pada perkuliahan ke dalam bentuk aplikasi nyata.

1.2 Tujuan Praktikum

1. Memahami dasar-dasar pembuatan *Graphical User Interface (GUI)* menggunakan Java.
2. Mampu merancang dan membangun tampilan aplikasi kalkulator yang sederhana dan mudah digunakan.
3. Mengimplementasikan *event handling* untuk menangani aksi pengguna pada tombol-tombol kalkulator.
4. Mengembangkan logika perhitungan aritmatika dasar dalam aplikasi Java.

5. Melatih kemampuan pemrograman berorientasi objek (OOP) dalam pengembangan aplikasi berbasis *GUI*.

1.3 Manfaat Praktikum

1. Menambah pemahaman mengenai pembuatan antarmuka pengguna (*GUI*) menggunakan Java.
2. Melatih kemampuan dalam menggunakan komponen-komponen *GUI* seperti tombol, teks, dan *layout*.
3. Meningkatkan keterampilan dalam menerapkan *event handling* untuk interaksi pengguna.
4. Memberikan pengalaman praktis dalam mengimplementasikan konsep pemrograman berorientasi objek (OOP).
5. Menjadi dasar untuk mengembangkan aplikasi Java yang lebih kompleks di masa mendatang.

BAB II

PEMBAHASAN

2.1 Dasar Teori

1. Pemrograman Berorientasi Objek (Object-Oriented Programming – OOP)

Java merupakan bahasa pemrograman yang menerapkan paradigma pemrograman berorientasi objek. Konsep utama OOP meliputi:

- *Class dan Object*: Class adalah blueprint atau cetak biru, sedangkan object adalah *instance* dari class.
- *Encapsulation*: Pengelompokan data dan metode dalam satu unit untuk menjaga keamanan data.
- *Inheritance*: Pewarisan atribut dan metode dari satu class ke class lain.
- *Polymorphism*: Kemampuan suatu metode untuk memiliki bentuk berbeda tergantung konteks.

Konsep OOP menjadi dasar pengembangan aplikasi kalkulator karena program disusun dalam bentuk class yang menangani tampilan maupun logika perhitungan.

2. Graphical User Interface (GUI)

GUI adalah antarmuka yang memungkinkan pengguna berinteraksi dengan aplikasi melalui elemen visual seperti tombol, label, dan kotak teks. Dalam Java, pembuatan GUI umumnya menggunakan dua library utama:

- *AWT (Abstract Window Toolkit)*
- *Swing*, yang merupakan pengembangan dari AWT dengan fitur lebih lengkap.

Pada praktikum ini biasanya digunakan *Java Swing* karena menyediakan komponen GUI yang lebih modern, stabil, dan mudah dikembangkan.

3. Java Swing

Java Swing adalah *library* yang digunakan untuk membangun antarmuka grafis. Beberapa komponen penting dalam pembuatan kalkulator GUI antara lain:

- *JFrame*: Jendela utama aplikasi.

- *JPanel*: Kontainer yang menampung komponen lain.
- *JButton*: Tombol yang digunakan untuk input angka dan operator.
- *JTextField*: Kotak teks untuk menampilkan dan memasukkan angka.

Swing juga mendukung pengaturan tata letak (layout), misalnya:

- *FlowLayout*
- *GridLayout* → sering digunakan untuk kalkulator karena menampilkan tombol dalam bentuk grid.

4. Event Handling

Event handling adalah mekanisme untuk menangani aksi yang dilakukan oleh pengguna, seperti klik tombol. Java menggunakan konsep *event-driven programming*, di mana program merespons peristiwa yang terjadi.

Untuk menangani event pada komponen Swing, Java menggunakan interface seperti:

- *ActionListener*
yang akan dijalankan ketika tombol ditekan.

5. Logika Perhitungan Aritmatika

Aplikasi kalkulator membutuhkan penerapan logika dasar operasi aritmatika, seperti:

- Penjumlahan
- Pengurangan
- Perkalian
- Pembagian

Logika perhitungan biasanya diimplementasikan dalam method tersendiri atau melalui pengaturan state program untuk menyimpan nilai input pertama, operator, dan nilai berikutnya

6. IDE Eclipse

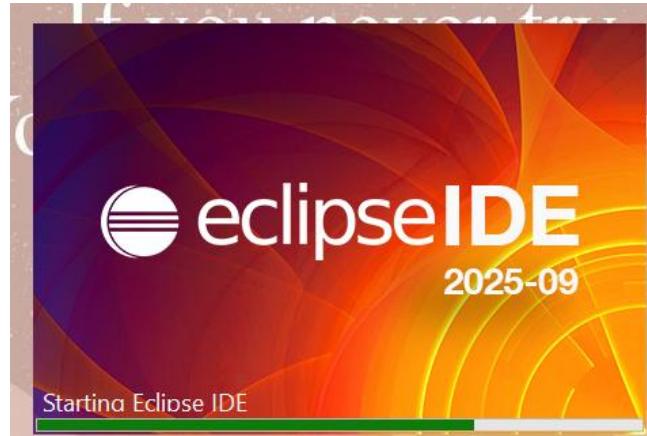
Eclipse adalah salah satu *Integrated Development Environment* (IDE) yang banyak digunakan untuk mengembangkan aplikasi Java. Kelebihannya:

- Mendukung penulisan kode yang terstruktur dan mudah dikelola.
- Memiliki fitur *auto-complete*, *debugging*, dan integrasi *library Swing*.
- Memudahkan pengorganisasian project melalui struktur folder yang jelas.

Penggunaan Eclipse membantu mahasiswa memahami pengembangan aplikasi Java dengan lingkungan kerja yang profesional.

2.2 Langkah Kerja

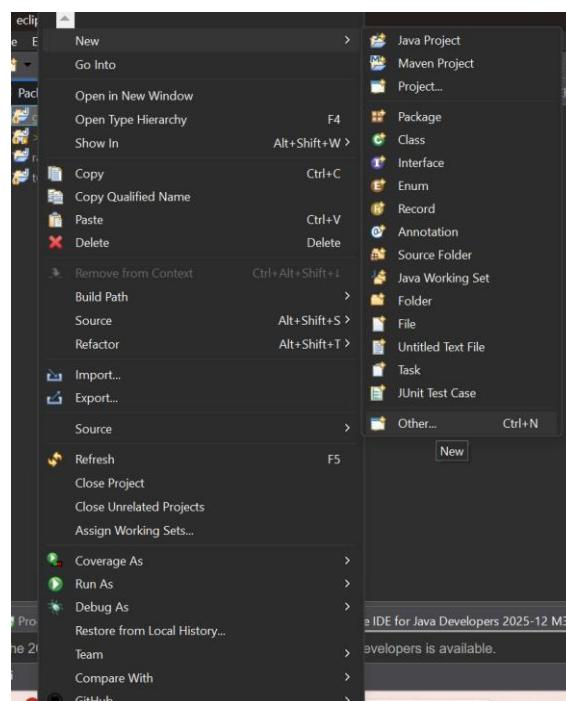
1. Membuka aplikasi Eclipse



Gambar 2.1 Membuka aplikasi eclipse

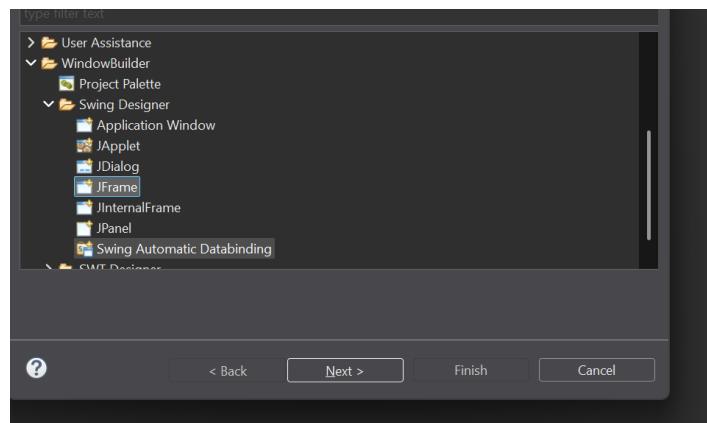
2. Membuat package dan class

- Klik folder src
- Klik file pilih New dan Other



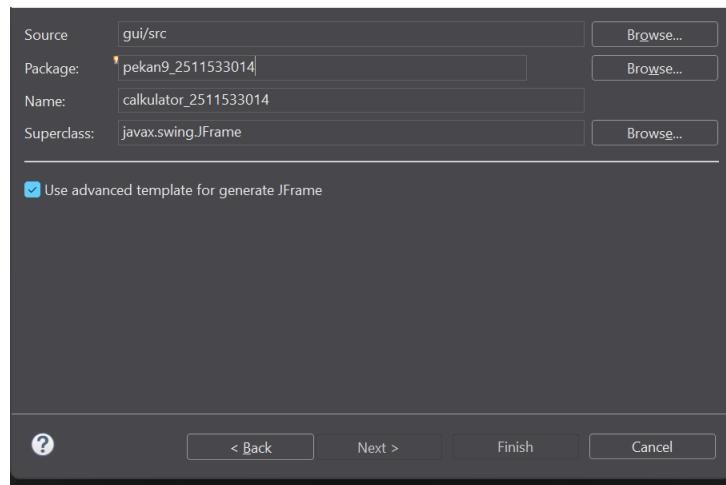
Gambar 2.2 Membuat package dan class

- pilih Jframe Klik next



Gambar 2.3 Mencari *JFrame*

- Membuat package pekan9_2511533014 dan class calkulator_2511533014

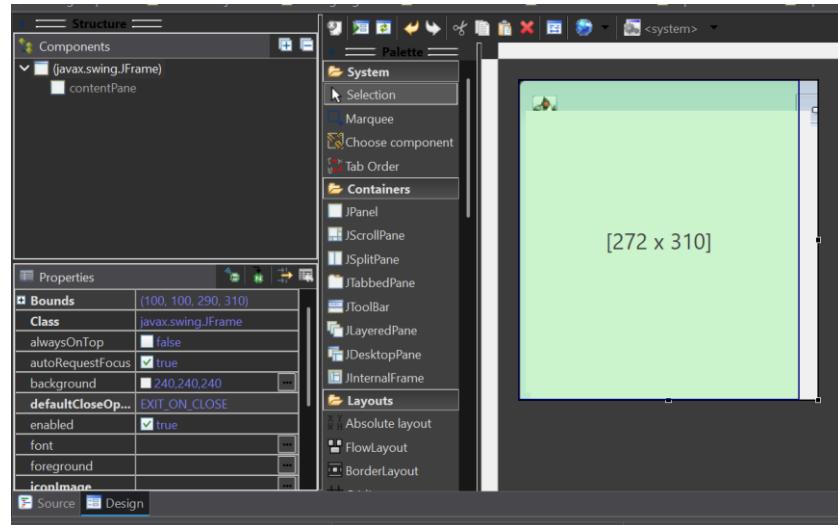


Gambar 2.4 Membuat Nama package dan class

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

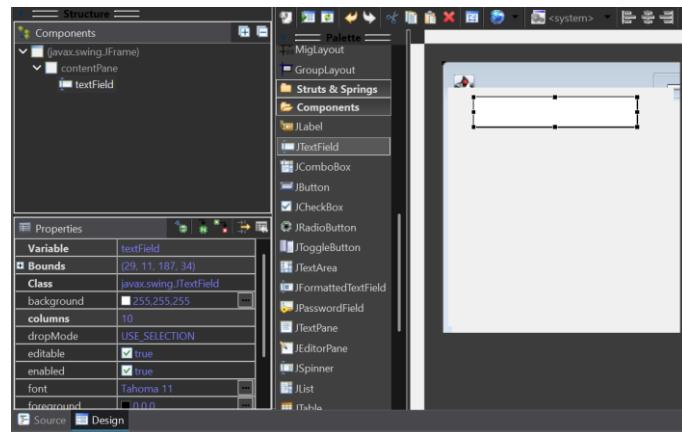
3. Mendesign GUI

- Atur ukuran size *Jframennya* dan klik *absolute*



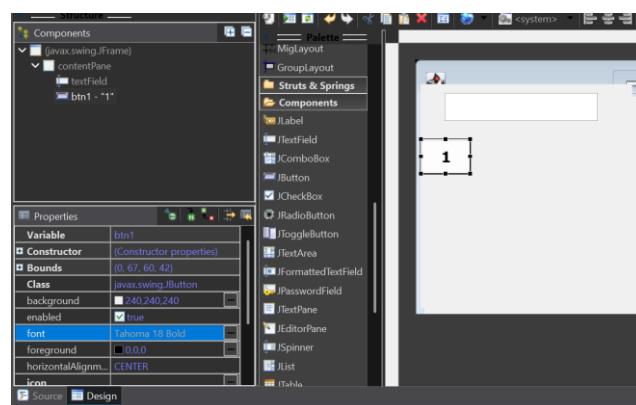
Gambar 2.5 Mengatur design GUI

- Pilih JTextField dan atur ukurannya



Gambar 2.6 Membuat dan mengatur ukuran JTextField

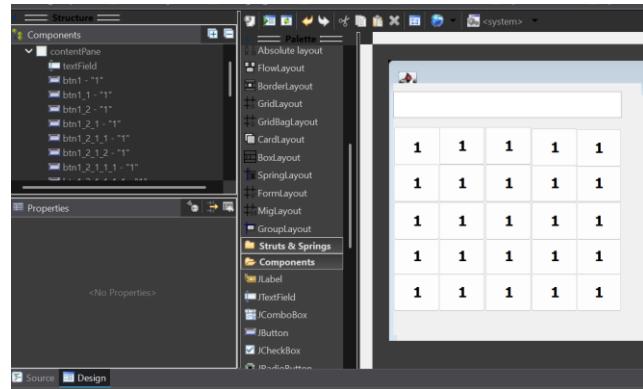
- Pilih JButton



Gambar 2.7 JButton untuk membuat antarmuka

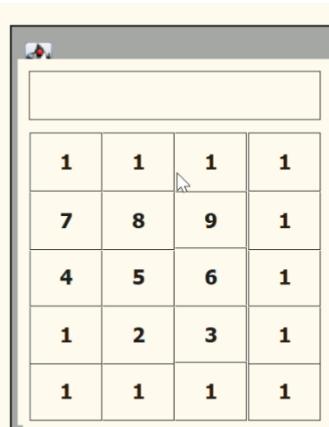
- Atur size, ubah Text → 1, dan ganti Variable → btn1
- Atur ketebalan dari JButton → Tahoma → Bold → 18

- Copy paste semua JButton dengan 5 vertikal dan 4 horizontal



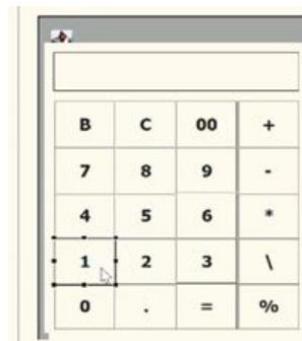
Gambar 2.8 Membuat JButton

- Ganti text 1 menjadi 2, lakukan dari 0-9, 00, dan.
- Ganti variable btn1 menjadi btn2 ,lakukan dari btn0-btn9, btnDot
- Hasil sementara dari Variable angka



Gambar 2.9 Hasil sementara dari Variable angka

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



Gambar 2.10 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 → =

4. Source code

1. Deklarasi Variabel Utama

Variabel ini digunakan untuk menyimpan nilai perhitungan dan operator yang dipilih pengguna.

```

1 package pekan9_2511533014;
2
3+ import java.awt.EventQueue;□
11
12 public class calkulator_2511533014 {
13
14     private JFrame frame;
15     private JTextField textField;
16
17     double first;
18     double second;
19     double result;
20     String operation;
21     String answer;
22

```

Gambar 2.11 Potongan source code

Penjelasan:

- first → nilai pertama sebelum memilih operator
- second → nilai kedua setelah operator
- result → hasil perhitungan
- operation → menyimpan jenis operasi seperti "+", "-", "*", "/", atau "%"

2. Membuat Frame Utama Aplikasi

Frame merupakan jendela utama yang menampilkan kalkulator.

```

51         frame = new JFrame();
52         frame.setBounds(100, 100, 275, 331);
53         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
54         frame.getContentPane().setLayout(null);
55

```

Gambar 2.12 Potongan source code

Penjelasan:

- Menentukan ukuran jendela
- Mengatur aksi saat aplikasi ditutup
- Layout null memberi kebebasan menempatkan komponen secara manual

3. TextField sebagai Layar Kalkulator

Tempat menampilkan angka dan hasil operasi.

```

55
56     textField = new JTextField();
57     textField.setBounds(10, 11, 212, 39);
58     textField.setFont(new Font("Tahoma", Font.BOLD, 18));
59     frame.getContentPane().add(textField);
60     textField.setColumns(10);

```

Gambar 2.13 Potongan source code

Penjelasan:

- JTextField untuk menampilkan input & output
- Mengatur ukuran font agar lebih jelas

4. Tombol Angka (0–9) dan Tombol Titik

Semua tombol angka dibuat dengan fungsi yang sama yaitu menambahkan angka ke textField.

Contoh tombol angka:

```

78
79     JButton btn7 = new JButton("7");
80     btn7.setBounds(10, 129, 58, 39);
81     btn7.addActionListener(new ActionListener() {
82         public void actionPerformed(ActionEvent e) {
83             String number=textField.getText()+btn7.getText();
84             textField.setText(number);
85         }
86     });
87 
```

Gambar 2.14 Potongan source code

Penjelasan:

- Menangani event tombol ditekan
- Nilai tombol ditambahkan ke layar (textField)

5. Tombol Operator (+, -, , /, %)

Tombol operator menyimpan nilai pertama dan jenis operasi.

Contoh tombol penjumlahan:

```

137     JButton btnPlus = new JButton("+");
138     btnPlus.addActionListener(new ActionListener() {
139         public void actionPerformed(ActionEvent e) {
140             first=Double.parseDouble(textField.getText());
141             textField.setText("");
142             operation="+";
143         }
144     });
145 
```

Gambar 2.15 potongan source code

Penjelasan:

- Mengambil nilai input sebagai first
- Membersihkan layer
- Menyimpan operator yang dipilih

6. Tombol Backspace

Menghapus satu karakter dari belakang.

```

62     JButton btnBackspace = new JButton("\u00d7");
63     btnBackspace.addActionListener(new ActionListener() {
64         public void actionPerformed(ActionEvent e) {
65             String backSpace=null;
66             if(textField.getText().length()>0)
67             {
68                 StringBuilder str=new StringBuilder(textField.getText());
69                 str.deleteCharAt(textField.getText().length()-1);
70                 backSpace=str.toString();
71                 textField.setText(backSpace);
72             }
73         }
74     });
    
```

Gambar 2.16 Potongan source code

Penjelasan:

- StringBuilder untuk memanipulasi teks
- Menghilangkan satu karakter terakhir

7. Tombol Clear (C)

```

127     JButton btnClear = new JButton("C");
128     btnClear.addActionListener(new ActionListener() {
129         public void actionPerformed(ActionEvent e) {
130             textField.setText(null);
131         }
132     });
    
```

Gambar 2.17 Potongan source code

Penjelasan:

- Membersihkan layar input secara penuh

8. Tombol Sama Dengan (=)

Bagian terpenting yang memproses operasi matematika.

```

209
210     JButton btnEqual = new JButton("=");
211     btnEqual.addActionListener(new ActionListener() {
212         public void actionPerformed(ActionEvent e) {
213             String answer;
214             second=Double.parseDouble(textField.getText());
215             if(operation=="+")
216             {
217                 result=first+second;
218                 answer=String.format("%.2f" , result);
219                 textField.setText(answer);
220             }
221             else if(operation=="-")
222             {
223                 result=first-second;
224                 answer=String.format("%.2f" , result);
225                 textField.setText(answer);
226             }
227             else if(operation=="*")
228             {
229                 result=first*second;
230                 answer=String.format("%.2f" , result);
231                 textField.setText(answer);
232             }
233             else if(operation=="/")
234             {
235                 result=first/second;
236                 answer=String.format("%.2f" , result);
237                 textField.setText(answer);
238             }
239         }
240     }
241 }

```

Gambar 2.18 Potongan source code

Penjelasan:

- Mengambil nilai kedua (second)
- Melakukan proses sesuai operator yang dipilih
- Menampilkan hasil ke textField
- Hasil diformat menjadi 2 angka di belakang koma

9. Event Handling (Inti dari Program GUI)

Semua tombol menggunakan konsep event-driven programming:

```

63     btnBackspace.addActionListener(new ActionListener() {
64         public void actionPerformed(ActionEvent e) {

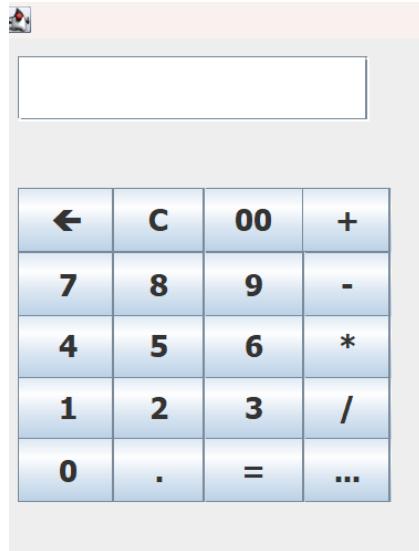
```

Gambar 2.19 Potongan source code

Penjelasan:

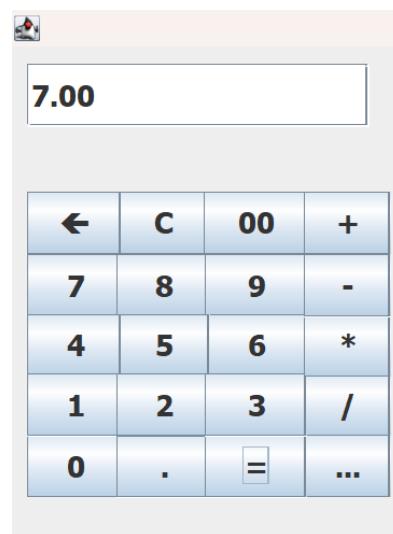
- Program berjalan berdasarkan aksi pengguna
- Event akan memicu fungsi tertentu

5. Output Program



Gambar 2.20 Output kode program calkulatorGUI

- Uji setiap fitur
- Langkah-langkah:
 1. Penjumlahan: Masukkan “5” → “+” → “2” → “=”
→ hasil “7.00”.



Gambar 2.21 Output dari percobaan penjumlahan

2. Pengurangan: “50” → “-” → “17.5” → “=” → “32.50”.
3. Perkalian: “10” → “*” → “6” → “=” → “60.00”.
4. Pembagian: “8” → “/” → “2” → “=” → “4.00”.
5. Modulus: “10” → “%” → “3” → “=” → “1.00”.

6. Backspace: Ketik “123” → Backspace → “12”.
7. Clear: Ketik angka → “C” → kosong.

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

BAB III

PENUTUP

3.1 Kesimpulan

Praktikum pembuatan kalkulator GUI menggunakan bahasa pemrograman Java berhasil dilakukan dengan baik. Program yang dibuat mampu menampilkan antarmuka grafis yang interaktif dan fungsional, meliputi tombol angka, operator aritmatika dasar, serta fungsi tambahan seperti *clear* dan *backspace*.

Implementasi *event handling* pada setiap tombol memungkinkan pengguna untuk memasukkan angka, memilih operasi, dan mendapatkan hasil perhitungan secara *real-time*. Selain itu, penggunaan komponen *Swing* seperti *JFrame*, *JTextField*, dan *JButton* membuktikan kemudahan dalam membangun aplikasi berbasis GUI di Java. Melalui praktikum ini, penulis memperoleh pemahaman lebih mendalam tentang pengelolaan *event*, manipulasi tampilan GUI, dan penerapan logika dasar kalkulator dalam pemrograman berorientasi objek.

Dengan demikian, kalkulator GUI ini tidak hanya sebagai sarana latihan pemrograman Java, tetapi juga sebagai dasar pengembangan aplikasi GUI yang lebih kompleks di masa depan.

DAFTAR PUSTAKA

- [1] D. J. Eck, *Introduction to Programming Using Java, Version 7.3*. [Online]. Available: <http://math.hws.edu/javanotes/>. [Accessed: Nov. 29, 2025].
- [2] Oracle, *Java Platform, Standard Edition - Swing Tutorial*, 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/>. [Accessed: Nov. 29, 2025].
- [3] GeeksforGeeks, "Java Swing | Simple Calculator." [Online]. Available: https://www.geeksforgeeks.org/java-swing-simple-calculator/?utm_source=chatgpt.com. [Accessed: Nov. 29, 2025].
- [4] Java Code Geeks, "Java Calculator using Swing Example." [Online]. Available: https://examples.javacodegeeks.com/java-calculator-using-swing-example/?utm_source=chatgpt.com. [Accessed: Nov. 29, 2025].