

LAPORAN PRAKTIKUM
AGORITMA PEMROGRAMAN

“PEMROGRAMAN GUI 2”

disusun Oleh:

SOFIAN ARBA’I

2511533029

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T

Asisten Praktikum:

JOVANTRI IMMANUEL GULO



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur kehadiran Allah SWT karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan Laporan Praktikum Pekan 9 dengan judul "*Pemrograman GUI 2*" ini dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum serta sebagai sarana untuk meningkatkan pemahaman mahasiswa mengenai konsep dan penerapan *Graphical User Interface (GUI)* dalam pemrograman.

Dalam praktikum pekan ini, saya mempelajari lebih lanjut mengenai pemrograman GUI ini. Pada pekan ini, saya diarahkan untuk membuat aplikasi sederhana yaitu kalkulator. Saya mengikuti intruksi untuk membuat aplikasi berdasarkan link yang diberikan oleh dosen. Pembuatan aplikasi ini membuat saya menjadi lebih memahami tentang pemrograman GUI.

Penyusunan laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Dosen pengampu mata kuliah Praktikum Algoritma dan Pemrograman yang telah memberikan ilmu, arahan, serta bimbingannya.
2. Asisten praktikum yang telah membimbing dan memberikan penjelasan selama kegiatan praktikum berlangsung.

Saya meminta maaf jika banyak terdapat kesalahan pada penulisan laporan ini. Oleh karena itu, saya menerima kritik dan saran yang membangun demi perbaikan di masa mendatang.

Padang, 25 November 2025

Sofian Arba'i

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat	2
BAB II PEMBAHASAN	3
2.1 Pengertian GUI.....	3
2.2 Desain dan Komponen pada Aplikasi Kalkulator	3
2.3 Aplikasi Penggunaan Pemrograman GUI.....	5
2.3.1 Tampilan Aplikasi Kalkulator	5
2.3.2 Kode Program Aplikasi Kalkulator	6
2.3.3 Kode Program Aplikasi Kalkulator	7
2.3.4 Kode Program Aplikasi Kalkulator	8
2.3.5 Kode Program Aplikasi Kalkulator	9
BAB III PENUTUP	11
3.1 Kesimpulan	11
3.2 Saran	11
DAFTAR PUSTAKA.....	12

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi perangkat lunak semakin menuntut aplikasi yang tidak hanya mampu memproses data, tetapi juga memiliki antarmuka yang interaktif dan mudah digunakan. Java menyediakan fasilitas untuk membangun antarmuka pengguna melalui toolkit seperti AWT dan Swing. Pada praktikum pekan sebelumnya, mahasiswa telah mempelajari dasar-dasar pembuatan GUI menggunakan komponen-komponen Swing. Untuk memperdalam pemahaman tersebut, praktikum pekan 9 berfokus pada implementasi lebih lanjut melalui pembuatan aplikasi kalkulator berbasis Java GUI.

Aplikasi kalkulator dipilih sebagai studi kasus karena memiliki struktur antarmuka yang umum ditemui dan memerlukan interaksi langsung melalui tombol serta input data. Melalui pembuatan kalkulator, mahasiswa tidak hanya belajar menempatkan komponen GUI, tetapi juga memahami cara mengelola event listener, menangani input, serta menghubungkan antarmuka dengan logika program. Dengan demikian, praktikum ini menjadi langkah penting dalam mengasah keterampilan dalam pemrograman event-driven dan pembangunan aplikasi GUI yang fungsional.

1.2 Tujuan

Tujuan pelaksanaan praktikum pekan 9 yaitu pemrograman GUI 2 sebagai berikut.

1. Memahami penerapan lanjutan komponen-komponen Swing dalam pembuatan aplikasi GUI yang lebih kompleks.
2. Melatih kemampuan mengatur layout dan posisi komponen, seperti tombol angka, operator, dan tampilan hasil pada kalkulator.
3. Menghubungkan antarmuka dengan logika perhitungan sehingga aplikasi kalkulator dapat melakukan operasi aritmatika dasar.

4. Mengembangkan aplikasi GUI sederhana fungsional sebagai hasil praktik pemrograman Java secara langsung.

1.3 Manfaat

Manfaat yang didapatkan setelah melaksanakan praktikum pekan 9 tentang pemrograman GUI 2 sebagai berikut.

1. Menambah keterampilan dalam membangun aplikasi GUI nyata, bukan sekadar komponen terpisah.
2. Meningkatkan pemahaman tentang alur kerja aplikasi event-driven, khususnya dalam menangani aksi pada tombol kalkulator.
3. Membiasakan diri menggunakan komponen Swing secara efisien, termasuk penggunaan container dan layout manager.
4. Menjadi dasar yang kuat untuk mengembangkan aplikasi Java GUI yang lebih kompleks, seperti aplikasi kasir, form data, atau aplikasi desktop sederhana lainnya.

BAB II

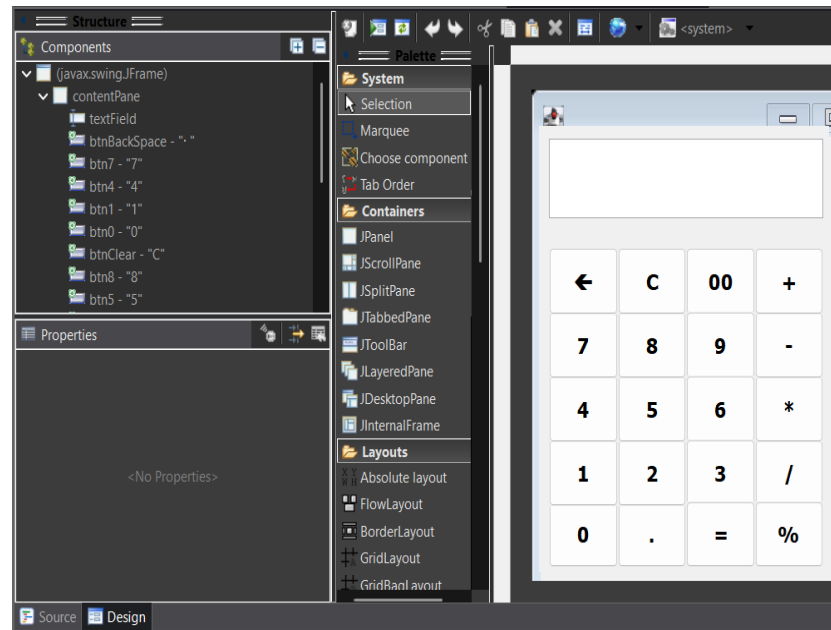
PEMBAHASAN

2.1 Pengertian GUI

GUI atau Graphical User Interface merupakan sistem yang memungkinkan para pengguna untuk melakukan interaksi dengan komputer. GUI merupakan sebuah desain antar muka pada komputer, selain GUI ada pula CLI atau Command Line Interface, yang berbasis text dimana para pengguna dapat melakukan interaksi dengan komputer melalui perintah text atau Command Line. GUI merupakan antarmuka pada system yang menggunakan menu grafis. Menu grafis ini maksudnya terdapat tampilan yang lebih ditekankan untuk membuat system yang user-friendly agar para pengguna lebih nyaman menggunakannya. Secara sederhana, GUI adalah suatu media virtual yang dapat membuat pengguna memberikan perintah tertentu pada komputer tanpa mengetik perintah atau Command. Dengan GUI, perintah dapat dikonversi menjadi ikon dalam layar monitor yang dapat diklik untuk memulai fungsinya.

2.2 Desain dan Komponen pada Aplikasi Kalkulator

Berikut adalah tampilan desain untuk aplikasi kalkulator sederhana yang saya buat pada ketika praktikum.



Gambar 2.1

Gambar di atas merupakan tampilan desain dari aplikasi kalkulator. Pada bagian *Components* (Structure) yang terlihat di sisi kiri, ditampilkan susunan atau hierarki seluruh komponen GUI yang telah ditambahkan ke dalam *contentPane* dari *JFrame*. Di dalamnya terlihat berbagai tombol seperti *btn7*, *btn4*, *btn1*, *btn0*, *btnClear*, dan komponen lainnya yang membentuk layout kalkulator. Bagian ini membantu developer memahami struktur keseluruhan interface, seperti komponen mana yang berada di dalam panel tertentu, serta memudahkan dalam memilih, menghapus, atau memindahkan komponen. Di bawahnya terdapat panel *Properties*, yang berfungsi menampilkan dan mengatur berbagai pengaturan dari komponen yang sedang dipilih, seperti teks, ukuran huruf, warna, alignment, nama variabel, hingga event listener terkait. Melalui *Properties*, setiap komponen dapat dikustomisasi sesuai kebutuhan tampilan maupun fungsionalitas program. Sementara itu, pada bagian kanan terdapat *Palette*, yaitu kumpulan komponen-komponen GUI yang dapat digunakan dalam desain, seperti label, text field, button, panel, layout manager, dan berbagai elemen Swing lainnya. Pengguna cukup melakukan *drag and drop* komponen dari *Palette* ke area desain untuk membangun tampilan aplikasi. Ketiga bagian ini bekerja bersama untuk memudahkan proses perancangan GUI: *Palette* menyediakan

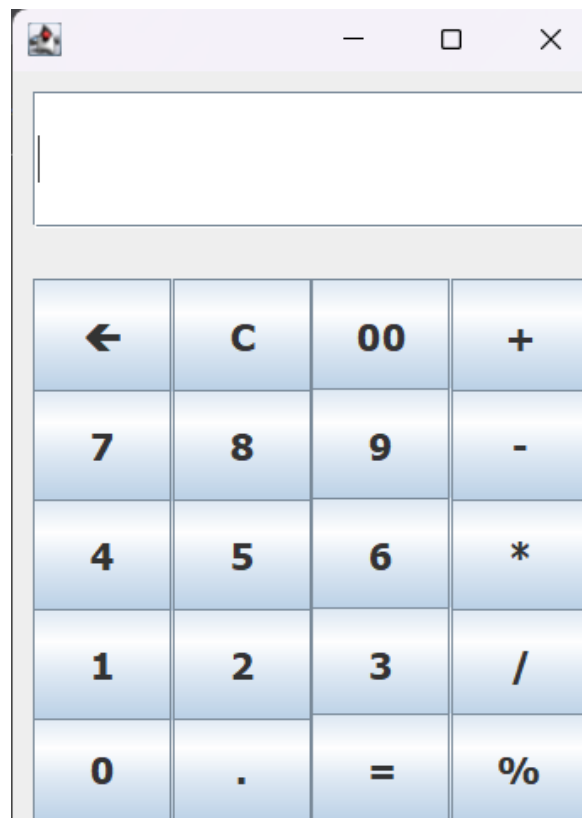
komponen, Structure menampilkan susunannya, dan Properties memungkinkan pengaturan detail setiap komponen.

2.3 Aplikasi Penggunaan Pemrograman GUI

Pada praktikum pekan 9 ini, mahasiswa diminta untuk membuat sebuah aplikasi yang dimiliki oleh semua orang di hpnya yaitu kalkulator. Mahasiswa diminta membuat kalkulator menggunakan pemrograman GUI sesuai dengan video yang diberikan oleh dosen. Berikut adalah penerapan atau aplikasi GUI dalam pembuatan kalkulator.

2.3.1 Tampilan Aplikasi Kalkulator

Berikut adalah tampilan aplikasi kalkulator yang sudah saya buat dan sesuai dengan yang saya tampilkan pada penjelasan sebelumnya.



Gambar 2.2

2.3.2 Kode Program Aplikasi Kalkulator

Berdasarkan penjelasan di atas, pada pekan 9 ini mahasiswa diminta untuk membuat aplikasi kalkulator. Tampilan desain juga telah saya paparkan pada sub bab sebelumnya. Supaya aplikasi kalkulator yang saya buat bisa dijalankan, maka diperlukan beberapa kode program supaya aplikasi bisa dijalankan sebagaimana mestinya. Berikut adalah beberapa kode program untuk menjalankan aplikasi kalkulator yang saya buat.

```

100      JButton btn1 = new JButton("1");
101      btn1.addActionListener(new ActionListener() {
102          public void actionPerformed(ActionEvent e) {
103              String number=textField.getText()+btn1.getText();
104              textField.setText(number);
105          }
106      });
107      btn1.setFont(new Font("Tahoma", Font.BOLD, 18));
108      btn1.setBounds(10, 255, 66, 54);
109      contentPane.add(btn1);

```

Kode Program 2.1

Kode program di atas adalah kode program yang membuat sebuah tombol bernama btn1 yang berlabel “1” menggunakan komponen JButton. Setelah tombol dibuat, ditambahkan sebuah ActionListener, yaitu mekanisme yang membuat tombol merespon ketika ditekan. Di dalam metode actionPerformed, program mengambil teks yang sudah ada di textField kemudian menambahkan teks dari tombol btn1—yaitu angka “1”—ke bagian akhir teks tersebut. Hasil penggabungan ini disimpan dalam variabel number, lalu ditampilkan kembali ke textField menggunakan setText(). Dengan cara ini, setiap kali pengguna menekan tombol angka 1, angka tersebut akan muncul (atau ditambahkan) pada layar kalkulator.

Pada potongan kode program 2.1, kode program tersebut merupakan bagian dari keseluruhan kode program. Kode program di atas pada intinya adalah membuat sebuah tombol yang ketika di tekan

akan menampilkan angka 1 di layar aplikasi kalkulator. Kode program di atas juga untuk berlaku pada kode program selanjutnya yaitu 2-9, serta sama dengan kode program untuk menampilkan angka 0 dan 00 pada layar kalkulator.

2.3.3 Kode Program Aplikasi Kalkulator

Kode program selanjutnya untuk aplikasi kalkulator yang saya buat sebagai berikut.

```

62      JButton btnBackSpace = new JButton("\uF0E7");
63      btnBackSpace.addActionListener(new ActionListener() {
64          public void actionPerformed(ActionEvent e) {
65              String backSpace=null;
66              if (textField.getText().length()>0) {
67                  StringBuilder str=new StringBuilder(textField.getText());
68                  str.deleteCharAt(textField.getText().length()-1);
69                  backSpace=str.toString();
70                  textField.setText(backSpace);

```

Kode Program 2.2

Kode program di atas merupakan kode program yang menggunakan button supaya ketikan tombol di tekan akan melakukan suatu perintah atau aksi. Di dalam metode actionPerformed, pertama-tama dibuat variabel backSpace yang digunakan untuk menyimpan hasil akhir setelah penghapusan karakter. Program kemudian memeriksa apakah textField memiliki isi dengan mengecek panjang teksnya. Selanjutnya, deleteCharAt() digunakan untuk menghapus karakter terakhir, yaitu karakter pada indeks length()-1. Setelah karakter terakhir dihapus, hasilnya dikonversi kembali menjadi string dan disimpan pada variabel backSpace. Terakhir, nilai ini ditampilkan kembali pada textField menggunakan setText(), sehingga efeknya adalah menghapus satu karakter terakhir setiap kali tombol backspace ditekan. Berikut adalah lanjutan kode program untuk aplikasi kalkulator.

```

122     JButton btnClear = new JButton("C");
123     btnClear.addActionListener(new ActionListener() {
124         public void actionPerformed(ActionEvent e) {
125             textField.setText(null);
126         }
127     });
128     btnClear.setFont(new Font("Tahoma", Font.BOLD, 18));
129     btnClear.setBounds(76, 99, 66, 54);
130     contentPane.add(btnClear);
131

```

Kode Program 2.3

Potongan kode program pada gambar yaitu sebagai Tombol ini berfungsi untuk menghapus seluruh isi tampilan pada textField. Sama seperti fungsi tombol c pada kalkulator yang sering kita temui. Inti dari kode program pada gambar adalah menghapus seluruh angka yang ada di layar kalkulator sehingga pada layar kalkulator menjadi kosong.

2.3.4 Kode Program Aplikasi Kalkulator

Kode program selanjutnya adalah kode program untuk operasi yang ada pada kalkulator seperti penjumlahan, pengurangan, perkalian, pembagian dan sisa hasil bagi.

```

268     JButton btnPlus_2511533029 = new JButton("+");
269     btnPlus_2511533029.addActionListener(new ActionListener() {
270         public void actionPerformed(ActionEvent e) {
271             first=Double.parseDouble(textField.getText());
272             textField.setText("");
273             operation="+";
274         }
275     });
276     btnPlus_2511533029.setFont(new Font("Tahoma", Font.BOLD, 18));
277     btnPlus_2511533029.setBounds(208, 99, 66, 54);
278     contentPane.add(btnPlus_2511533029);
279
280     JButton btnMin_2511533029 = new JButton("-");
281     btnMin_2511533029.addActionListener(new ActionListener() {
282         public void actionPerformed(ActionEvent e) {
283             first=Double.parseDouble(textField.getText());
284             textField.setText("");
285             operation="-";
286         }
287     });
288     btnMin_2511533029.setFont(new Font("Tahoma", Font.BOLD, 18));
289     btnMin_2511533029.setBounds(208, 151, 66, 54);
290     contentPane.add(btnMin_2511533029);
291

```

Kode Program 2.4

Pada kode program ini mahasiswa diminta untuk menambahkan NIM untuk setiap nama operator yang ada di kalkulator. Kode program yang ada pada gambar adalah kode program untuk operator penjumlahan dan pengurangan. Kode program untuk operasi perkalian, pembagian, sisa bagi hampir sama seperti kode program penjumlahan dan pengurangan, hanya saja nama button dan tanda operasi harus sesuai dengan operasi yang kita buat.

2.3.5 Kode Program Aplikasi Kalkulator

Pada sub bab ini kode program yang akan dibahas merupakan kode program yang bisa dikatakan kode program yang cukup penting yaitu kode program untuk tombol sama dengan atau *equals*. Berikut adalah kode programnya.

```

176      JButton btnSamadengan_2511533029 = new JButton("=");
177      btnSamadengan_2511533029.addActionListener(new ActionListener() {
178          public void actionPerformed(ActionEvent e) {
179              String answer;
180              second=Double.parseDouble(textField.getText());
181              if (operation=="+") {
182                  result=first+second;
183                  answer=String.format("%.2f" , result);
184                  textField.setText(answer);
185              }
186              else if (operation=="-") {
187                  result=first-second;
188                  answer=String.format("%.2f" , result);
189                  textField.setText(answer);
190              }
191              else if (operation=="*") {
192                  result=first*second;
193                  answer=String.format("%.2f" , result);
194                  textField.setText(answer);
195              }
196              else if (operation=="/") {
197                  result=first/second;
198                  answer=String.format("%.2f" , result);
199                  textField.setText(answer);}
200
201              else if (operation=="%") {
202                  result=first%second;
203                  answer=String.format("%.2f" , result);
204                  textField.setText(answer);
205              }

```

Kode Program 2.5

Bagian kode program pada gambar di atas menangani proses perhitungan utama ketika pengguna menekan tombol sama dengan (=) pada kalkulator. Ketika tombol tersebut diklik, sebuah ActionListener dijalankan. Pertama, nilai kedua (second) diambil dari isi textField dan dikonversi menjadi tipe double menggunakan Double.parseDouble(). Setelah itu, program memeriksa jenis operasi yang disimpan sebelumnya dalam variabel operation, seperti penjumlahan, pengurangan, perkalian, pembagian, atau modulus. Pemeriksaan dilakukan menggunakan rangkaian if-else if, di mana setiap kondisi membandingkan nilai operation dengan simbol operator tertentu. Jika operator yang digunakan cocok, maka program melakukan perhitungan dengan menggunakan variabel first dan second, lalu menyimpan hasilnya ke dalam variabel result. Terakhir, hasil perhitungan ini ditampilkan kembali ke textField melalui setText(answer), sehingga pengguna dapat melihat hasil akhirnya pada layar kalkulator. Kode program terakhir sebagai berikut.

```

165         JButton btnDot = new JButton(".");
166         btnDot.addActionListener(new ActionListener() {
167             public void actionPerformed(ActionEvent e) {
168                 String nomor=textField.getText()+btnDot.getText();
169                 textField.setText(nomor);
170             }
171         });
172         btnDot.setFont(new Font("Tahoma", Font.BOLD, 18));
173         btnDot.setBounds(76, 305, 66, 54);
174         contentPane.add(btnDot);

```

Kode Program 2.6

Kode program di atas adalah kode program yang akan menambahkan “.” atau “,” sesuai dengan bahasa pada komputer masing-masing untuk bilangan desimal yang akan melakukan operator aritmatika.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum pekan 9 dengan materi Pemrograman GUI 2, dapat disimpulkan bahwa pembuatan aplikasi kalkulator berbasis Java Swing memberikan pemahaman yang lebih mendalam mengenai cara kerja antarmuka grafis dan pemrograman berbasis event. Melalui praktik langsung, mahasiswa mampu menerapkan berbagai komponen GUI seperti JButton, JTextField, panel, serta memahami cara mengatur tata letak dan properti komponen. Selain itu, mahasiswa juga belajar menghubungkan antarmuka dengan logika program, seperti proses penjumlahan, pengurangan, perkalian, pembagian, serta operasi modulus. Pembuatan aplikasi kalkulator ini membuktikan bahwa mahasiswa telah memahami konsep dasar pemrograman event-driven, penanganan input pengguna, serta teknik manipulasi data melalui aksi pada tombol. Secara keseluruhan, praktikum ini memberikan pengalaman nyata dalam membangun aplikasi GUI sederhana yang fungsional dan interaktif.

3.2 Saran

Pada praktikum selanjutnya, disarankan agar mahasiswa lebih memperhatikan struktur penulisan kode agar program lebih rapi dan mudah dipahami. Selain itu, akan lebih baik jika eksperimen lanjutan dilakukan dengan menambahkan fitur tambahan pada aplikasi kalkulator, seperti tombol Clear Entry (CE), operasi pangkat, atau riwayat perhitungan untuk meningkatkan kapabilitas aplikasi. Bagi pengajar, penyediaan contoh studi kasus tambahan atau penjelasan tentang penggunaan layout manager yang lebih kompleks dapat membantu mahasiswa memahami perancangan GUI yang lebih profesional. Dengan terus berlatih dan mengeksplorasi fitur Java Swing lainnya, mahasiswa akan semakin terbiasa dalam membangun aplikasi desktop yang lebih baik dan efektif.

DAFTAR PUSTAKA

- [1] BANGWOK.ID, “Jurnal Java AWT dan Swing,”. [Daring]. Tersedia pada:
[Jurnal Java AWT dan Swing - Bangwok.id](#) [Diakses: 25-Nov-2025].