

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“PEMROGRAMAN GUI 2 (CALCULATOR)”



disusun Oleh:
NAIRA RAMADHANI HALIL
2511533027
Dosen Pengampu:
Dr. WAHYUDI, S.T, M.T.
Asisten Praktikum:
JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “**PEMROGRAMAN GUI 2 (CALCULATOR)**” dapat diselesaikan tepat waktu.

Laporan ini disusun untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran dalam memahami konsep dasar pemrograman khususnya materi pemrograman GUI pada Bahasa Java.

Pada kesempatan ini, penulis menyampaikan terimakasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku dosen pengampu mata kuliah Algoritma dan pemrograman yang telah memberikan bimbingan dan arahan.
2. Bang Jovantri Immanuel Gulo selaku asisten praktikum kelas A yang telah membantu pelaksanaan praktikum.
3. Teman-teman mahasiswa yang telah membantu dan memberi dukungan serta berdiskusi bersama dalam menyelesaikan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca dan menambah wawasan mengenai pemrograman Java.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB II PE,BAHASAN.....	3
2.1 Langkah Kerja Praktikum	3
2.2 Analisis Hasil dan Pembahasan	16
BAB III PENUTUP.....	18
3.1 Kesimpulan	18
DAFTAR PUSTAKA	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di pekan 9 ini, dimana pekan terakhir untuk kami belajar sebelum UAP (Ujian Akhir Praktikum), kami membuat GUI Calculator. Kali ini, Java menyediakan berbagai library untuk membuat aplikasi GUI, salah satunya adalah Java Swing. Swing memungkinkan programmer membuat tampilan aplikasi yang interaktif seperti tombol, text field, window, dan event-event yang merespon aksi pengguna.

Java Swing memberikan berbagai komponen visual seperti tombol (JButton), kotak teks (JTextField), dan jendela aplikasi (JFrame) yang memungkinkan programmer membuat aplikasi yang interaktif dan mudah digunakan. Pemahaman mengenai pemrograman GUI sangat penting karena sebagian besar aplikasi yang berinteraksi langsung dengan pengguna memerlukan tampilan grafis yang intuitif dan responsif. Dengan memahami dasar-dasar pembuatan GUI, mahasiswa dapat mengembangkan aplikasi yang tidak hanya berjalan melalui terminal, tetapi juga memiliki tampilan visual yang lebih menarik dan fungsional.

Pada program membuat kalkulator sederhana ini, program ini terdiri dari komponen tombol angka, tombol operasi, text field sebagai layer kalkulator, serta proses perhitungan menggunakan event handling. Pembuatan kalkulator ini penting sebagai dasar untuk memahami pengembangan aplikasi GUI lebih lanjut. Di pekan terakhir ini juga semua materi yang telah dipelajari dari awal akan masuk dan digunakan kedalam program kalkulator ini.

1.2 Tujuan Praktikum

1. Mengenal komponen dasar GUI Java Swing.
2. Memahami cara kerja event handling (ActionListener).
3. Membuat layout dan mengatur posisi komponen GUI.
4. Menerapkan logika perhitungan dalam program GUI.
5. Membuat aplikasi kalkulator sederhana menggunakan Java.

1.3 Manfaat Praktikum

1. Mahasiswa dapat membuat aplikasi GUI sederhana.
2. Meningkatkan kemampuan pemrograman berbasis event.
3. Memahami cara menghubungkan komponen GUI dengan logika program.
4. Menjadi dasar untuk membuat aplikasi desktop lebih kompleks.

BAB II

PEMBAHASAN

2.1 Langkah Kerja Praktikum

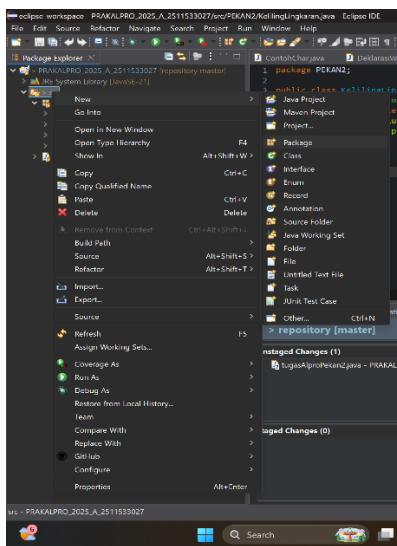
1. Persiapan Awal

- Buka aplikasi Eclipse IDE di laptop atau di komputer.

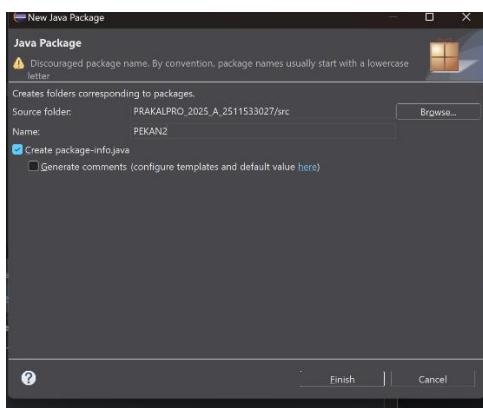


2. Membuat Package

- Klik kana pada **src**, lalu pilih **New**, terakhir klik **Package**.

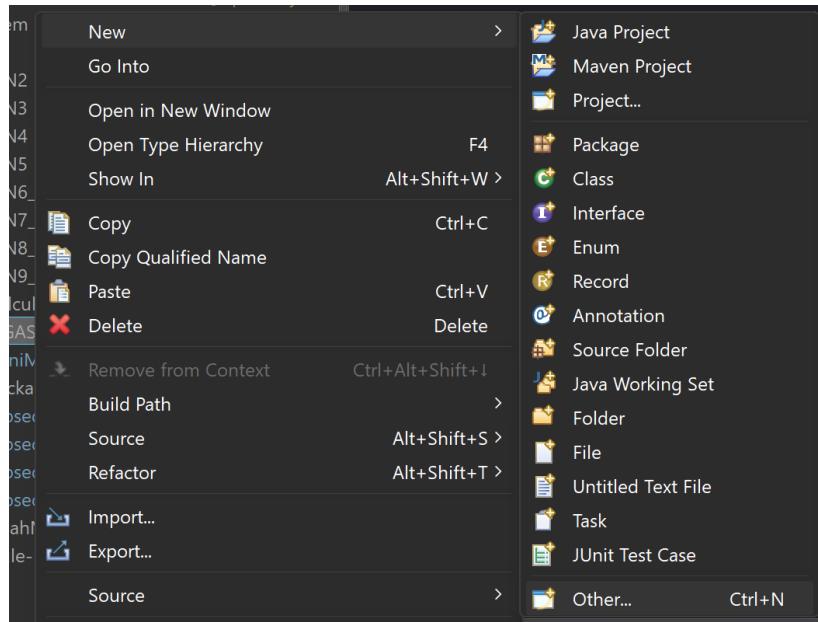


- Beri nama package **PEKAN9_2511533027**.

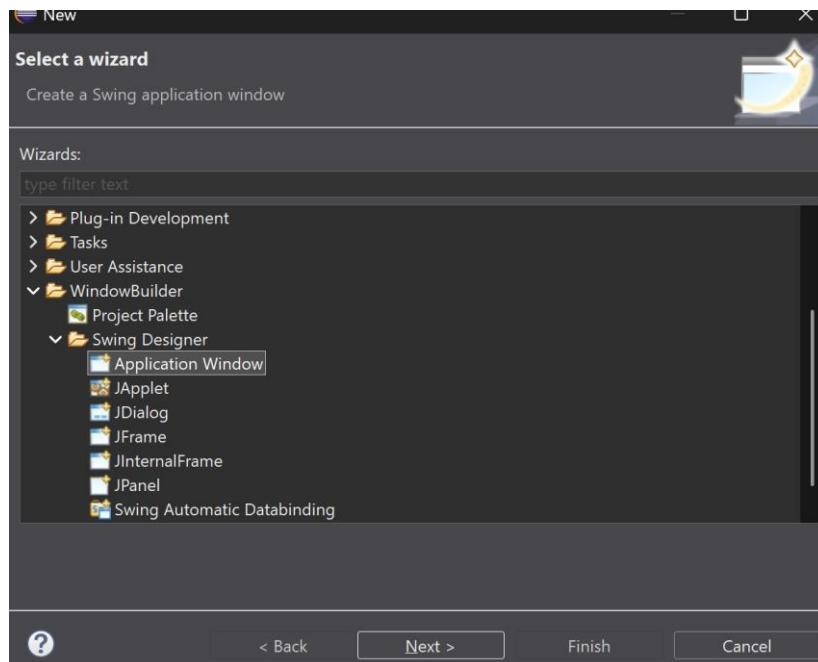


3. Membuat Window Builder

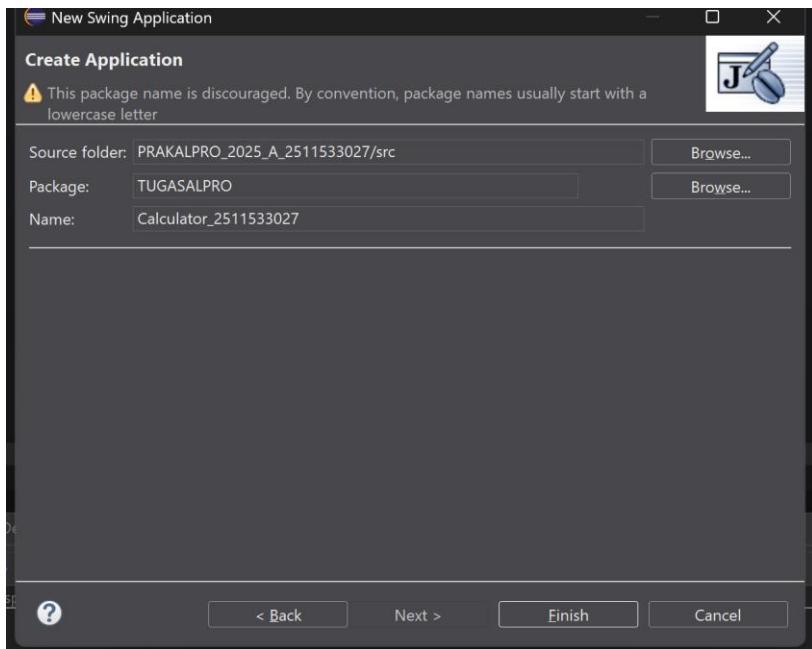
- a) Klik kanan pada package **PEKAN9_2511533027**, lalu klik **New**, terakhir klik **Other**.



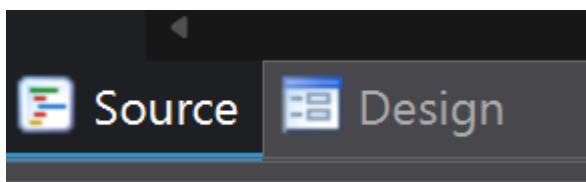
- b) Klik tanda panah > pada **WindowBuilder**, klik tanda panah lagi pada **Swing Designer**, dan terakhir klik Application Window.



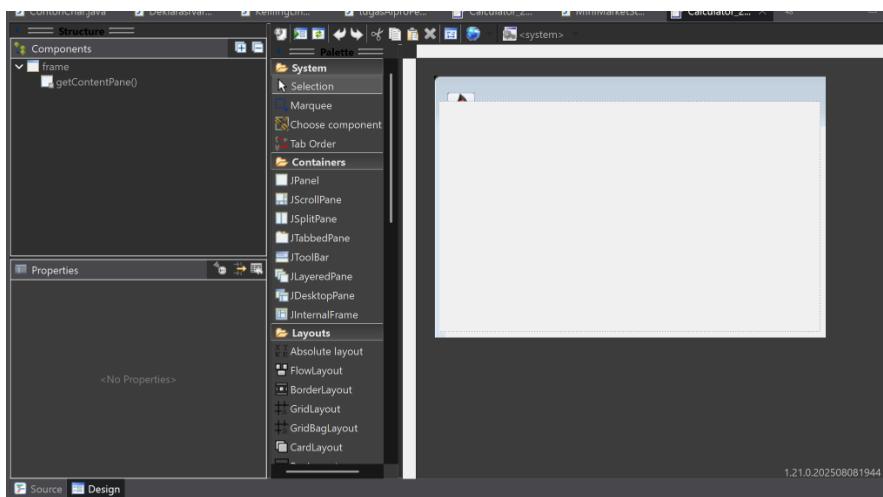
- c) Lalu beri nama **Calculator_NIM**.



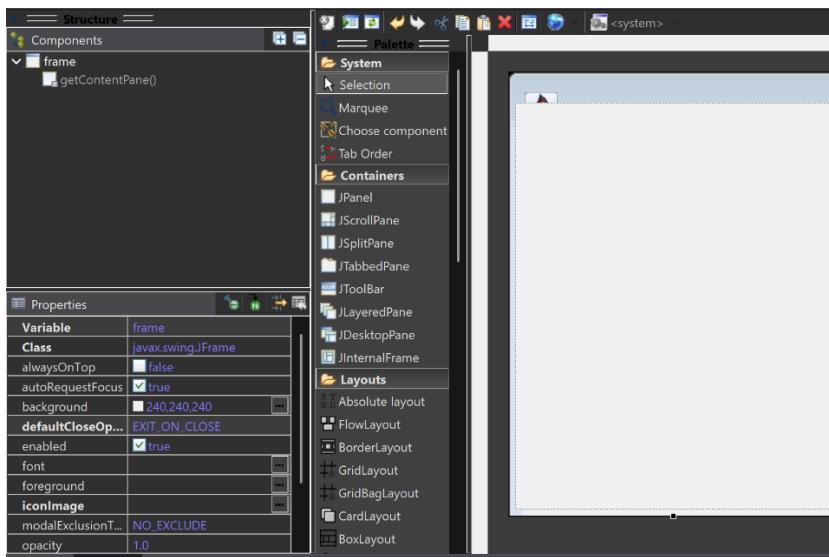
d) Lalu klik **Design**.



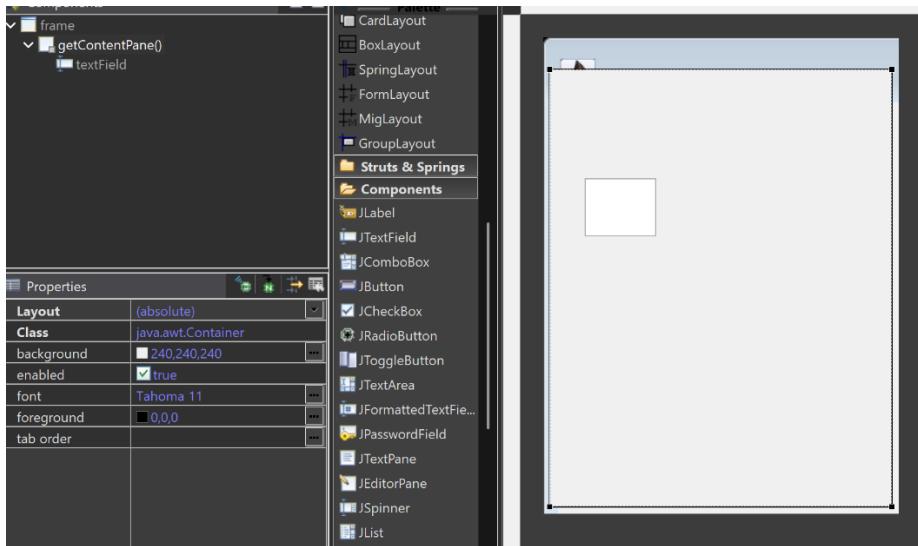
Sehingga menampilkan:



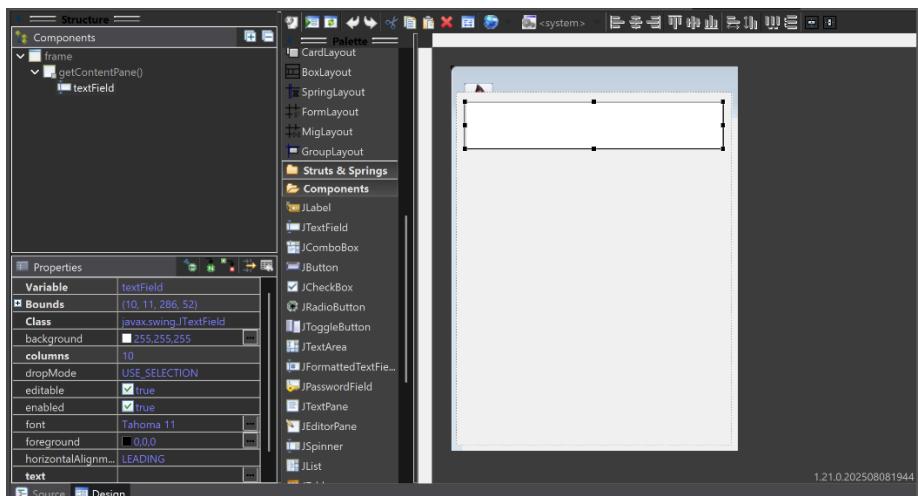
e) Sesuaikan ukuran design, lalu klik **Absolute layout**.



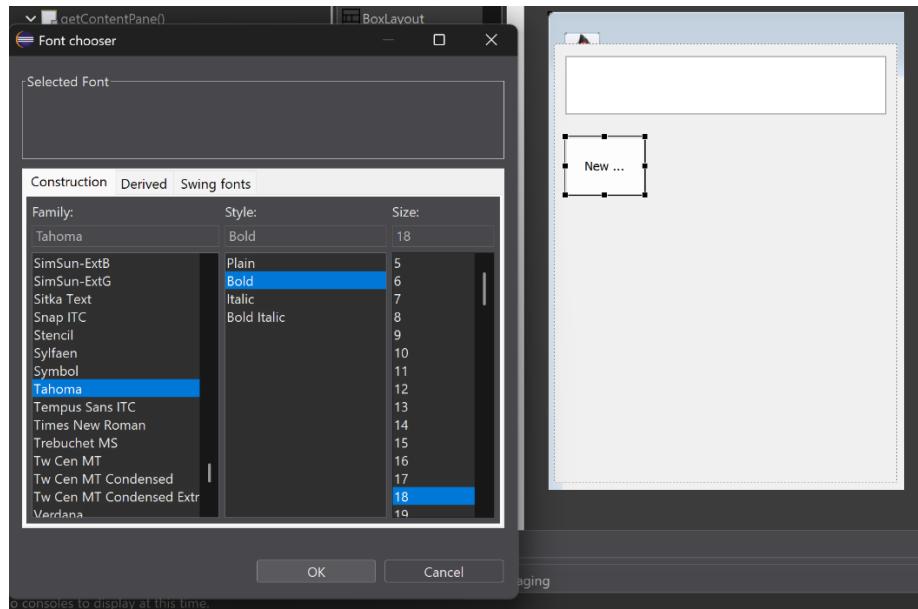
- f) Klik JTextField lalu sesuaikan ukuran untuk layar kecil yang akan menampilkan angkanya. Lalu klik **font** untuk memperbesar dan mempertebal tulisan.



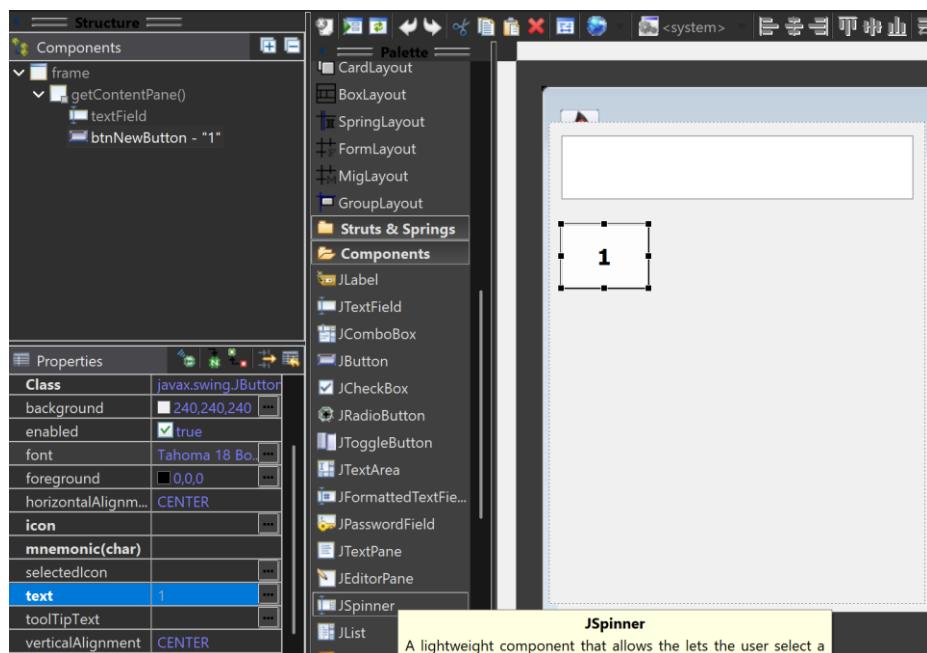
Resize sehingga menjadi seperti ini:



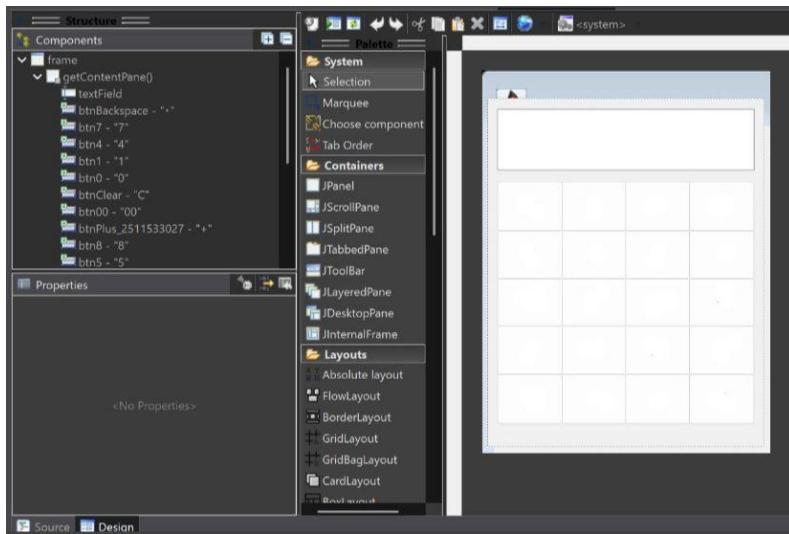
- g) Klik JButton, lalu resize untuk tombol angka kalkulator dan klik **font** untuk memperbesar dan mempertebal (bold) tulisan.



Ganti text pada Properties menjadi angka 1.



Copy paste JButton dan susun 4x5 sehingga menjadi seperti ini:



Lalu pada setiap JButton (khususnya untuk tombol untuk angka 0-9) ganti text menjadi angka 0-9 dan pada **Variable** pada **Properties** ubah text menjadi **btn1** sesuai dengan angka pada text.

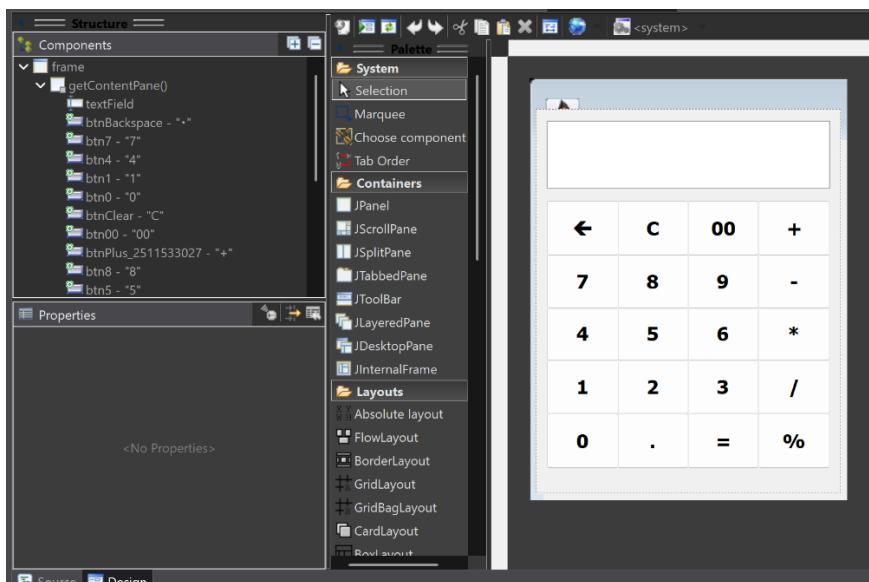
Pada 3 JButton pertama, text digunakan untuk Backspace, Clear, dan 00. Sehingga **Variable** dan **text** diganti dengan Backspace, Clear, dan 00.

Untuk 5 JButton pertama dari kiri digunakan untuk Operator Aritmatika (+, -, /, *, %).

Ganti **Variable** dan **text** dengan operator aritmatika secara berurutan.

JButton yang tersisa diganti dengan Dot (.) dan Equal (=).

Sehingga menjadi:



Jika sudah siap semua, lanjut menuliskan kode program agar kalkulator bisa digunakan.

4. Menulis Kode Program

Lalu pada kode program nya, tambahkan event handling (ActionListener). Untuk setiap tombol angka dan tombol “00” atau titik, tambahkan ActionListener agar ketika diklik, teks tombol ditambahkan ke textField. Untuk tombol operasi (+, -, ×, ÷, %): ketika diklik, simpan nilai pertama (first), kosongkan textField, dan simpan jenis operasi ke variabel operation. Untuk tombol backspace: jika textField tidak kosong, hapus karakter terakhir. Untuk tombol clear (“C”): kosongkan seluruh isi textField. Terakhir, membuat logika perhitungan. Saat tombol “=” diklik:

```

1 package PEKAN9_2511533027;
2
3@import java.awt.EventQueue;□
11
12 public class Calculator_2511533027 {
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
23
24@     /**
25      * Launch the application.
26      */
27@     public static void main(String[] args) {
28@         EventQueue.invokeLater(new Runnable() {
29@             public void run() {
30@                 try {
31                     Calculator_2511533027 window = new Calculator_2511533027();
32                     window.frame.setVisible(true);
33@                 } catch (Exception e) {
34                     e.printStackTrace();
35                 }
36             }
37         });
38     }
39
40@     /**
41      * Create the application.
42      */
43@     public Calculator_2511533027() {
44         initialize();
45     }
46
47@     /**
48      * Initialize the contents of the frame.
49      */
50@     private void initialize() {
51         frame = new JFrame();
52         frame.setBounds(100, 100, 299, 397);
53         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
54         frame.getContentPane().setLayout(null);
55
56         textField = new JTextField();
57         textField.setFont(new Font("Tahoma", Font.BOLD, 18));
58         textField.setBounds(10, 11, 267, 64);
59         frame.getContentPane().add(textField);
60         textField.setColumns(10);
61
62         JButton btnBackspace = new JButton("\u00d7");
63@         btnBackspace.addActionListener(new ActionListener() {
64@             public void actionPerformed(ActionEvent e) {
65                 String backSpace=null;

```

```

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     btnBackspace.setFont(new Font("Wingdings", Font.BOLD, 18));
75     btnBackspace.setBounds(10, 86, 68, 52);
76     frame.getContentPane().add(btnBackspace);
77
78     JButton btn7 = new JButton("7");
79     btn7.addActionListener(new ActionListener() {
80         public void actionPerformed(ActionEvent e) {
81             String number=textField.getText()+btn7.getText();
82             textField.setText(number);
83         }
84     });
85     btn7.setFont(new Font("Tahoma", Font.BOLD, 18));
86     btn7.setBounds(10, 136, 68, 52);
87     frame.getContentPane().add(btn7);
88
89     JButton btn4 = new JButton("4");
90     btn4.addActionListener(new ActionListener() {
91         public void actionPerformed(ActionEvent e) {
92             String number=textField.getText()+btn4.getText();
93
94             public void actionPerformed(ActionEvent e) {
95                 String number=textField.getText()+btn4.getText();
96                 textField.setText(number);
97             }
98         });
99         btn4.setFont(new Font("Tahoma", Font.BOLD, 18));
100        btn4.setBounds(10, 185, 68, 52);
101        frame.getContentPane().add(btn4);
102
103        JButton btn1 = new JButton("1");
104        btn1.addActionListener(new ActionListener() {
105            public void actionPerformed(ActionEvent e) {
106                String number=textField.getText()+btn1.getText();
107                textField.setText(number);
108            }
109        });
110        btn1.setFont(new Font("Tahoma", Font.BOLD, 18));
111        btn1.setBounds(10, 235, 68, 52);
112        frame.getContentPane().add(btn1);
113
114        JButton btn0 = new JButton("0");
115        btn0.addActionListener(new ActionListener() {
116            public void actionPerformed(ActionEvent e) {
117                String number=textField.getText()+btn0.getText();
118                textField.setText(number);
119            }
120        });
121        btn0.setFont(new Font("Tahoma", Font.BOLD, 18));
122        btn0.setBounds(10, 286, 68, 52);
123        frame.getContentPane().add(btn0);

```

```

120     btn0.setBounds(10, 286, 68, 52);
121     frame.getContentPane().add(btn0);
122
123     JButton btnClear = new JButton("C");
124     btnClear.addActionListener(new ActionListener() {
125         public void actionPerformed(ActionEvent e) {
126             textField.setText(null);
127         }
128     });
129     btnClear.setFont(new Font("Tahoma", Font.BOLD, 18));
130     btnClear.setBounds(76, 86, 68, 52);
131     frame.getContentPane().add(btnClear);
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, 18));
141     btn00.setBounds(142, 86, 68, 52);
142     frame.getContentPane().add(btn00);
143
144     JButton btnPlus_2511533027 = new JButton("+");
145     btnPlus_2511533027.addActionListener(new ActionListener() {
146         public void actionPerformed(ActionEvent e) {
147             first=Double.parseDouble(textField.getText());
148             textField.setText("");
149             operation="+";
150         }
151     });
152     btnPlus_2511533027.setFont(new Font("Tahoma", Font.BOLD, 18));
153     btnPlus_2511533027.setBounds(209, 86, 68, 52);
154     frame.getContentPane().add(btnPlus_2511533027);
155
156     JButton btn8 = new JButton("8");
157     btn8.addActionListener(new ActionListener() {
158         public void actionPerformed(ActionEvent e) {
159             String number=textField.getText()+btn8.getText();
160             textField.setText(number);
161         }
162     });
163     btn8.setFont(new Font("Tahoma", Font.BOLD, 18));
164     btn8.setBounds(76, 136, 68, 52);
165     frame.getContentPane().add(btn8);
166
167     JButton btn5 = new JButton("5");
168     btn5.addActionListener(new ActionListener() {
169         public void actionPerformed(ActionEvent e) {
170             String number=textField.getText()+btn5.getText();
171             textField.setText(number);
172         }
173     });
174     btn5.setFont(new Font("Tahoma", Font.BOLD, 18));
175     btn5.setBounds(76, 185, 68, 52);
176     frame.getContentPane().add(btn5);

```

```

175     btn5.setBounds(76, 185, 68, 52);
176     frame.getContentPane().add(btn5);
177
178     JButton btn2 = new JButton("2");
179     btn2.addActionListener(new ActionListener() {
180         public void actionPerformed(ActionEvent e) {
181             String number=textField.getText()+btn2.getText();
182             textField.setText(number);
183         }
184     });
185     btn2.setFont(new Font("Tahoma", Font.BOLD, 18));
186     btn2.setBounds(76, 235, 68, 52);
187     frame.getContentPane().add(btn2);
188
189     JButton btnDot = new JButton(".");
190     btnDot.addActionListener(new ActionListener() {
191         public void actionPerformed(ActionEvent e) {
192             String number=textField.getText()+btnDot.getText();
193             textField.setText(number);
194         }
195     });
196     btnDot.setFont(new Font("Tahoma", Font.BOLD, 18));
197     btnDot.setBounds(76, 286, 68, 52);
198     frame.getContentPane().add(btnDot);
199
200     JButton btn9 = new JButton("9");
201     btn9.addActionListener(new ActionListener() {
202         public void actionPerformed(ActionEvent e) {
203             String number=textField.getText()+btn9.getText();
204             textField.setText(number);
205
206             String number=textField.getText()+btn9.getText(),
207             textField.setText(number);
208         }
209     });
210     btn9.setFont(new Font("Tahoma", Font.BOLD, 18));
211     btn9.setBounds(142, 136, 68, 52);
212     frame.getContentPane().add(btn9);
213
214     JButton btnSub_2511533027 = new JButton("-");
215     btnSub_2511533027.addActionListener(new ActionListener() {
216         public void actionPerformed(ActionEvent e) {
217             firstDouble.parseDouble(textField.getText());
218             textField.setText("");
219             operation="-";
220         }
221     });
222     btnSub_2511533027.setFont(new Font("Tahoma", Font.BOLD, 18));
223     btnSub_2511533027.setBounds(209, 136, 68, 52);
224     frame.getContentPane().add(btnSub_2511533027);
225
226     JButton btn6 = new JButton("6");
227     btn6.addActionListener(new ActionListener() {
228         public void actionPerformed(ActionEvent e) {
229             String number=textField.getText()+btn6.getText();
230             textField.setText(number);
231         }
232     });
233     btn6.setFont(new Font("Tahoma", Font.BOLD, 18));
234     btn6.setBounds(142, 185, 68, 52);
235     frame.getContentPane().add(btn6);

```

```

231     btn6.setBounds(142, 185, 68, 52);
232     frame.getContentPane().add(btn6);
233
234     JButton btnMul_2511533027 = new JButton("*");
235     btnMul_2511533027.addActionListener(new ActionListener() {
236         public void actionPerformed(ActionEvent e) {
237             first=Double.parseDouble(textField.getText());
238             textField.setText("");
239             operation="*";
240         }
241     });
242     btnMul_2511533027.setFont(new Font("Tahoma", Font.BOLD, 18));
243     btnMul_2511533027.setBounds(209, 185, 68, 52);
244     frame.getContentPane().add(btnMul_2511533027);
245
246     JButton btn3 = new JButton("3");
247     btn3.addActionListener(new ActionListener() {
248         public void actionPerformed(ActionEvent e) {
249             String number=textField.getText()+btn3.getText();
250             textField.setText(number);
251         }
252     });
253     btn3.setFont(new Font("Tahoma", Font.BOLD, 18));
254     btn3.setBounds(142, 235, 68, 52);
255     frame.getContentPane().add(btn3);
256
257     JButton btnEqual = new JButton "=";
258     btnEqual.addActionListener(new ActionListener() {
259         public void actionPerformed(ActionEvent e) {
260             String answer;
261
262             String answer;
263             second=Double.parseDouble(textField.getText());
264             if(operation=="+")
265             {
266                 result=first+second;
267                 answer=String.format("%.2f",result);
268                 textField.setText(answer);
269             }
270             else if(operation=="-")
271             {
272                 result=first-second;
273                 answer=String.format("%.2f",result);
274                 textField.setText(answer);
275             }
276             else if(operation=="*")
277             {
278                 result=first*second;
279                 answer=String.format("%.2f",result);
280                 textField.setText(answer);
281             }
282             else if(operation=="/")
283             {
284                 result=first/second;
285                 answer=String.format("%.2f",result);
286                 textField.setText(answer);
287             }
288         }
289     });

```

```

288
289     }
290     else if(operation=="%")
291     {
292         result=first%second;
293         answer=String.format("%.2f",result);
294         textField.setText(answer);
295     }
296 }
297 });
298 btnEqual.setFont(new Font("Tahoma", Font.BOLD, 18));
299 btnEqual.setBounds(142, 286, 68, 52);
300 frame.getContentPane().add(btnEqual);
301
302 JButton btnDivide_2511533027 = new JButton("/");
303 btnDivide_2511533027.addActionListener(new ActionListener() {
304     public void actionPerformed(ActionEvent e) {
305         first=Double.parseDouble(textField.getText());
306         textField.setText("");
307         operation="/";
308     }
309 });
310 btnDivide_2511533027.setFont(new Font("Tahoma", Font.BOLD, 18));
311 btnDivide_2511533027.setBounds(209, 235, 68, 52);
312 frame.getContentPane().add(btnDivide_2511533027);
313
314 JButton btnPercent_2511533027 = new JButton("%");
315 btnPercent_2511533027.addActionListener(new ActionListener() {
316     public void actionPerformed(ActionEvent e) {
317         first=Double.parseDouble(textField.getText());
318         textField.setText("");
319         operation="%";
320     }
321 });
322 btnPercent_2511533027.setFont(new Font("Tahoma", Font.BOLD, 18));
323 btnPercent_2511533027.setBounds(209, 286, 68, 52);
324 frame.getContentPane().add(btnPercent_2511533027);
325
326 }
327
328 }
329

```

Pada program ini, kode yang saya tambahkan sendiri yaitu baris 65-73, 82-83, 93-94, 104-105, 115-116, 125-126, 136-137, 147-150, 159-160, 170-171, 181-182, 192-193, 214-217, 203-204, 226-227, 237-240, 249-250, 306-309, 318-321.

Untuk tombol angka 0-9, 00, dan Dot (.), tombol angka berfungsi untuk memasukkan nilai numerik ke dalam layar kalkulator (textField). Ketika salah satu tombol angka ditekan, event ActionListener akan dijalankan. Program mengambil teks yang sudah ada pada layar, kemudian menambahkan karakter angka baru sesuai tombol yang ditekan. Misalnya, ketika pengguna menekan tombol “7”, program akan menggabungkan isi textField sebelumnya dengan teks “7”. Proses ini membuat angka dapat disusun secara bertahap dari kiri ke kanan, sebagaimana kalkulator pada umumnya.

Tombol “00” bekerja dengan cara yang sama, hanya saja menambahkan dua digit angka sekaligus sehingga mempermudah input angka besar seperti 1000 atau 5000. Selain itu, tombol titik (.) digunakan untuk memasukkan angka pecahan. Ketika tombol titik ditekan, karakter “.” akan ditambahkan ke textField sehingga pengguna dapat memasukkan bilangan desimal, misalnya 12.45. Dalam implementasi dasar, tombol titik belum memiliki pembatasan, sehingga pengguna masih bisa memasukkan lebih dari satu titik dalam satu angka, namun secara fungsional tombol ini tetap mendukung operasi desimal.

Untuk tombol Bakspace, tombol backspace digunakan untuk menghapus satu karakter terakhir di layar kalkulator. Fungsinya sama seperti tombol backspace pada keyboard komputer. Ketika tombol ini ditekan, program mengambil teks yang saat ini muncul pada textField, kemudian menggunakan objek StringBuilder untuk menghapus karakter pada posisi paling akhir. Hasil perubahan ini kemudian ditampilkan kembali ke layar.

Mekanisme ini sangat berguna ketika pengguna salah memasukkan angka dan hanya ingin menghapus sebagian input tanpa harus menghapus seluruh isi layar. Tombol backspace juga dilengkapi dengan pengecekan panjang teks, sehingga operasi penghapusan hanya dilakukan ketika teks tidak kosong. Jika textField kosong, tombol backspace tidak melakukan apa pun agar program tetap aman dari error seperti StringIndexOutOfBoundsException.

Untuk tombol Clear, tombol clear bertugas untuk menghapus seluruh isi layar kalkulator. Ketika tombol ini ditekan, program mengosongkan textField dengan memberikan nilai kosong (empty string). Dengan demikian, seluruh input sebelumnya hilang dan layar kembali ke kondisi awal. Tombol clear biasanya digunakan ketika pengguna ingin memulai perhitungan baru dari awal tanpa dipengaruhi oleh angka atau operasi sebelumnya.

Pada implementasi saat ini, tombol clear hanya membersihkan bagian tampilan (textField), namun dalam penerapannya tombol ini juga dapat diperluas untuk mereset variabel internal seperti first, second, result, dan operation agar perhitungan benar-benar dimulai dari kondisi nol. Meskipun demikian, fungsi utamanya tetap sebagai penghapus cepat seluruh input.

Terakhir, tombol operator aritmatika, tombol operator aritmatika digunakan untuk menentukan jenis operasi yang akan dilakukan oleh kalkulator. Ketika salah satu operator ditekan, program pertama-tama membaca angka yang sedang tampil pada

layar dan menyimpannya ke dalam variabel first. Angka ini merupakan operand pertama pada operasi. Setelah itu, textField dikosongkan agar pengguna dapat memasukkan operand kedua. Pada saat yang sama, jenis operator yang dipilih disimpan ke dalam variabel operation dalam bentuk string, misalnya “+”, “-”, “*”, “/”, atau “%”.

Proses perhitungan baru dilakukan ketika tombol “=” ditekan. Pada tahap ini, program mengambil input terbaru dari textField dan menyimpannya sebagai second. Program kemudian mengevaluasi isi variabel operation untuk menentukan operasi apa yang harus dilakukan. Jika operator yang disimpan adalah “+”, maka kalkulator menjumlahkan first dan second. Jika operator adalah “-”, maka dilakukan pengurangan, dan seterusnya untuk operasi perkalian, pembagian, dan modulus. Setelah hasil dihitung, program memformat angka menjadi dua digit desimal dan menampilkannya kembali di textField.

2.2 Analisis Hasil dan Pembahasan

1. Tombol Angka 0-9, 00, dan Dot(.)
 - a) Hasil: Semua tombol angka bekerja dengan baik. Ketika ditekan, angka langsung muncul pada layar. Tombol “00” menambahkan dua angka nol sekaligus, dan tombol titik dapat digunakan untuk membuat angka desimal.
 - b) Analisis: Setiap tombol angka akan menambahkan teks baru ke layar dengan cara mengambil isi sebelumnya, kemudian menambah angka yang ditekan. Ini sesuai dengan cara kerja kalkulator pada umumnya. Tombol titik juga dapat dipakai berulang kali karena belum ada pembatasan, namun untuk kebutuhan dasar praktikum, tombol ini sudah bekerja dengan baik.
2. Tombol Backspace
 - a) Hasil: Tombol backspace dapat menghapus satu karakter terakhir di layar. Jika layar kosong, tombol ini tidak menyebabkan error.
 - b) Analisis: Program mengambil teks pada layar, lalu menghapus huruf terakhir menggunakan StringBuilder. Fitur ini membantu memperbaiki kesalahan input tanpa harus menghapus semua angka. Pengecekan panjang teks juga sudah benar, sehingga lebih aman digunakan.
3. Tombol Clear
 - a) Hasil: Tombol clear menghapus seluruh isi layar dan mengembalikan tampilan menjadi kosong.

- b) Analisis: Clear bekerja dengan memberikan string kosong pada textField. Dengan begitu, pengguna bisa langsung memulai perhitungan baru. Walaupun variabel internal seperti first dan operation tidak ikut di-reset, fungsi clear tetap berjalan sesuai kebutuhan praktikum dasar.
4. Tombol Operator Aritmatika (+, -, *, /, %)
- Hasil: Semua tombol operator dapat menyimpan angka pertama, lalu menunggu angka kedua. Setelah tombol “=” ditekan, hasil perhitungan tampil dengan benar dalam dua angka desimal.
 - Analisis: Ketika operator ditekan, angka pertama disimpan sebagai variabel first, dan operator disimpan dalam variabel operation. Setelah pengguna memasukkan angka kedua dan menekan “=”, program melakukan perhitungan sesuai jenis operator. Cara kerja ini sudah sesuai dengan konsep kalkulator sederhana.
5. Logika Pemrograman dan Event Handling
- Hasil: Setiap tombol merespons dengan cepat dan langsung menampilkan perubahan pada layar.
 - Analisis: Program menggunakan konsep event-driven, yaitu setiap tombol memiliki ActionListener yang berjalan ketika tombol ditekan. Penyimpanan angka pertama, operator, angka kedua, dan hasil sudah sesuai alurnya. Hasil akhir yang ditampilkan juga sudah diformat sehingga tampil lebih rapi.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa program kalkulator sederhana menggunakan Java Swing dapat berjalan dengan baik dan mampu melakukan operasi aritmatika dasar. Mahasiswa dapat memahami bagaimana cara membuat antarmuka GUI menggunakan komponen-komponen seperti JFrame, JButton, dan JTextField. Selain itu, penggunaan ActionListener juga memberikan gambaran kepada mahasiswa tentang cara kerja event pada program berbasis GUI.

Fungsi-fungsi utama seperti tombol angka, tombol operator, tombol backspace, tombol clear, dan tombol sama dengan dapat bekerja sesuai dengan tujuan praktikum. Program juga memperlihatkan bagaimana sebuah aplikasi menyimpan sementara nilai yang dimasukkan pengguna, lalu memprosesnya ketika operasi dijalankan. Dari praktikum ini, mahasiswa dapat memahami alur dasar pembuatan aplikasi GUI serta mampu menghubungkan antara tampilan visual dengan logika pemrograman.

Secara keseluruhan, praktikum ini membantu mahasiswa mengenal lebih jauh pemrograman Java, khususnya pada bagian antarmuka grafis dan event handling yang nantinya menjadi dasar penting untuk membangun aplikasi yang lebih kompleks.

DAFTAR PUSTAKA

- [1] Wahyudi. *Materi Praktikum Pemrograman Java – GUI (Java Swing)*. Program Studi Informatika, Universitas Andalas, 2025.
- [2] Oracle. (2024). *The Java™ Tutorials – Creating a GUI With Swing*.
<https://docs.oracle.com/javase/tutorial/uiswing/>
- [3] Deitel, P., & Deitel, H. (2020). *Java: How to Program (11th ed.)*. Pearson.
- [4] Schildt, H. (2021). *Java: The Complete Reference (12th ed.)*. McGraw-Hill Education.
- [5] YouTube — “*GUI Calculator in Java using Eclipse IDE*”, diakses pada 2025.
<https://www.youtube.com/watch?v=mZVD5OssHQM>