

LAPORAN PRATIKUM PEKAN 9

ALGORITMA DAN PEMROGRAMAN

“Pemograman GUI 2”



Dosen Pengampu:

DR. Wahyudi, S.T., M.T.

Asisten Lab:

Muhammad Zaki Al Hafiz

Disusun Oleh:

Hamdi Sidqi Alifi

2511531009

Fakultas Teknologi Informasi

Departemen Informatika

Universitas Andalas

2025

KATA PENGANTAR

Puji dan syukur senantiasa penulis panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan ini guna memenuhi laporan praktikum mata kuliah Algoritma Pemrograman, dengan judul: “Pemograman GUI 2”. Laporan ini berisikan hasil analisis dari praktikum pemograman java penulis di Praktikum Kelas D yang dilaksanakan pada hari Rabu, tanggal 26 November 2025.

Pada kesempatan ini, Penulis ingin mengucapkan terima kasih kepada Bapak DR. Wahyudi. .S.T.M.T, dosen pengampu mata kuliah Algoritma Pemrograman, yang telah memberikan tugas ini. Penulis juga ingin mengucapkan terima kasih kepada pihak-pihak yang telah mendukung serta membantu penulis dalam penyelesaian laporan praktikum Algoritma Pemrograman.

Penulis sadar bahwa laporan ini masih memiliki beberapa kekurangan dikarenakan terbatasnya pengalaman dan pengetahuan yang penulis miliki. Oleh karena itu, penulis harap adanya bentuk saran dan kritik yang membangun dari berbagai pihak.

Akhir kata, penulis berharap laporan ini dapat bermanfaat bagi perkembangan dunia pendidikan.

Padang, 29 November 2025

Hamdi Sidqi Alifi
2511531009

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat Pratikum.....	1
BAB II PEMBAHASAN	2
2.1 Teori	2
2.2 Kode Pemrograman.....	5
BAB III KESIMPULAN	9
3.1 Kesimpulan	9
3.2 Saran.....	9
BAB IV DAFTAR PUSTAKA.....	3

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pengembangan program menggunakan WindowBuilder menawarkan pendekatan yang sangat efisien dalam merancang *Graphical User Interface* (GUI) berbasis Java, menjadikan pembuatan aplikasi fungsional sederhana seperti kalkulator jauh lebih intuitif dan cepat. Dengan fitur *design view* yang mendukung mekanisme *drag-and-drop*, pengembang dapat menyusun komponen visual seperti tombol angka, panel layar, dan label operasional secara presisi tanpa harus menuliskan kode tata letak yang rumit secara manual dari nol. Otomatisasi ini tidak hanya memangkas waktu pengembangan secara signifikan, tetapi juga memungkinkan pemrogram untuk mengalihkan fokus utama mereka pada penulisan logika aritmatika di balik aplikasi, memastikan bahwa program tidak hanya terlihat rapi secara visual tetapi juga berjalan akurat secara fungsional.

1.2 Tujuan

1. Menjelaskan mengenai penggunaan plugin WindowBuilder pada perangkat IDE seperti Eclipse.
2. Menjelaskan cara mengembangkan sebuah aplikasi yang dapat memproses algoritma dasar dengan plugin WindowBuilder.

1.3 Manfaat Pratikum

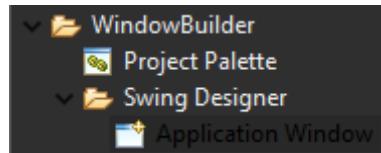
1. Memahami konsep penggunaan Application Window pada WindowBuilder Swing Designer
2. Mampu mengembangkan aplikasi algoritma sederhana dengan WindowBuilder.

BAB II

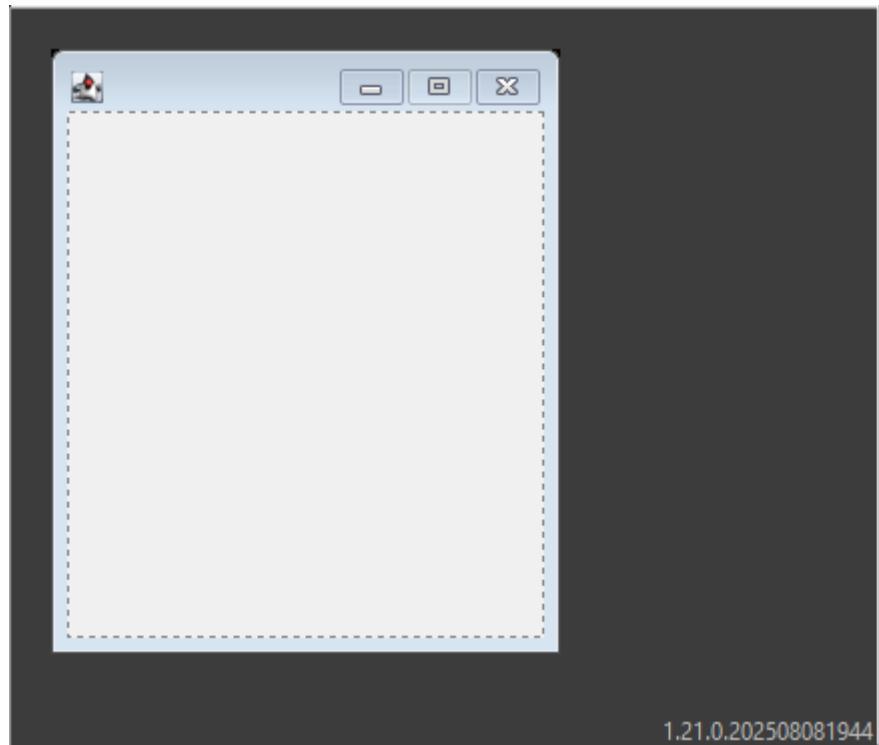
PEMBAHASAN

2.1 Teori

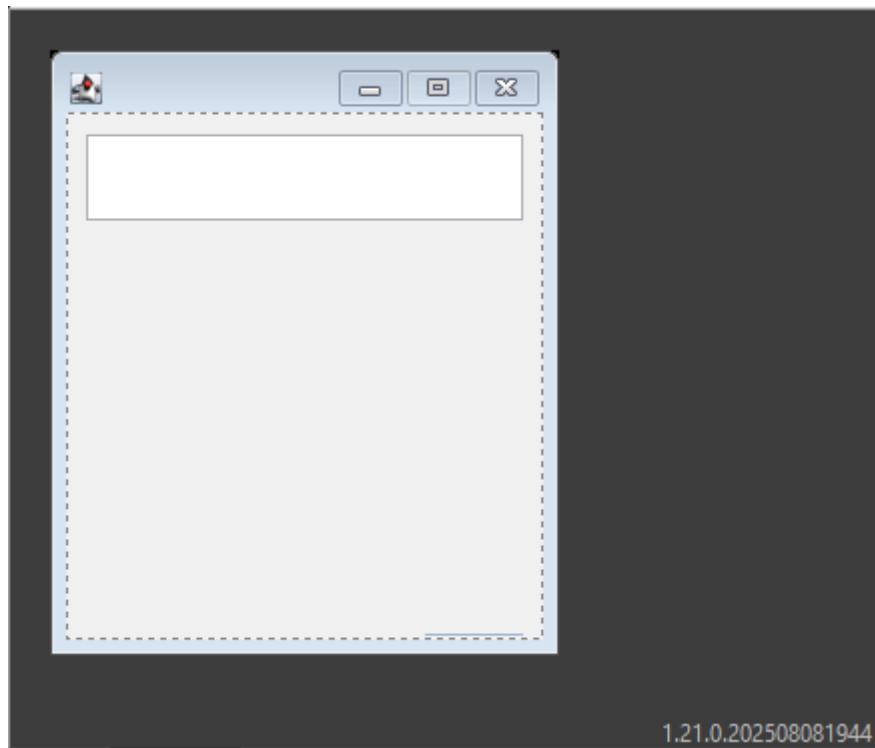
Selain JFrame, kita juga bisa menggunakan Application Window untuk membuat sebuah aplikasi dengan GUI.



Selayaknya JFrame, kita bisa mengubah berbagai hal dengan Application Window.

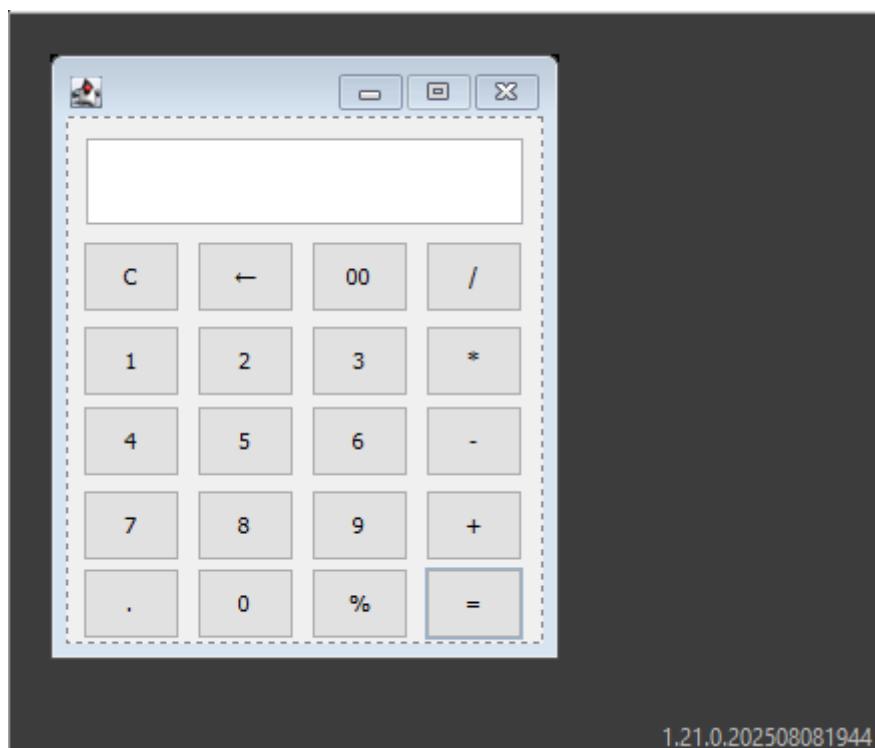


Tak lupa pula untuk menerapkan absolute layout pada GUI kita



1.21.0.202508081944

Setelah itu kita menambahkan JTextField untuk tampilan hasil pada kalkulator nantinya.

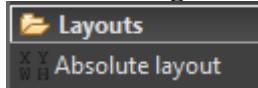


1.21.0.202508081944

Kemudian tambahkan JButton untuk setiap komponen yang kita gunakan dan beri variabel

```
clear - "C"
delete - "←"
num00 - "00"
div - "/"
num1 - "1"
num2 - "2"
num3 - "3"
mul - "*"
num4 - "4"
num5 - "5"
num6 - "6"
sub - "-"
num7 - "7"
num8 - "8"
num9 - "9"
add - "+"
dot - "."
num0 - "0"
percent - "%"
equals - "="
```

2.2 Kode Pemrograman



Pada langkah pertama, penulis menerapkan absolute layout pada window, hal ini bertujuan agar komponen-komponen yang kita pasang berada pada posisi dan ukuran yang tetap walaupun jendela aplikasinya kita tarik dan regangkan saat program di compile.

```
JButton num1 = new JButton("1");
num1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if (isResultDisplayed) {
            textField.setText("");
            isResultDisplayed = false;
        }
        String number = textField.getText()+num1.getText();
        textField.setText(number);
    }
});
num1.setBounds(8, 104, 49, 36);
frame.getContentPane().add(num1);
```

Double Click pada nomor 1 untuk beralih ke source code dan menciptakan Action Listener. Line tersebut berfungsi untuk menampilkan tombol tombol yang ditekan nantinya. Algoritma ini bekerja pada tombol 0 hingga 9 beserta tombol koma (Standar Amerika Serikat (.)) dan double nol (00).

Kemudian untuk menghapus keseluruhan unit pada tampilan JTextField, kita menuliskan program berikut.

```
JButton clear = new JButton("c");
clear.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        textField.setText(null);
        isResultDisplayed = false;
    }
});
clear.setBounds(8, 62, 49, 36);
frame.getContentPane().add(clear);
```

Pada line yang mengandung setText, terdapat null, berfungsi sebagai merewrite nilai menjadi kosong.

Selain button Clear, disini kita juga membutuhkan button delete/backspace.

```

 JButton delete = new JButton("⌫");
 delete.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
         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);
         }
     }
 });
 delete.setBounds(65, 62, 49, 36);
 frame.getContentPane().add(delete);

```

Program button delete bekerja dengan menginisialisasikan string yang diberi variabel backSpace yang memiliki nilai tetap *null*. Program akan memeriksa terlebih dahulu apakah pada textField terdapat bilangan atau tidak, jika ada maka program akan menghapus unit paling terakhir pada textField. Namun, jika tidak ada unit yang tampil pada textField maka program tidak melakukan apapun.

```

263     JButton add = new JButton("+");
264     add.addActionListener(new ActionListener() {
265         public void actionPerformed(ActionEvent e) {
266             first = Double.parseDouble(textField.getText().replace(",","."));
267             textField.setText("");
268             operation = "+";
269             isResultDisplayed = false;
270         }
271     });
272     add.setBounds(179, 186, 49, 36);
273     frame.getContentPane().add(add);

```

Logika utama yang dieksekusi saat tombol operator (+ dalam contoh ini) ditekan adalah menyimpan angka yang saat ini ditampilkan dan mengatur status program untuk input berikutnya. Langkah pertama adalah mengambil nilai dari layar (text field), mengubahnya menjadi tipe data numerik (Double), dan menyimpannya sebagai angka pertama (first) yang akan dioperasikan. Setelah angka tersebut berhasil dicatat, layar kemudian dikosongkan untuk memungkinkan pengguna memasukkan angka kedua. Secara simultan, program mencatat jenis operasi yang dipilih, yaitu +, dalam variabel operation. Proses ini diakhiri dengan pengaturan flag isResultDisplayed menjadi false, yang menandakan bahwa status program telah direset dan siap menerima input angka baru.

Logika dasar ini yaitu mengambil dan menyimpan angka pertama, mengosongkan layar, dan menetapkan operasi harus direplikasi dan disesuaikan untuk setiap tombol operator lain yang ada pada kalkulator, seperti kurang (-), kali (*), bagi (/) dan sisa bagi/modulus (%).

```
double first;
double second;
double result;
String operation;
String answer;
boolean isResultDisplayed = false;

315      JButton equals = new JButton("=");
316      equals.addActionListener(new ActionListener() {
317          public void actionPerformed(ActionEvent e) {
318              String answer;
319              second = Double.parseDouble(textField.getText().replace(",","."));
320              if (operation.equals("+")) {
321                  result = first + second;
322              } else if (operation.equals("-")) {
323                  result = first - second;
324              } else if (operation.equals("*")) {
325                  result = first * second;
326              } else if (operation.equals("/")) {
327                  result = first / second;
328              } else if (operation.equals("%")) {
329                  result = first % second;
330              } else {
331                  result = second;
332              }
333              answer = String.format("%.2f", result);
334              textField.setText(answer);
335              isResultDisplayed = true;
336          }
337      });
338      equals.setBackground(systemColor.activeCaption);
339      equals.setBounds(179, 225, 49, 36);
340      frame.getContentPane().add>equals);
```

Ketika tombol sama dengan (=) ditekan, fungsi utamanya adalah mengambil angka kedua dan menjalankan perhitungan yang tertunda. Langkah pertama, program segera mengambil nilai yang ada di layar (angka kedua) dan mengubahnya menjadi format numerik yang siap dihitung. Setelah itu, program akan memeriksa operator apa yang sebelumnya disimpan dalam variabel operation (misalnya, +, -, *, /, atau %). Berdasarkan operator yang ditemukan, perhitungan yang sesuai (penjumlahan, pengurangan, perkalian, pembagian, atau sisa bagi) akan dilakukan menggunakan angka pertama yang sudah disimpan sebelumnya (first) dan angka kedua (second) yang baru saja diambil. Hasil dari perhitungan tersebut kemudian disimpan ke dalam variabel result.

Setelah mendapatkan hasil akhir, program melanjutkan ke langkah penyelesaian dan pengaturan status. Nilai result diubah kembali menjadi teks (String) dan diformat secara rapi, memastikan, misalnya, hanya menampilkan dua angka di belakang koma, sebelum akhirnya ditampilkan kembali ke text field (layar kalkulator). Langkah terakhir dan penting adalah menetapkan flag isResultDisplayed menjadi true. Status ini memberitahu program bahwa tampilan yang ada saat ini adalah hasil dari suatu perhitungan, sehingga input angka baru berikutnya harus menghapus hasil tersebut, alih-alih menambahkannya ke hasil sebelumnya, membuat program siap untuk perhitungan baru.

BAB III **KESIMPULAN**

3.1 Kesimpulan

Dengan mengaplikasikan berbagai komponen seperti JButton, JLabel, JTextField, dan JComboBox, saya mampu menciptakan desain aplikasi yang tidak hanya fungsional tetapi juga interaktif.

Penggunaan WindowBuilder sangatlah efisien dalam mendesain GUI. Alat ini memfasilitasi perancangan visual melalui teknik *drag and drop*. Keunggulannya terletak pada kemampuannya untuk memproduksi kode Java yang rapi dan minim dependensi eksternal, ditambah fitur *editing* dua arah. Hal ini menjadikannya instrumen yang ideal untuk kegiatan belajar-mengajar, memungkinkan kami melihat secara simultan korelasi antara tampilan grafis dan struktur kode, sehingga logika di balik setiap komponen menjadi jauh lebih mudah dipahami.

Ditambah dengan fitur Application Window yang dapat diexport dalam bentuk format tertentu (seperti .jar) memungkinkan kita untuk mempublikasi aplikasi yang telah kita kembangkan agar bisa digunakan oleh orang lain.

3.2 Saran

Penulis harap kita sebagai mahasiswa informatika harus dapat memahami dan mampu mengembangkan aplikasi dengan menggunakan windowbuilder.

BAB IV

DAFTAR PUSTAKA

Tech Projects. (2020, April 3). *GUI Calculator in JAVA using eclipse ide / Tech Projects.* YouTube.

<https://www.youtube.com/watch?si=gHWOK4c6lZLLeQg&v=mZVD5OssHQM&feature=youtu.be>