

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMOGRAMAN
CALCULATOR GUI
PEKAN 9

Disusun Oleh :

KINAYA NOVRYA MANDA
(2511531016)

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

MUHAMMAD ZAKI AL HAFIZ



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis sampaikan kehadirat Allah SWT. Salawat dan salam disampaikan kepada Nabi Muhammad SAW. Karena thaufik dan hidayah-Nya, laporan praktikum Java ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dari kegiatan praktikum yang telah dilaksanakan, sekaligus sebagai sarana untuk memperdalam pemahaman mengenai konsep dasar serta penerapan bahasa pemrograman Java.

Melalui praktikum ini, penulis memperoleh pengalaman langsung dalam memahami hubungan antara logika perhitungan dan desain antarmuka yang saling mendukung menggunakan Java. Diharapkan laporan ini dapat memberikan gambaran yang jelas mengenai materi yang telah dipelajari serta hasil dari percobaan yang dilakukan selama praktikum berlangsung.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa yang akan datang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta semua pihak yang telah memberikan bimbingan, arahan, dan dukungan sehingga laporan praktikum ini dapat tersusun. Semoga laporan ini dapat bermanfaat bagi pembaca, khususnya dalam memahami dasar-dasar pemrograman Java.

Padang, November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR LAMPIRAN.....	iii
BAB 1 PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	2
1.3 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN	
2.1 Kode Program.....	3
2.2 Langkah Kerja.....	8
2.3 Analisis Hasil.....	16
BAB 3 KESIMPULAN	
3.1 Hasil Praktikum.....	17
3.2 Saran Pengembangan.....	18
DAFTAR KEPUSTAKAAN.....	19

DAFTAR LAMPIRAN

Kode Program 2.1 (Kode Program CalculatorGUI).....	3
Gambar 2.1 (Langkah 1 CalculatorGUI).....	8
Gambar 2.2 (Langkah 2 CalculatorGUI).....	8
Gambar 2.3 (Langkah 3 CalculatorGUI).....	9
Gambar 2.4 (Langkah 4 CalculatorGUI).....	9
Gambar 2.5 (Langkah 5 CalculatorGUI).....	10
Gambar 2.6 (Langkah 6 CalculatorGUI).....	10
Gambar 2.7 (Langkah 7 CalculatorGUI).....	11
Gambar 2.8 (Langkah 8 CalculatorGUI).....	11
Gambar 2.9 (Langkah 9 CalculatorGUI).....	12
Gambar 2.10 (Langkah 10 CalculatorGUI).....	12
Gambar 2.11 (Langkah 11 CalculatorGUI).....	13
Gambar 2.12 (Langkah 12 CalculatorGUI).....	13
Gambar 2.13 (Langkah 13 CalculatorGUI).....	14
Gambar 2.14 (Langkah 14 CalculatorGUI).....	14
Gambar 2.15 (Langkah 15 CalculatorGUI).....	15
Gambar 2.16 (Langkah 16 CalculatorGUI).....	15
Gambar 3.1 (Hasil Praktikum CalculatorGUI).....	17

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pembelajaran pemrograman Java, mahasiswa tidak hanya perlu memahami sintaks dan logika dasar, tetapi juga harus mampu membuat aplikasi yang dapat berinteraksi langsung dengan pengguna. Salah satu bentuk aplikasi yang sering digunakan sebagai media latihan adalah kalkulator berbasis GUI (*Graphical User Interface*). Melalui tampilan GUI, pengguna dapat memasukkan angka dan memilih operasi dengan lebih mudah dibandingkan melalui *console*.

Pembuatan kalkulator GUI juga menjadi langkah awal bagi mahasiswa untuk mengenal komponen antarmuka seperti tombol, label, dan *text field*. Selain itu, praktikum ini membantu mahasiswa memahami bagaimana sebuah program merespons tindakan pengguna melalui *event handling*, misalnya saat tombol ditekan untuk melakukan perhitungan. Konsep ini sangat penting karena menjadi dasar dalam pengembangan aplikasi *desktop* yang interaktif.

Dengan adanya praktikum ini, mahasiswa dapat mengembangkan kemampuan dalam menggabungkan tampilan antarmuka dengan logika program. Hal ini akan menjadi bekal penting sebelum membuat aplikasi yang lebih kompleks di perkuliahan berikutnya. Praktikum ini juga melatih mahasiswa agar lebih kreatif dan terampil dalam membuat tampilan yang mudah dipahami dan digunakan.

1.2 Tujuan Praktikum

Tujuan dilakukannya praktikum ini adalah sebagai berikut :

1. Memahami konsep dasar pembuatan GUI pada Java.
2. Mengimplementasikan operasi aritmatika dasar ke dalam program kalkulator.
3. Melatih kemampuan menghubungkan komponen GUI dengan *event handler*.
4. Meningkatkan kemampuan mahasiswa dalam membuat aplikasi yang interaktif dan mudah digunakan.

1.3 Manfaat Praktikum

Manfaat dilakukannya praktikum ini adalah sebagai berikut :

1. Mahasiswa memperoleh pengalaman langsung dalam membuat aplikasi berbasis GUI.
2. Menambah pemahaman tentang penggunaan komponen visual dalam program.
3. Meningkatkan kemampuan dalam menangani input pengguna dan memproses data.
4. Melatih kreativitas dalam membuat tampilan program yang menarik dan mudah digunakan.

BAB II

PEMBAHASAN

2.1 Kode Program

```

1  package Pekan9_2511531016;
2
3  import java.awt.EventQueue;
11
12  public class Calculator_2511531016 {
13
14      private JFrame frame;
15      private JTextField textField;
16      private JButton btn7;
17      private JButton btn4;
18      private JButton btn1;
19      private JButton btn0;
20      private JButton btnClear;
21      private JButton btn8;
22      private JButton btn5;
23      private JButton btn2;
24      private JButton btnDot;
25      private JButton btn00;
26      private JButton btn9;
27      private JButton btn6;
28      private JButton btn3;
29      private JButton btnEqual;
30      private JButton btnPlus;
31      private JButton btnSub;
32      private JButton btnMul;
33      private JButton btnDivide;
34      private JButton btnPercent;
35
36      double first;
37      double second;
38      double result;
39      String operation;
40      String answer;
41
42      /**
43       * Launch the application.
44       */
45      public static void main(String[] args) {
46          EventQueue.invokeLater(new Runnable() {
47              public void run() {
48                  try {
49                      Calculator_2511531016 window = new Calculator_2511531016();
50                      window.frame.setVisible(true);
51                  } catch (Exception e) {
52                      e.printStackTrace();
53                  }
54              }
55          });
56      }
57
58      /**
59       * Create the application.
60       */
61      public Calculator_2511531016() {
62          initialize();
63      }
64
65      /**
66       * Initialize the contents of the frame.
67       */
68      private void initialize() {
69          frame = new JFrame();
70          frame.setBounds(100, 100, 301, 349);
71          frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
72          frame.getContentPane().setLayout(null);
73

```

```

74      textField = new JTextField();
75      textField.setFont(new Font("Tahoma", Font.BOLD, 18));
76      textField.setBounds(10, 11, 271, 56);
77      frame.getContentPane().add(textField);
78      textField.setColumns(10);
79
80      JButton btnBackspace = new JButton("\uF0E7");
81      btnBackspace.addActionListener(new ActionListener() {
82          public void actionPerformed(ActionEvent arg0) {
83              String backSpace=null;
84              if(textField.getText().length()>0)
85              {
86                  StringBuilder str=new StringBuilder(textField.getText());
87                  str.deleteCharAt(textField.getText().length()-1);
88                  backSpace=str.toString();
89                  textField.setText(backSpace);
90              }
91          }
92      });
93      btnBackspace.setFont(new Font("Windings", Font.BOLD, 18));
94      btnBackspace.setBounds(10, 78, 68, 44);
95      frame.getContentPane().add(btnBackspace);
96
97      btn7 = new JButton("7");
98      btn7.addActionListener(new ActionListener() {
99          public void actionPerformed(ActionEvent e) {
100              String number=textField.getText()+btn7.getText();
101              textField.setText(number);
102          }
103      });
104      btn7.setFont(new Font("Tahoma", Font.BOLD, 18));
105      btn7.setBounds(10, 122, 68, 44);
106      frame.getContentPane().add(btn7);
107

```

```

108      btn4 = new JButton("4");
109      btn4.addActionListener(new ActionListener() {
110          public void actionPerformed(ActionEvent e) {
111              String number=textField.getText()+btn4.getText();
112              textField.setText(number);
113          }
114      });
115      btn4.setFont(new Font("Tahoma", Font.BOLD, 18));
116      btn4.setBounds(10, 167, 68, 44);
117      frame.getContentPane().add(btn4);
118
119      btn1 = new JButton("1");
120      btn1.addActionListener(new ActionListener() {
121          public void actionPerformed(ActionEvent e) {
122              String number=textField.getText()+btn1.getText();
123              textField.setText(number);
124          }
125      });
126      btn1.setFont(new Font("Tahoma", Font.BOLD, 18));
127      btn1.setBounds(10, 212, 68, 44);
128      frame.getContentPane().add(btn1);
129
130      btn0 = new JButton("0");
131      btn0.addActionListener(new ActionListener() {
132          public void actionPerformed(ActionEvent e) {
133              String number=textField.getText()+btn0.getText();
134              textField.setText(number);
135          }
136      });
137      btn0.setFont(new Font("Tahoma", Font.BOLD, 18));
138      btn0.setBounds(10, 257, 68, 44);
139      frame.getContentPane().add(btn0);
140

```



```

141      btnClear = new JButton("C");
142      btnClear.addActionListener(new ActionListener() {
143          public void actionPerformed(ActionEvent e) {
144              textField.setText(null);
145          }
146      });
147      btnClear.setFont(new Font("Tahoma", Font.BOLD, 18));
148      btnClear.setBounds(77, 78, 68, 44);
149      frame.getContentPane().add(btnClear);
150
151      btn8 = new JButton("8");
152      btn8.addActionListener(new ActionListener() {
153          public void actionPerformed(ActionEvent e) {
154              String number=textField.getText()+btn8.getText();
155              textField.setText(number);
156          }
157      });
158      btn8.setFont(new Font("Tahoma", Font.BOLD, 18));
159      btn8.setBounds(77, 122, 68, 44);
160      frame.getContentPane().add(btn8);
161
162      btn5 = new JButton("5");
163      btn5.addActionListener(new ActionListener() {
164          public void actionPerformed(ActionEvent e) {
165              String number=textField.getText()+btn5.getText();
166              textField.setText(number);
167          }
168      });
169      btn5.setFont(new Font("Tahoma", Font.BOLD, 18));
170      btn5.setBounds(77, 167, 68, 44);
171      frame.getContentPane().add(btn5);
172
173      btn2 = new JButton("2");
174      btn2.addActionListener(new ActionListener() {
175
176          public void actionPerformed(ActionEvent e) {
177              String number=textField.getText()+btn2.getText();
178              textField.setText(number);
179          }
180      });
181      btn2.setFont(new Font("Tahoma", Font.BOLD, 18));
182      btn2.setBounds(77, 212, 68, 44);
183      frame.getContentPane().add(btn2);
184
185      btnDot = new JButton(".");
186      btnDot.addActionListener(new ActionListener() {
187          public void actionPerformed(ActionEvent e) {
188              String number=textField.getText()+btnDot.getText();
189              textField.setText(number);
190          }
191      });
192      btnDot.setFont(new Font("Tahoma", Font.BOLD, 18));
193      btnDot.setBounds(77, 257, 68, 44);
194      frame.getContentPane().add(btnDot);
195
196      btn00 = new JButton("00");
197      btn00.addActionListener(new ActionListener() {
198          public void actionPerformed(ActionEvent e) {
199              String number=textField.getText()+btn00.getText();
200              textField.setText(number);
201          }
202      });
203      btn00.setFont(new Font("Tahoma", Font.BOLD, 18));
204      btn00.setBounds(144, 78, 68, 44);
205      frame.getContentPane().add(btn00);
206
207      btn9 = new JButton("9");
208      btn9.addActionListener(new ActionListener() {

```

```

208 public void actionPerformed(ActionEvent e) {
209     String number=textField.getText()+btn9.getText();
210     textField.setText(number);
211 }
212 });
213 btn9.setFont(new Font("Tahoma", Font.BOLD, 18));
214 btn9.setBounds(144, 122, 68, 44);
215 frame.getContentPane().add(btn9);
216
217 btn6 = new JButton("6");
218 btn6.addActionListener(new ActionListener() {
219     public void actionPerformed(ActionEvent e) {
220         String number=textField.getText()+btn6.getText();
221         textField.setText(number);
222     }
223 });
224 btn6.setFont(new Font("Tahoma", Font.BOLD, 18));
225 btn6.setBounds(144, 167, 68, 44);
226 frame.getContentPane().add(btn6);
227
228 btn3 = new JButton("3");
229 btn3.addActionListener(new ActionListener() {
230     public void actionPerformed(ActionEvent e) {
231         String number=textField.getText()+btn3.getText();
232         textField.setText(number);
233     }
234 });
235 btn3.setFont(new Font("Tahoma", Font.BOLD, 18));
236 btn3.setBounds(144, 212, 68, 44);
237 frame.getContentPane().add(btn3);
238
239 btnEqual = new JButton("=");
240 btnEqual.addActionListener(new ActionListener() {
241     public void actionPerformed(ActionEvent e) {
242         String answer;
243         second=Double.parseDouble(textField.getText());
244         if(operation=="+")
245         {
246             result=first+second;
247             answer=String.format("%.2f", result);
248             textField.setText(answer);
249         }
250         else if(operation=="-")
251         {
252             result=first-second;
253             answer=String.format("%.2f", result);
254             textField.setText(answer);
255         }
256         else if(operation=="*")
257         {
258             result=first*second;
259             answer=String.format("%.2f", result);
260             textField.setText(answer);
261         }
262         else if(operation=="/")
263         {
264             result=first/second;
265             answer=String.format("%.2f", result);
266             textField.setText(answer);
267         }
268         else if(operation=="%")
269         {
270             result=first%second;
271             answer=String.format("%.2f", result);
272             textField.setText(answer);
273         }
274     }

```

```

275     }
276 });
277 btnEqual.setFont(new Font("Tahoma", Font.BOLD, 18));
278 btnEqual.setBounds(144, 257, 68, 44);
279 frame.getContentPane().add(btnEqual);
280
281 btnPlus = new JButton("+");
282 btnPlus.addActionListener(new ActionListener() {
283     public void actionPerformed(ActionEvent e) {
284         first=Double.parseDouble(textField.getText());
285         textField.setText("");
286         operation="+";
287     }
288 });
289 btnPlus.setFont(new Font("Tahoma", Font.BOLD, 18));
290 btnPlus.setBounds(213, 78, 68, 44);
291 frame.getContentPane().add(btnPlus);
292
293 btnSub = new JButton("-");
294 btnSub.addActionListener(new ActionListener() {
295     public void actionPerformed(ActionEvent e) {
296         first=Double.parseDouble(textField.getText());
297         textField.setText("");
298         operation="-";
299     }
300 });
301 btnSub.setFont(new Font("Tahoma", Font.BOLD, 18));
302 btnSub.setBounds(213, 122, 68, 44);
303 frame.getContentPane().add(btnSub);
304
305 btnMul = new JButton("*");
306 btnMul.addActionListener(new ActionListener() {
307     public void actionPerformed(ActionEvent e) {
308
309         first=Double.parseDouble(textField.getText());
310         textField.setText("");
311         operation="*";
312     }
313 });
314 btnMul.setFont(new Font("Tahoma", Font.BOLD, 18));
315 btnMul.setBounds(213, 167, 68, 44);
316 frame.getContentPane().add(btnMul);
317
318 btnDivide = new JButton("/");
319 btnDivide.addActionListener(new ActionListener() {
320     public void actionPerformed(ActionEvent e) {
321         first=Double.parseDouble(textField.getText());
322         textField.setText("");
323         operation="/";
324     }
325 });
326 btnDivide.setFont(new Font("Tahoma", Font.BOLD, 18));
327 btnDivide.setBounds(213, 212, 68, 44);
328 frame.getContentPane().add(btnDivide);
329
330 btnPercent = new JButton("%");
331 btnPercent.addActionListener(new ActionListener() {
332     public void actionPerformed(ActionEvent e) {
333         first=Double.parseDouble(textField.getText());
334         textField.setText("");
335         operation="%";
336     }
337 });
338 btnPercent.setFont(new Font("Tahoma", Font.BOLD, 18));
339 btnPercent.setBounds(213, 257, 68, 44);
340 frame.getContentPane().add(btnPercent);
341 }

```

Code Program 2.1 (Code Program CalculatorGUI)

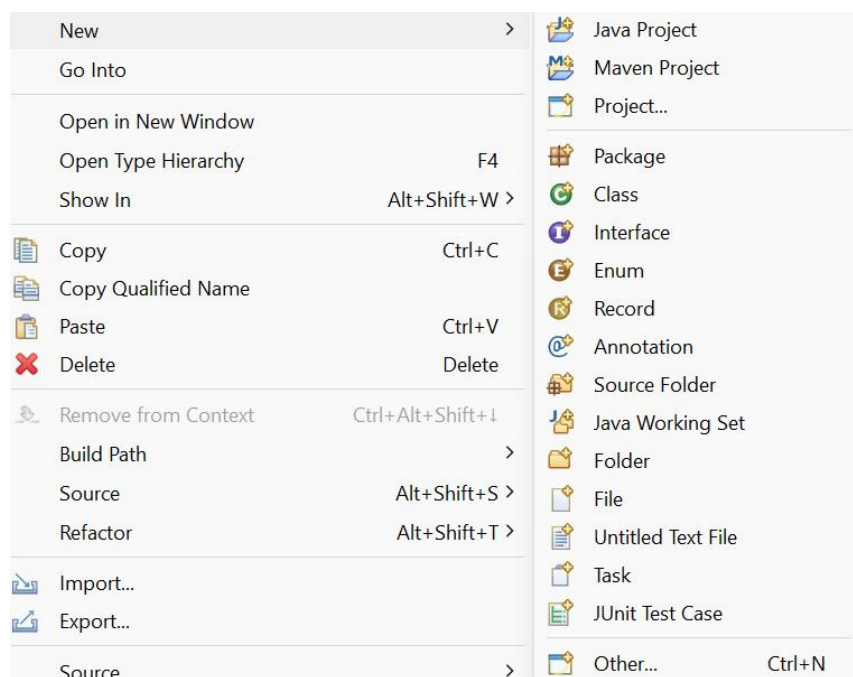
2.2 Langkah Kerja

1. Membuat *package* bernama *Pekan9_2511531016*.



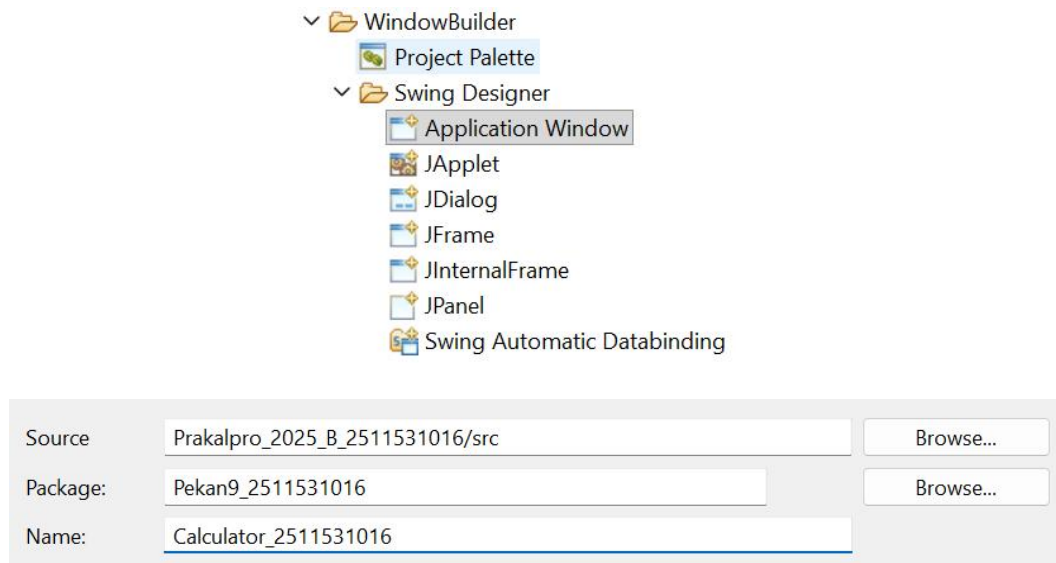
Gambar 2.1 (Langkah 1 CalculatorGUI)

2. Klik kanan pada package dan pilih *new* lalu klik *other*.



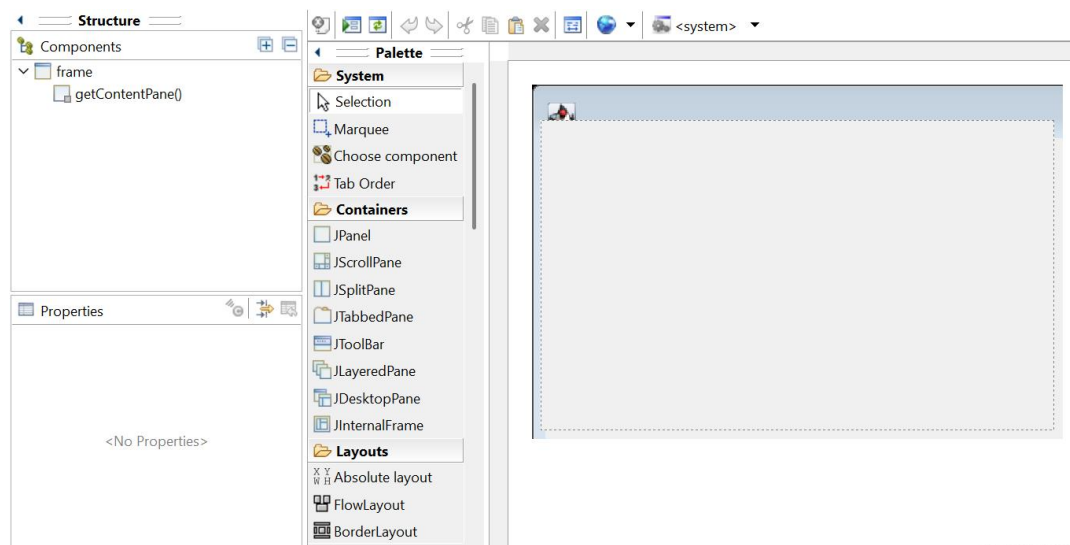
Gambar 2.2 (Langkah 2 CalculatorGUI)

3. Klik *Application Window* pada menu *WindowBuilder* dan beri nama *Calculator_2511531016*.



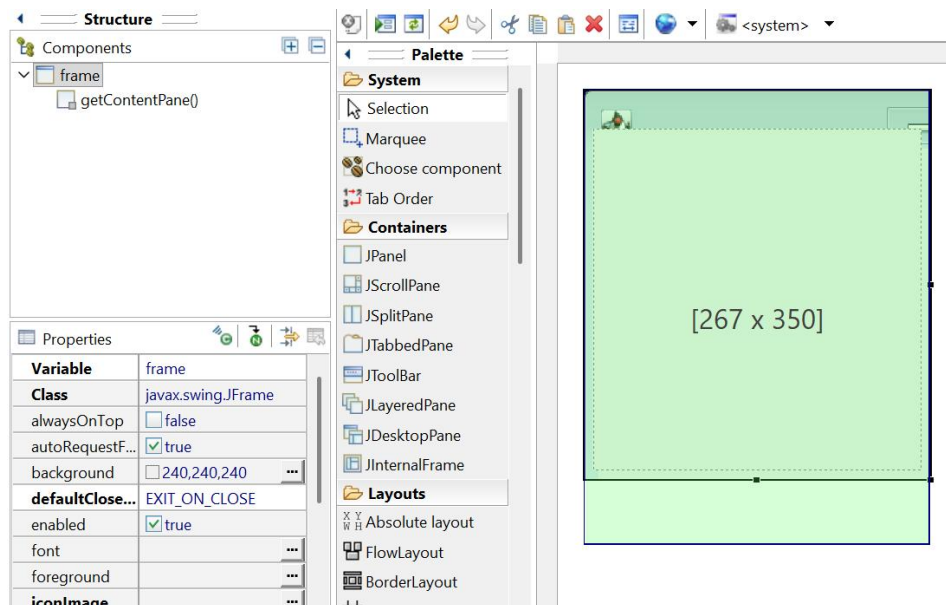
Gambar 2.3 (Langkah 3 CalculatorGUI)

4. Akan muncul tampilan seperti ini.



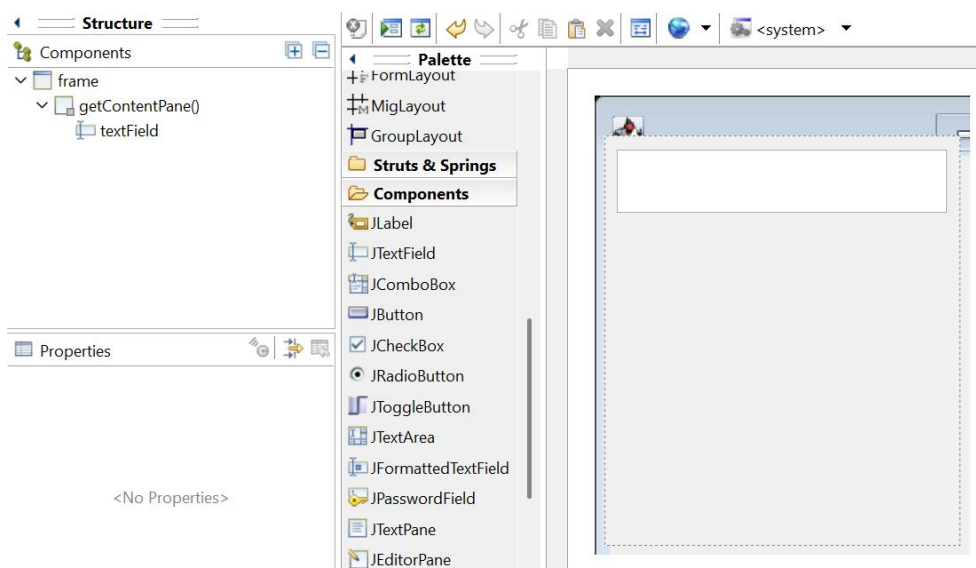
Gambar 2.4 (Langkah 4 CalculatorGUI)

5. Setelah itu, klik *frame* dan ubah *layout* nya menjadi *absolute layout*. Lalu atur ukuran sesuai keinginan.



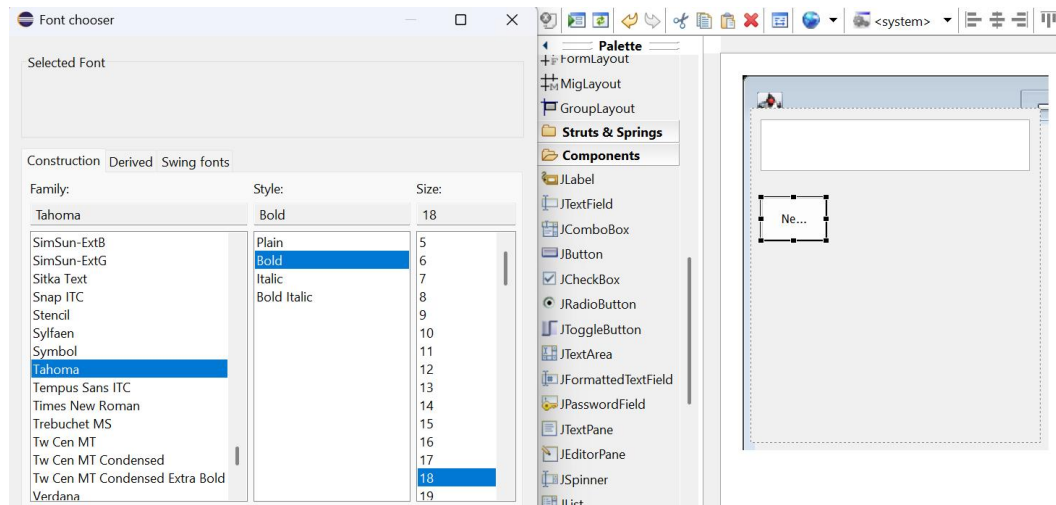
Gambar 2.5 (Langkah 5 CalculatorGUI)

6. Pada menu *component* klik *JTextField* (tempat dimana semua angka ditampilkan), lalu buat ukurannya seperti gambar dibawah ini.



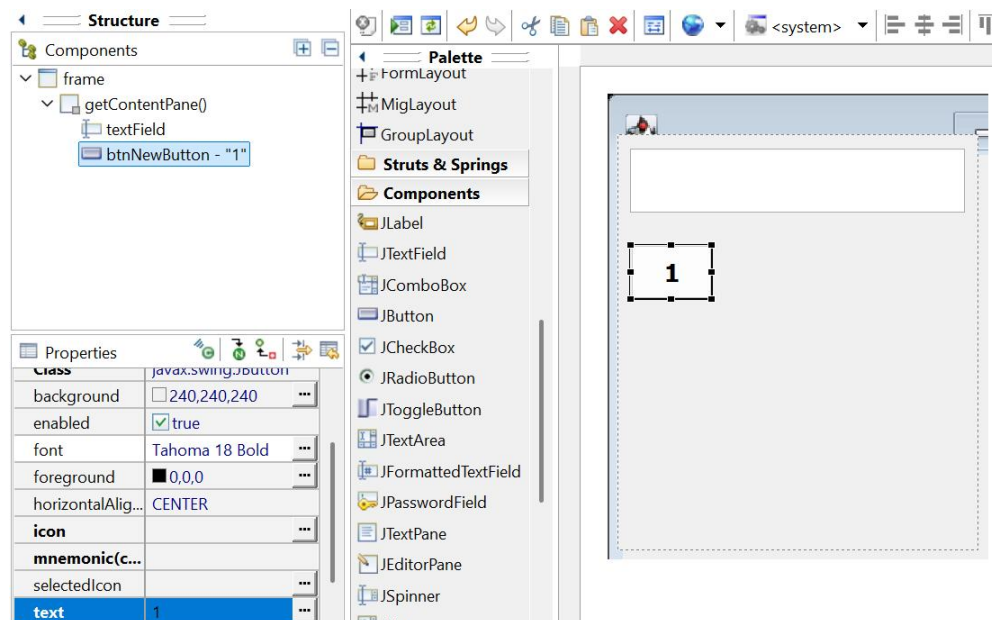
Gambar 2.6 (Langkah 6 CalculatorGUI)

7. Tambahkan *JButton* dan ubah ukurannya sesuai keinginan. Lalu sesuaikan *font*, *style* dan *size*.



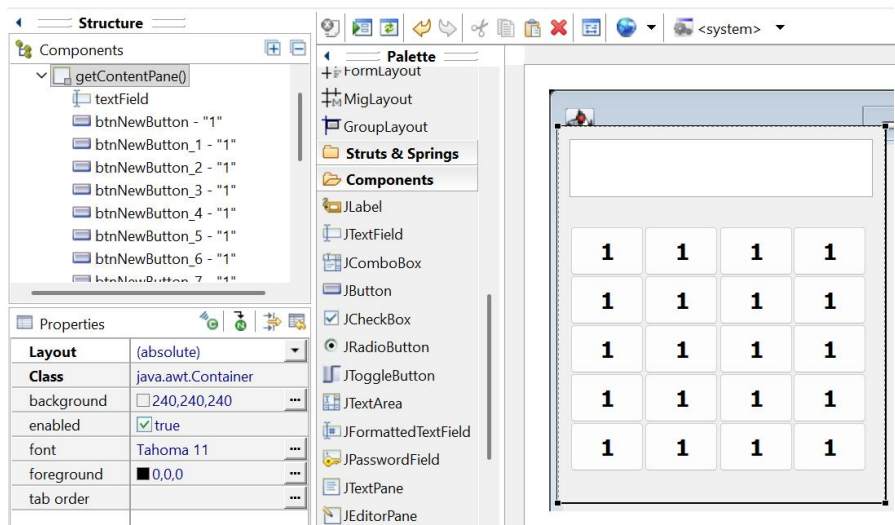
Gambar 2.7 (Langkah 7 CalculatorGUI)

8. Ubah *text* menjadi “1”



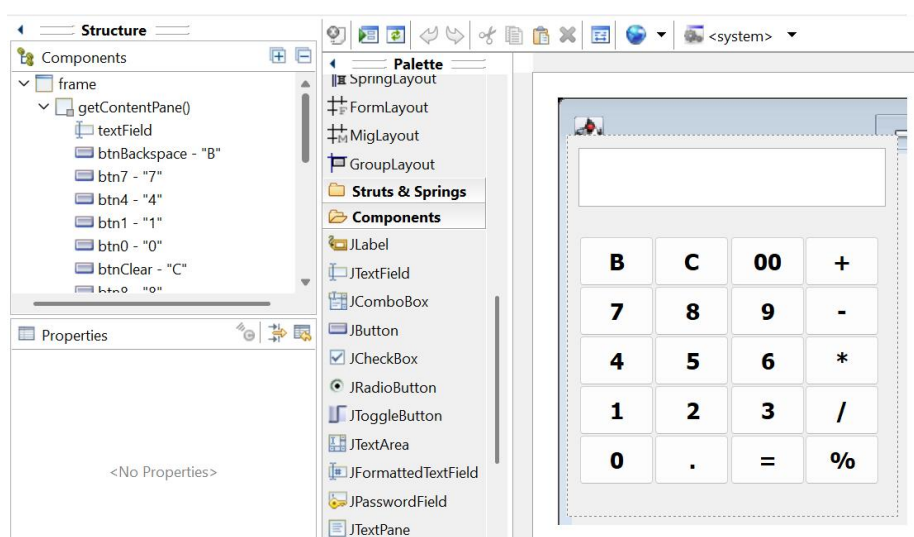
Gambar 2.8 (Langkah 8 CalculatorGUI)

9. *Copy* dan *paste* angka “1” tersebut menjadi 20 bagian, 4 ke kanan dan 5 ke bawah.



Gambar 2.9 (Langkah 9 CalculatorGUI)

10. Ubah teks dan nama variabel. Contoh : 1, maka nama variabelnya diganti menjadi *btn1*. Begitu seterusnya dari 00 hingga 9. Untuk operator nya juga sama, Contoh : C, maka nama variabelnya diganti menjadi *btnClear*. Ikuti langkah yang sama untuk operator lainnya.



Gambar 2.10 (Langkah 10 CalculatorGUI)

11. Untuk *btnBackspace*, kita akan mengubah “B” menjadi panah, maka gunakan kode program dibawah ini untuk mengubah dan menjalankannya.

```

JButton btnBackspace = new JButton("\uF0E7");
btnBackspace.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        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);
        }
    }
});

```

Gambar 2.11 (Langkah 11 CalculatorGUI)

12. Agar tombol berfungsi dan bisa digunakan, maka kita akan menambahkan *source code* pada masing-masing angka (00 hingga 9) dan “.” (*dot*). Cara nya yaitu, tekan 2x pada masing-masing angka di design, lalu nanti akan langsung diarahkan pada *source code*. Kode nya sama, hanya perlu di ubah pada bagian “*btn0*” menjadi *btn* masing-masing angka dan “.” (*dot*).

```

btn0 = new JButton("0");
btn0.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String number=textField.getText()+btn0.getText();
        textField.setText(number);
    }
});

```

Gambar 2.12 (Langkah 12 CalculatorGUI)

13. Untuk *btnClear*, gunakan *source code* dibawah ini untuk menjalankannya.

```
btnClear = new JButton("C");
btnClear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
    }
}
```

Gambar 2.13 (Langkah 13 CalculatorGUI)

14. Untuk menjalankan operator (+ - * / %), gunakan kode dibawah ini. Namun, perlu diganti pada bagian *btnPlus* menjadi nama variabel masing-masing operator dan pada *operation* juga diganti menjadi simbol masing-masing operator.

```
btnPlus = new JButton("+");
btnPlus.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        first=Double.parseDouble(textField.getText());
        textField.setText("");
        operation="+";
    }
}
```

Gambar 2.14 (Langkah 14 CalculatorGUI)

15. Untuk menjalankan *btnEqual*, gunakan *source code* dibawah ini.

```
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);
        }
    }
});
```

Gambar 2.15 (Langkah 15 CalculatorGUI)

16. Jangan lupa untuk menambahkan *source code* dibawah ini untuk menjalankan operator.

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

Gambar 2.16 (Langkah 16 CalculatorGUI)

2.3 Analisis Hasil

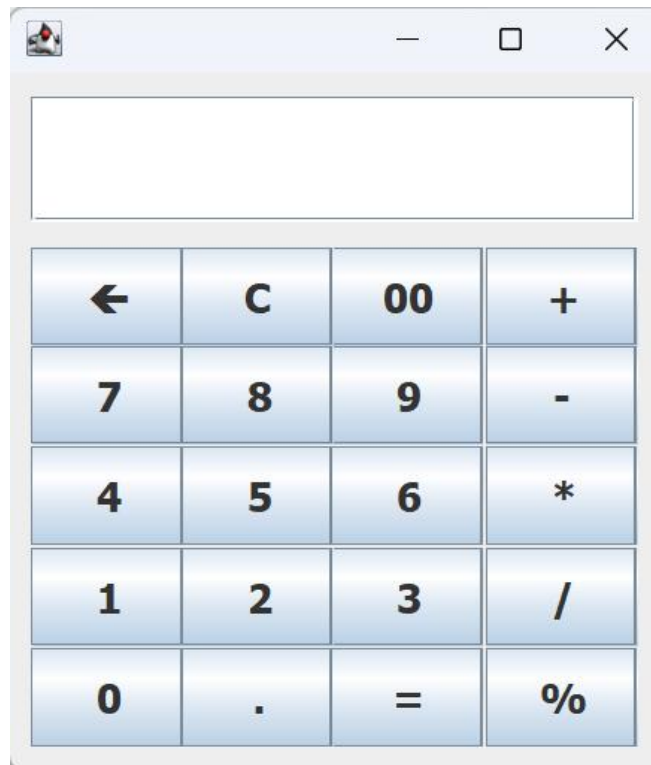
Berdasarkan hasil percobaan dari kode program yang telah dijalankan, dapat dilihat bahwa aplikasi kalkulator berbasis Java GUI telah berhasil berjalan dengan baik dan menampilkan antarmuka yang lengkap serta mudah digunakan. Program menampilkan sebuah layar input di bagian atas yang berfungsi sebagai tempat menampilkan angka yang ditekan dan hasil perhitungan. Di bawahnya terdapat tombol-tombol angka, operator aritmatika seperti $+$, $-$, $*$, dan $/$, serta fitur tambahan seperti tombol *backspace*, *clear* (C), *persen* (%), titik desimal, dan tombol "00". Susunan tombol yang rapi dan menyerupai kalkulator fisik membuat pengguna lebih mudah memahami cara pengoperasiannya, sehingga proses input dan perhitungan menjadi lebih intuitif.

Selain itu, fungsi *event handling* pada program terlihat berjalan dengan baik, di mana setiap tombol yang ditekan memberikan respons sesuai perintah, seperti menambahkan angka ke layar input atau melakukan operasi perhitungan ketika tombol "=" ditekan. Hal ini menunjukkan bahwa logika program dan penghubungan antara komponen GUI dengan fungsi perhitungan telah diterapkan dengan benar. Tampilan tombol yang dibuat menggunakan *layout* terstruktur juga memberikan kesan profesional dan *user-friendly*.

Secara keseluruhan, hasil praktikum ini menunjukkan bahwa mahasiswa mampu mengimplementasikan konsep dasar GUI pada Java, mulai dari penempatan komponen, pengaturan *layout*, hingga pemrosesan input pengguna. Program kalkulator yang dihasilkan tidak hanya fungsional, tetapi juga memiliki tampilan yang menarik dan mudah digunakan, sehingga dapat menjadi dasar pengembangan aplikasi GUI yang lebih kompleks di masa mendatang.

BAB III KESIMPULAN

3.1 Hasil Praktikum



Gambar 3.1 (Hasil Praktikum CalculatorGUI)

Output di atas menunjukkan tampilan program kalkulator berbasis Java GUI yang telah berhasil dijalankan dengan baik. Pada tampilan tersebut terdapat sebuah layar *input* di bagian atas yang berfungsi untuk menampilkan angka dan hasil perhitungan, serta kumpulan tombol angka, operator aritmatika (+, -, *, /), tombol fungsi seperti *backspace*, *clear* (C), *persen* (%), titik desimal, dan tombol "=" sebagai hasil. Susunan tombol yang rapi menyerupai kalkulator fisik membuat program mudah digunakan dan dipahami oleh pengguna, sehingga interaksi dan proses perhitungan dapat dilakukan secara lebih cepat dan praktis.

3.2 Saran Pengembangan

Berdasarkan hasil analisis dari kode program yang telah dijalankan, terdapat beberapa saran pengembangan agar program menjadi lebih interaktif, efisien, dan bermanfaat dalam penerapan nyata.

Saran pengembangan untuk program kalkulator ini dapat dimulai dengan penambahan fitur yang lebih lengkap agar penggunaan menjadi lebih nyaman dan efisien. Misalnya, menambahkan riwayat perhitungan (*history*) yang memungkinkan pengguna melihat hasil sebelumnya tanpa harus menghitung ulang. Selain itu, penanganan *error* juga perlu ditingkatkan, seperti memberikan notifikasi atau pesan khusus saat terjadi pembagian dengan nol atau input yang tidak valid, sehingga program menjadi lebih aman dan tidak membuat pengguna bingung. Pengaturan tampilan juga dapat diperbaiki dengan desain yang lebih modern, penggunaan warna yang lebih menarik, serta tata letak tombol yang lebih proporsional.

Ke depannya, kalkulator ini juga dapat dikembangkan menjadi versi *scientific calculator* dengan menambahkan fitur seperti akar, pangkat, trigonometri, dan fungsi matematika lainnya agar penggunaannya lebih luas. Integrasi dengan *keyboard* input juga dapat menjadi nilai tambah, sehingga pengguna bisa melakukan perhitungan tanpa harus selalu menekan tombol pada GUI. Dengan pengembangan tersebut, kalkulator tidak hanya menjadi media latihan, tetapi juga dapat digunakan sebagai aplikasi yang lebih fungsional dan menarik.

DAFTAR PUSTAKA

- [1] H. M. Deitel and P. J. Deitel, *Java: How to Program*, 10th ed. Boston: Pearson, 2015.
- [2] J. N. Cross, *Java Swing*, 2nd ed. Sebastopol: O'Reilly Media, 2014.
- [3] Oracle, “*Creating a GUI With Swing*,” *The Java Tutorials*,
<https://docs.oracle.com/javase/tutorial/uiswing/TOC.html>. [Oracle Documentation+1](#)
- [4] J. Farley, *Introducing JavaFX*, New York: Apress, 2012.
- [5] P. G. Bishop, “*Event Handling and GUI Design in Java Applications*,”
Journal of Computing Education, vol. 6, no. 2, pp. 45–52, 2021.