

LAPORAN PRAKTIKUM  
PEMOGRAMAN ALGORITMA PEMOGRAMAN  
PEMOGRAMAN GUI 2 PADA JAVA



OLEH:  
**LEXI MULIA YUNASPI**  
**(2511531006)**

DOSEN PENGAMPU:  
**DR. WAHYUDI, S.T, M.T**

ASISTEN PRAKTIKUM :  
**MUHAMMAD ZAKI AL HAFIZ**

FAKULTAS TEKNOLOGI INFORMASI  
DEPARTEMEN INFORMATIKA  
UNIVERSITAS ANDALAS  
PADANG  
2025

## **DAFTAR ISI**

DAFTAR ISI .....	i
KATA PENGANTAR .....	.ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang .....	1
1.2 Tujuan Praktikum .....	1
1.3 Manfaat Praktikum .....	2
BAB II PEMBAHASAN.....	3
2.1 Praktikum “Kalkulator GUI” .....	12
BAB III KESIMPULAN .....	13
3.1 Kesimpulan .....	14
DAFTAR PUSTAKA .....	14

## KATA PENGANTAR

Laporan praktikum ini disusun sebagai bentuk pertanggungjawaban atas pelaksanaan kegiatan praktikum mata kuliah Algoritma Pemrograman, khususnya pada materi mengenai Pemrograman GUI (Graphical User Interface) di Java. Penyusunan laporan ini tidak hanya bertujuan untuk memenuhi kewajiban akademik, tetapi juga sebagai sarana bagi penulis untuk memperdalam pemahaman terhadap konsep-konsep dasar pemrograman berbasis antarmuka grafis yang telah dipelajari selama kegiatan praktikum. Melalui proses penyusunan laporan ini, penulis memperoleh kesempatan untuk berlatih menyusun dokumentasi ilmiah secara sistematis, melatih ketelitian dalam melakukan pengamatan, serta meningkatkan kemampuan dalam menuangkan ide dan hasil praktikum ke dalam bentuk tulisan yang sesuai dengan kaidah penulisan akademik tingkat dasar. Selain itu, laporan ini diharapkan dapat menjadi media pembelajaran yang membantu penulis maupun pembaca dalam memahami langkah kerja, logika pemrograman, serta implementasi komponen GUI dalam bahasa Java.

Penulis menyadari sepenuhnya bahwa laporan ini masih jauh dari kata sempurna, baik dari segi isi, teknik penulisan, maupun penyajian materi. Oleh karena itu, penulis sangat terbuka dan mengharapkan segala bentuk kritik serta saran yang bersifat membangun, guna memperbaiki kualitas laporan praktikum di masa mendatang sekaligus meningkatkan pemahaman penulis terhadap materi yang dipelajari. Akhir kata, penulis menyampaikan terima kasih kepada dosen pengampu, asisten praktikum, dan semua pihak yang telah memberikan bimbingan, dukungan, serta fasilitas sehingga kegiatan praktikum dan penyusunan laporan ini dapat terlaksana dengan baik. Semoga laporan ini dapat memberikan manfaat bagi siapa saja yang membacanya dan menjadi referensi sederhana dalam mempelajari pemrograman GUI di Java.

Padang, 2025

Penulis

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Praktikum ini membahas pembuatan aplikasi kalkulator sederhana menggunakan Java dan Swing sebagai sarana untuk memahami konsep dasar GUI dan logika pemrograman. Melalui praktikum ini, dipelajari penggunaan komponen seperti JFrame, JTextField, dan JButton, serta penerapan event-driven programming dengan ActionListener. Pengembangan kalkulator ini juga melatih pengelolaan state aplikasi dan pemrosesan operasi aritmatika dasar secara interaktif.

#### **1.2 Tujuan Praktikum**

1. Memahami dasar pembuatan GUI menggunakan Java Swing.
2. Mampu menggunakan komponen GUI seperti JFrame, JTextField, dan JButton.
3. Mempraktikkan penanganan event dengan ActionListener.
4. Mengembangkan logika operasi aritmatika beserta fitur tambahan seperti persen dan backs

#### **1.3 Manfaat Praktikum**

1. Memahami cara kerja aplikasi berbasis GUI di Java.
2. Mampu membuat antarmuka yang mudah digunakan.
3. Terbiasa menangani interaksi pengguna.
4. Mampu menerapkan logika perhitungan dan menampilkan hasil secara langsung seperti kalkulator nyata.

## BAB II

### PEMBAHASAN

#### 2.1 Praktikum “KalkulatorGUI”

```
1 package Pekan9_2511531006;
2
3
4@import java.awt.EventQueue;[]
14
15 public class Kalkulator_2511531006 extends JFrame {
16
17     private static final long serialVersionUID = 1L;
18     private JPanel contentPane;
19     private JTextField textField;
20
21     double first;
22     double second;
23     double results;
24     String operation;
25     String answer;
26
27@    /**
28     * Launch the application.
29     */
30@    public static void main(String[] args) {
31        EventQueue.invokeLater(new Runnable() {
32            public void run() {
33                try {
34                    Kalkulator_2511531006 frame = new Kalkulator_2511531006();
35                    frame.setVisible(true);
36                } catch (Exception e) {
37                    e.printStackTrace();
38                }
39            }
40        });
41    }
42
43@    /**
44     * Create the frame.
45     */
46@    public Kalkulator_2511531006() {
47        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
48        setBounds(100, 100, 370, 400);
49        contentPane = new JPanel();
50        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
51        setContentPane(contentPane);
52        contentPane.setLayout(null);
53
54        textField = new JTextField();
55        textField.setFont(new Font("Tahoma", Font.BOLD, 18));
56        textField.setBounds(10, 11, 334, 60);
57        contentPane.add(textField);
58        textField.setColumns(10);
59
60        JButton btnBackspace = new JButton("\u00d7");
61        btnBackspace.addActionListener(new ActionListener() {
62            public void actionPerformed(ActionEvent e) {
63                String backspace=null;
64@                if(textField.getText().length()>0)
65                {
66                    StringBuilder str=new StringBuilder(textField.getText());
67                    str.deleteCharAt(textField.getText().length()-1);
68                    backspace=str.toString();
69                    textField.setText(backspace);
70                }
71            }
72        });
73        btnBackspace.setFont(new Font("Wingdings", Font.BOLD, 14));
74        btnBackspace.setBounds(10, 89, 82, 52);
75        contentPane.add(btnBackspace);
76
77        JButton btn7 = new JButton("7");
78@        btn7.addActionListener(new ActionListener() {
79            public void actionPerformed(ActionEvent e) {
80                String number =textField.getText()+btn7.getText();
81                textField.setText(number);
82            }
83        });
84        btn7.setFont(new Font("Tahoma", Font.BOLD, 14));
85        btn7.setBounds(10, 141, 82, 52);
86        contentPane.add(btn7);
87
88        JButton btn4 = new JButton("4");
89        btn4.addActionListener(new ActionListener() {
90            public void actionPerformed(ActionEvent e) {
91                String number =textField.getText()+btn4.getText();
92                textField.setText(number);
93            }
94        });
95        btn4.setFont(new Font("Tahoma", Font.BOLD, 14));
96        btn4.setBounds(10, 193, 82, 52);
97        contentPane.add(btn4);
98
99        JButton btn1 = new JButton("1");
100        btn1.addActionListener(new ActionListener() {
101            public void actionPerformed(ActionEvent e) {
102                String number =textField.getText()+btn1.getText();
103                textField.setText(number);
104            }
105        });
106        btn1.setFont(new Font("Tahoma", Font.BOLD, 14));
107        btn1.setBounds(10, 245, 82, 52);
108        contentPane.add(btn1);
109
110        JButton btn0 = new JButton("0");
111        btn0.addActionListener(new ActionListener() {
```

```
112@        public void actionPerformed(ActionEvent e) {
113            String number =textField.getText()+btn0.getText();
114            textField.setText(number);
115        }
116    });
117    btn0.setFont(new Font("Tahoma", Font.BOLD, 14));
118    btn0.setBounds(10, 298, 82, 52);
119    contentPane.add(btn0);
120
121    JButton btnPlus = new JButton("+");
122    btnPlus.addActionListener(new ActionListener() {
123        public void actionPerformed(ActionEvent e) {
124            first=Double.parseDouble(textField.getText());
125            textField.setText("");
126            operation="+";
127        }
128    });
129    btnPlus.setFont(new Font("Tahoma", Font.BOLD, 14));
130    btnPlus.setBounds(262, 89, 82, 52);
131    contentPane.add(btnPlus);
132
133    JButton btn00 = new JButton("00");
134    btn00.addActionListener(new ActionListener() {
135        public void actionPerformed(ActionEvent e) {
136            String number =textField.getText()+btn00.getText();
137            textField.setText(number);
138        }
139    });
140    btn00.setFont(new Font("Tahoma", Font.BOLD, 14));
141    btn00.setBounds(176, 89, 82, 52);
142    contentPane.add(btn00);
143
144    JButton btnClear = new JButton("C");
145    btnClear.addActionListener(new ActionListener() {
```

```

146●    public void actionPerformed(ActionEvent e) {
147        textField.setText(null);
148    }
149 });
150 btnClear.setFont(new Font("Tahoma", Font.BOLD, 14));
151 btnClear.setBounds(95, 89, 82, 52);
152 contentPane.add(btnClear);
153
154 JButton btn8 = new JButton("8");
155● btn8.addActionListener(new ActionListener() {
156●     public void actionPerformed(ActionEvent e) {
157         String number =textField.getText()+btn8.getText();
158         textField.setText(number);
159     }
160 });
161 btn8.setFont(new Font("Tahoma", Font.BOLD, 14));
162 btn8.setBounds(95, 141, 82, 52);
163 contentPane.add(btn8);
164
165 JButton btn5 = new JButton("5");
166● btn5.addActionListener(new ActionListener() {
167●     public void actionPerformed(ActionEvent e) {
168         String number =textField.getText()+btn5.getText();
169         textField.setText(number);
170     }
171 });
172 btn5.setFont(new Font("Tahoma", Font.BOLD, 14));
173 btn5.setBounds(95, 193, 82, 52);
174 contentPane.add(btn5);
175
176 JButton btn2 = new JButton("2");
177● btn2.addActionListener(new ActionListener() {
178●     public void actionPerformed(ActionEvent e) {
179         String number =textField.getText()+btn2.getText();
180
181         textField.setText(number);
182     }
183 });
184 btn2.setFont(new Font("Tahoma", Font.BOLD, 14));
185 btn2.setBounds(95, 245, 82, 52);
186 contentPane.add(btn2);
187
188 JButton btnDot = new JButton(".");
189● btnDot.addActionListener(new ActionListener() {
190●     public void actionPerformed(ActionEvent e) {
191         String number =textField.getText()+btnDot.getText();
192         textField.setText(number);
193     }
194 });
195 btnDot.setFont(new Font("Tahoma", Font.BOLD, 14));
196 btnDot.setBounds(95, 298, 82, 52);
197 contentPane.add(btnDot);
198
199 JButton btn9 = new JButton("9");
200● btn9.addActionListener(new ActionListener() {
201●     public void actionPerformed(ActionEvent e) {
202         String number =textField.getText()+btn9.getText();
203         textField.setText(number);
204     }
205 });
206 btn9.setFont(new Font("Tahoma", Font.BOLD, 14));
207 btn9.setBounds(176, 141, 82, 52);
208 contentPane.add(btn9);
209
210 JButton btnMin = new JButton("-");
211● btnMin.addActionListener(new ActionListener() {
212●     public void actionPerformed(ActionEvent e) {
213         firstDouble.parseDouble(textField.getText());
214         textField.setText("");
215         operation="-";
216     }
217 });
218 btnMin.setFont(new Font("Tahoma", Font.BOLD, 14));
219 btnMin.setBounds(262, 141, 82, 52);
220 contentPane.add(btnMin);
221
222 JButton btnMul = new JButton("*");
223● btnMul.addActionListener(new ActionListener() {
224●     public void actionPerformed(ActionEvent e) {
225         firstDouble.parseDouble(textField.getText());
226         textField.setText("");
227         operation="*";
228     }
229 });
230 btnMul.setFont(new Font("Tahoma", Font.BOLD, 14));
231 btnMul.setBounds(262, 193, 82, 52);
232 contentPane.add(btnMul);
233
234 JButton btn6 = new JButton("6");
235● btn6.addActionListener(new ActionListener() {
236●     public void actionPerformed(ActionEvent e) {
237         String number =textField.getText()+btn6.getText();
238         textField.setText(number);
239     }
240 });
241 btn6.setFont(new Font("Tahoma", Font.BOLD, 14));
242 btn6.setBounds(176, 193, 82, 52);
243 contentPane.add(btn6);
244
245 JButton btn3 = new JButton("3");
246● btn3.addActionListener(new ActionListener() {
247●     public void actionPerformed(ActionEvent e) {
248         String number =textField.getText()+btn3.getText();
249
250         textField.setText(number);
251     }
252 });
253 btn3.setFont(new Font("Tahoma", Font.BOLD, 14));
254 btn3.setBounds(176, 245, 82, 52);
255 contentPane.add(btn3);
256
257 JButton btnDivide = new JButton("/");
258● btnDivide.addActionListener(new ActionListener() {
259●     public void actionPerformed(ActionEvent e) {
260         firstDouble.parseDouble(textField.getText());
261         textField.setText("");
262         operation="/";
263     }
264 });
265 btnDivide.setFont(new Font("Tahoma", Font.BOLD, 14));
266 btnDivide.setBounds(262, 245, 82, 52);
267 contentPane.add(btnDivide);
268
269 JButton btnEqual = new JButton "=";
270● btnEqual.addActionListener(new ActionListener() {
271●     public void actionPerformed(ActionEvent e) {
272         String answer;
273         secondDouble.parseDouble(textField.getText());
274         if(operation=="+")
275         {
276             results=first + second;
277             answer=String.format("%.2f", results);
278             textField.setText(answer);
279         }
280         else if(operation=="-")
281         {
282             results=first - second;
283             answer=String.format("%.2f", results);
284         }
285     }
286 });
287 btnEqual.setFont(new Font("Tahoma", Font.BOLD, 14));
288 btnEqual.setBounds(262, 298, 82, 52);
289 contentPane.add(btnEqual);

```

```

282         textField.setText(answer);
283     }
284     else if(operation=="*")
285     {
286         results=first * second;
287         answer=String.format("%.2f", results);
288         textField.setText(answer);
289     }
290     else if(operation=="/")
291     {
292         results=first / second;
293         answer=String.format("%.2f", results);
294         textField.setText(answer);
295     }
296     else if(operation=="%")
297     {
298         results=first % second;
299         answer=String.format("%.2f", results);
300         textField.setText(answer);
301     }
302
303
304
305     });
306     btnEqual.setFont(new Font("Tahoma", Font.BOLD, 14));
307     btnEqual.setBounds(176, 298, 82, 52);
308     contentPane.add(btnEqual);
309
310     JButton btnPercent = new JButton("%");
311     btnPercent.addActionListener(new ActionListener() {
312         public void actionPerformed(ActionEvent e) {
313             first=Double.parseDouble(textField.getText());
314             textField.setText("");
315
316             operation="%";
317         }
318     });
319     btnPercent.setFont(new Font("Tahoma", Font.BOLD, 14));
320     btnPercent.setBounds(262, 298, 82, 52);
321     contentPane.add(btnPercent);
322
323 }
324 }
```

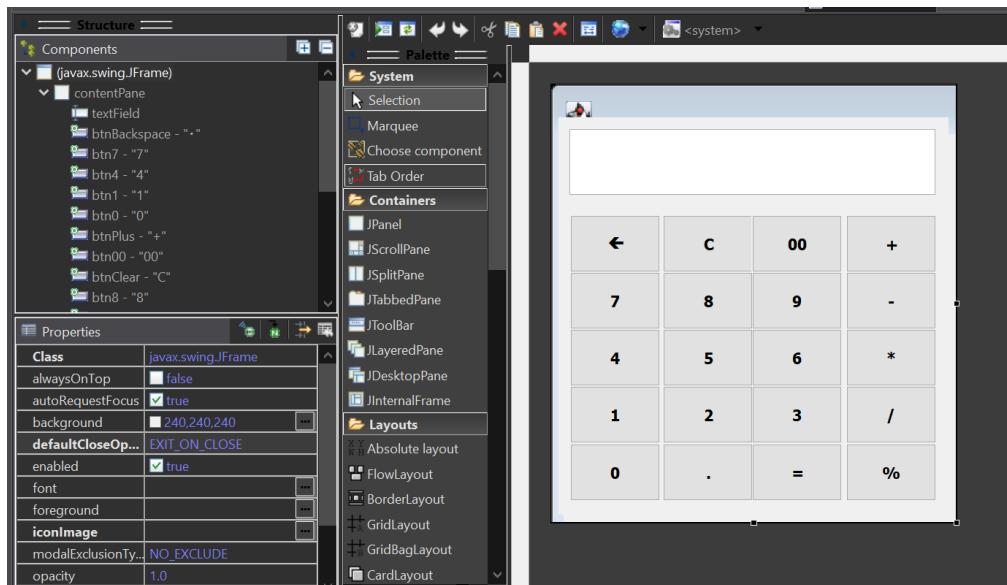
Gambar 2.1 Kode Program Praktikum “KalkulatorGUI”

Program ini dikembangkan menggunakan WindowBuilder, yaitu sebuah alat bantu visual yang memudahkan perancangan antarmuka dalam pemrograman Java, khususnya ketika bekerja dengan library Java Swing. Dengan memanfaatkan fitur *drag-and-drop* yang dimiliki WindowBuilder, proses pembuatan tampilan GUI dapat dilakukan dengan lebih cepat, terstruktur, dan presisi. Pada praktikum ini, tujuan utama pengembangan adalah merancang sebuah aplikasi kalkulator sederhana yang mampu melakukan operasi aritmetika dasar, yaitu penjumlahan, pengurangan, perkalian, dan pembagian. Desain antarmuka aplikasi dirancang sedemikian rupa agar mudah dipahami dan digunakan oleh

pengguna. Pengguna dapat memasukkan dua bilangan melalui komponen input teks (*JTextField*), kemudian memilih operator yang diinginkan—baik melalui tombol operasi maupun komponen pilihan lain yang telah disediakan. Setelah itu, pengguna cukup menekan tombol “Proses” untuk menjalankan perhitungan sesuai operasi yang dipilih. Hasil perhitungan akan ditampilkan secara otomatis pada area keluaran, sehingga pengguna dapat melihat hasilnya secara langsung.

Selain menampilkan hasil perhitungan, aplikasi ini juga dilengkapi dengan mekanisme validasi input untuk menjaga kestabilan dan keakuratan proses. Validasi tersebut mencakup pemeriksaan terhadap teks kosong, karakter non-numerik, serta kondisi khusus seperti pembagian dengan nol yang tidak diperbolehkan dalam operasi matematika. Jika ditemukan adanya kesalahan input, sistem akan memberikan pesan peringatan berupa dialog notifikasi sehingga pengguna dapat segera memperbaiki data yang dimasukkan. Dengan demikian, aplikasi ini tidak hanya berfungsi sebagai alat perhitungan, tetapi juga sebagai sarana pembelajaran mengenai pentingnya penanganan error dan validasi data dalam pengembangan perangkat lunak berbasis GUI.

Berikut adalah penjelasan mengenai bentuk desain kalkulator :



Gambar 2.2 Design Kalkulator

Berikut penjelasan layar output dan panel tombol secara jelas, rapi, dan cocok untuk dimasukkan ke laporan:

## **1. Layar Output (Display)**

Layar output pada kalkulator ditampilkan menggunakan komponen JTextField yang ditempatkan di bagian paling atas antarmuka.

A. Fungsi Utama Layar Output:

- Menampilkan angka yang diketik pengguna saat tombol angka ditekan.
- Menampilkan hasil operasi perhitungan, seperti penjumlahan, pengurangan, perkalian, atau pembagian.
- Menampilkan pesan khusus, misalnya ketika input tidak valid atau terjadi kesalahan (misalnya pembagian dengan nol).

B. Karakteristik Desain:

- Ukuran lebar, sehingga memudahkan pengguna membaca angka atau hasil.
- Latar berwarna putih, memberikan kontras tinggi dengan area tombol.
- Tanpa border tebal, sehingga tampilannya sederhana dan bersih.
- Dibuat satu baris sehingga menyerupai layar kalkulator fisik.
- Dapat menampilkan angka panjang berkat properti yang fleksibel.

Layar ini berfungsi sebagai pusat informasi selama proses perhitungan berlangsung.

## **2. Panel Tombol (Buttons Panel)**

Panel tombol merupakan area khusus yang berisi seluruh tombol angka dan operasi.

Tombol-tombol ini disusun dalam layout seperti grid sehingga tampak rapi dan mudah digunakan.

Susunan Tombol: Tombol disusun dalam 5 baris dan 4 kolom, yaitu: +, -, \*, /, %

Fungsi Tiap Kelompok Tombol:

- Tombol khusus:
  - *Backspace* ( $\leftarrow$ ) : menghapus satu karakter terakhir di layar.
  - *C (Clear)* : menghapus seluruh input dan mereset kalkulator.
  - *00* : menambahkan angka “00” dengan cepat.

- Tombol Angka (0–9):  
Digunakan untuk memasukkan bilangan.
- \*Tombol Operator (+, -, \*, /):  
Dipakai untuk memilih jenis operasi perhitungan.
- Tombol Lainnya:
  - “.” untuk angka desimal
  - “=” untuk menampilkan hasil perhitungan
  - “%” untuk menghitung persen

#### C.Karakteristik Desain Panel Tombol:

- Semua tombol memiliki ukuran sama sehingga tampilan seragam dan nyaman dilihat.
- Tombol menggunakan warna abu-abu muda, memberikan visual yang lembut dan profesional.
- Penempatan tombol operator berada di kolom kanan agar mudah diakses.
- Desain mengikuti bentuk kalkulator fisik sehingga pengguna dengan cepat mengenali fungsi tiap tombol.

### Penjelasan Logika Program Kalkulator

#### 1. Deklarasi Variabel Utama

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

Gambar 2.3 Deklarasi Variabel Utama

Penjelasan:

- first → menyimpan angka pertama sebelum operasi.
- second → menyimpan angka kedua setelah operator dipilih.
- results → menampung hasil perhitungan.
- operation → menyimpan jenis operator (+, -, \*, /, %).

answer → dipakai untuk memformat hasil sebelum ditampilkan.

#### 2. Konstruktor Frame dan Pengaturan GUI

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setBounds(100, 100, 370, 400);
```

Gambar 2.4 Konstruktor Frame dan Pengaturan GUI

Menentukan bahwa aplikasi akan tertutup ketika jendela ditutup. Dan mengatur ukuran dan posisi awal jendela kalkulator.

### 3. Komponen Layar Output (textField)

```
textField = new JTextField();
textField.setFont(new Font("Tahoma", Font.BOLD, 18));
```

Gambar 2.5 Komponen Layar Output (textField)

Membuat layar kalkulator untuk menampilkan input dan hasil. Dan memberikan font besar agar mudah dibaca.

### 4. Logika Tombol Backspace (←)

```
StringBuilder str=new StringBuilder(textField.getText());
str.deleteCharAt(textField.getText().length()-1);
textField.setText(str.toString());
```

Gambar 2.6 Logika Tombol Backspace (←)

Mengambil teks di layar, menghapus satu karakter terakhir

### 5. Logika Tombol Angka (0–9 dan 00)

```
String number =textField.getText()+btn7.getText();
textField.setText(number);
```

Gambar 2.7 Logika Tombol Angka (0–9 dan 00)

Saat sebuah tombol angka ditekan, program terlebih dahulu mengambil input sementara yang sudah ada pada layar kalkulator. Setelah itu, angka yang berasal dari tombol yang ditekan akan ditambahkan ke akhir teks tersebut sehingga membentuk nilai baru. Nilai yang sudah diperbarui ini kemudian ditampilkan kembali pada layar kalkulator sehingga pengguna dapat melihat angka terbaru yang sedang diketik.

### 6. Tombol Operator (+, -, /, %)

Contoh operator:

```
first=Double.parseDouble(textField.getText());
textField.setText("");
operation="+";
```

Gambar 2.8 Tombol Operator

Ketika pengguna menekan salah satu tombol operator, program akan mengambil nilai pertama yang sedang tampil di layar kemudian mengonversinya menjadi tipe data *double* agar dapat diproses dalam perhitungan. Setelah nilai tersebut disimpan, layar kalkulator dikosongkan untuk memberi ruang bagi pengguna dalam memasukkan angka kedua. Pada saat yang sama, operator yang dipilih juga disimpan ke dalam variabel *operation* sebagai penanda jenis operasi yang akan dijalankan ketika proses perhitungan dilakukan.

#### 7. Tombol Clear (C)

```
textField.setText(null);
```

Gambar 2.9 Tombol Clear

Menghapus seluruh input di layar berarti program membersihkan atau mereset nilai yang sedang tampil pada area tampilan, biasanya pada *text field* atau layar kalkulator. Ketika perintah ini dijalankan, semua angka atau karakter yang sebelumnya dimasukkan oleh pengguna akan dihapus sehingga layar kembali kosong. Fungsi ini digunakan agar pengguna dapat memulai perhitungan baru tanpa terganggu oleh data sebelumnya. Biasanya dilakukan dengan memanggil metode seperti *setText("")*, yang mengubah isi layar menjadi string kosong. Dengan demikian, tampilan kembali bersih dan siap menerima input baru.

#### 8. Tombol Sama Dengan (=)

```
second=Double.parseDouble(textField.getText());
if(operation=="=")
```

Gambar 2.10 Tombol Sama Dengan

Inilah inti logika perhitungan. Detail logikanya:

Mengambil nilai kedua dari layar. Dan mengecek operator yang disimpan sebelumnya:

- Jika "+" → lakukan first + second
- Jika "-" → lakukan first - second
- Jika "\*" → lakukan first \* second
- Jika "/" → lakukan first / second
- Jika "%" → lakukan first % second

Kesimpulan Singkat Logika Program :

- Tombol angka → menulis angka ke layar.
- Tombol operator → menyimpan angka pertama dan operator.
- Tombol "=" → membaca angka kedua, menghitung, lalu menampilkan hasil.
- Tombol "C" → menghapus layar.
- Tombol backspace → menghapus satu karakter terakhir.
- Semua hasil diformat menjadi dua angka desimal.

Berikut adalah contoh output program kalkulator :



### Gambar 2.11 Output Kode Program Kalkulator

Berikut penjelasan proses “9 kali 2” pada kalkulator seperti yang ada pada gambar:

Ketika pengguna menekan tombol 9, angka 9 akan muncul pada layar sebagai input pertama. Setelah itu tombol operasi kali ( $\times$ ) ditekan, sehingga program menyimpan nilai pertama (9) ke dalam variabel sementara, lalu menunggu input berikutnya. Kemudian pengguna menekan angka 2, sehingga layar menampilkan angka 2 sebagai nilai kedua. Ketika tombol = ditekan, program melakukan perhitungan sesuai operator yang dipilih, yaitu perkalian. Maka kalkulator menghitung  $9 \times 2 = 18$ . Hasil akhir tersebut kemudian ditampilkan pada layar, dalam hal ini muncul sebagai 18,00 sesuai format tampilan kalkulator yang menggunakan dua angka desimal.

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Praktikum ini berhasil menghasilkan sebuah aplikasi kalkulator sederhana berbasis GUI menggunakan Java Swing. Melalui proses perancangan dan implementasi program, mahasiswa dapat memahami bagaimana elemen-elemen antarmuka grafis—seperti tombol, label, dan text field—dapat digabungkan dengan logika pemrograman untuk menciptakan aplikasi yang interaktif dan responsif. Pembuatan kalkulator ini memberikan gambaran nyata mengenai cara menangani input pengguna, menampilkan output secara dinamis, serta mengelola data sementara seperti angka pertama, angka kedua, dan operator yang dipilih. Selain itu, praktikum ini memperkuat pemahaman terhadap konsep event-driven programming, khususnya melalui penggunaan *ActionListener* pada setiap tombol. Setiap aksi yang dilakukan pengguna, seperti menekan tombol angka atau operator, memicu logika tertentu yang mengubah tampilan atau melakukan perhitungan. Hal ini menunjukkan bagaimana alur program dalam aplikasi GUI tidak berjalan secara linear, melainkan berdasarkan event yang dipicu oleh interaksi pengguna.

Dari sisi antarmuka, desain kalkulator dibuat sederhana dan menyerupai kalkulator pada umumnya sehingga mudah dipahami dan digunakan. Proses pengujian aplikasi juga memberikan pengalaman dalam memastikan bahwa setiap tombol berfungsi dengan benar dan semua operasi aritmetika dasar—seperti penjumlahan, pengurangan, perkalian, dan pembagian—dapat dijalankan tanpa error. Secara keseluruhan, praktikum ini memberikan pengalaman menyeluruh dalam membangun aplikasi desktop sederhana, mulai dari perancangan antarmuka, pengkodean logika program, hingga pengujian fitur.

## **DAFTAR PUSTAKA**

M. Suyanto, Pengantar Pemrograman Java untuk Pemula. Yogyakarta, Indonesia: Andi Publisher, 2020.

Oracle, “The Java™ Tutorials: Creating a GUI With JFC/Swing,” Oracle Documentation,2025.[Online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/> [Accessed: Nov. 27, 2025].

[ Eclipse Foundation, “WindowBuilder User Guide,” Eclipse IDE Documentation, 2025. [Online]. Available: <https://www.eclipse.org/windowbuilder/> [Accessed: Nov. 27, 2025].