

**LAPORAN PRAKTIKUM**  
**ALGORITMA & PEMROGRAMAN**  
**“Laporan Praktikum Pekan 9”**

Disusun Oleh:

Rizky Fadhlurrahman Almatrria

2511532006

Dosen Pengampu : Dr. Wahyudi, S.T, M.T.

Asisten Praktikum : Muhammad Zaki



**DAPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI**  
**INFORMASI UNIVERSITAS**  
**ANDALAS**

2025

## **KATA PENGANTAR**

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga dapat menyelesaikan laporan praktikum ini dengan baik. Tidak lupa juga saya ucapkan terima kasih kepada bapak Wahyudi. Dr. S.T.M.T sebagai dosen pembimbing dan Muhammad Zaki Al Hafiz yang telah membantu dalam pelaksanaan praktikum ini. Laporan ini saya susun untuk memenuhi tugas praktikum Algoritma dan Pemrograman dari praktikum pertemuan Ke-9 mengenai pemahaman dasar pemrograman bahasa Java menggunakan IDE Eclipse. Praktikum ini bertema implementasi pembuatan Aplikasi dengan user interface menggunakan Window builder by Eclipse IDE. Saya berharap laporan ini dapat memberikan manfaat bagi pembaca dalam mengimplementasi kan pehaman dasar pembuatan UI untuk program Java.

Laporan ini dibuat dengan harapan dapat memberikan pemahaman dasar mengenai penggunaan Bahasa pemrograman Java, khususnya mengenai hal struktur dasar program dan fungsi.

Saya menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu, kritik dan saran sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang 30 November, 2025

Penulis

## Daftar Isi

### Daftar Isi

BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan .....	1
1.3 Manfaat.....	1
BAB II.....	3
PEMBAHASAN .....	3
2.1 Fitur Dasar “Window Builder” Eclipse IDE.....	3
2.1.1 Membuat Project “Window Builder” .....	3
2.1.2 Membuat tampilan UI Program. ....	3
2.2 Coding UI .....	4
2.2.1 Kode program.....	4
2.2.2 Contoh Output .....	6
.....	6
BAB III.....	7
KESIMPULAN .....	7
DAFTAR PUSTAKA .....	8

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Praktikum Pekan 9, Disusun dan dirancang untuk melakukan implementasi penggunaan “Window builder” dengan membuat aplikasi calculator yang bukan hanya UI kosong atau hanya bekerja di console, tapi juga bekerja di program yang di window terpisah dan dapat dengan mudah di gunakan oleh pengguna awam.

### **1.2 Tujuan**

1. Membuat Tampilan UI(*User Interface*) program kalkulator menggunakan “Window Builder”.
2. Melakukan coding agar program kalkulator dapat berjalan.

### **1.3 Manfaat**

1. Mahasiswa dapat Mengetahui dan dapat memanfaatkan fitur dasar dari “Window Builder” Eclipse IDE.
2. Mahasiswa dapat membuat Program calculator yang interaktif dan dapat dengan mudah di mengerti orang awam.

## BAB II

### PEMBAHASAN

#### 2.1 Pembuatan desain UI(*User Interface*) Program calculator

Pada praktikum ini, kita perlu mendesain bagaimana program kalkulator ini seharusnya terlihat, jadi kita akan menggunakan “Window Builder” untuk melakukannya

##### 2.1.1 Membuat Project “Window Builder”

- Sebelum mulai, kita perlu membuat project yang dapat menggunakan fitur2 “Window Builder”. Klik kanan mouse di Package explorer, lalu pilih new, pilih project, lalu akan muncul window yang memberi perintah untuk memilih wizard, lalu cari “Window Builder”, SWT Designer, SWT/JFace Java Project.
- Sesudah membuat project baru, CTRL+N lalu pilih Window Builder->Swing Designer -> “Application Window”, setelah itu beri nama package dan class nya, dan setelah itu akan tersedia menu design di menu diatas console, tekan di menu desain dan kamu siap untuk membuat desain UI mu.

##### 2.1.2 Membuat tampilan UI Program kalkulator.

Pada praktikum ini kita akan membuat program menggunakan 2 fitur yang disediakan “Window Builder”. Yaitu ‘JButton’ dan ‘JTextField’

- Di menu Palletes, pilih Layouts lalu cari layout yang sesuai dengan program kita. Pada praktikum ini kita menggunakan Absolute Layout, sehingga kita bisa Membuat Komponen dapat digerakkan dengan leluasa
- Pada Praktikum ini tambahkan komponen sesuai gambar dibawah



Merah = JTextField  
Hijau JButton  
NB:  
1. Ubah Nama Button jadi “btr” + Nama sesuai tulisan yang ada di gambar

- Untuk TextField Yang disebelah label Hasil ubah status “Editable” menjadi false (Hilangkan checklist)

## 2.2 Coding UI

Setelah kita mendesain UI dari program kita, selanjutnya kita perlu memberi codingan agar program dapat berjalan, bukan hanya tampilan.

### 2.2.1 Kode program

- Double click di Salah satu bottun di menu design, sebagai contoh kita pakai “btn0”.
- Lalu tambahkan code seperti gambar dibawah ini.

```
public void actionPerformed(ActionEvent e) {  
    String Number = textField.getText()+btn0.getText();  
    textField.setText(Number);  
}
```

- Kode ini berfungsi untuk mengambil teks dari dalam Jbutton lalu menambahkan teks tersebut kedalam textfield, disclaimer, untuk bagian text “btn0.getText()” ubah semua btn sesuai dengan angka yang ada.
- Lakukan hal yang sama kepada semua tombol angka dan titik (‘0’, ‘1’, ‘2’, ‘3’, ‘4’, ‘5’, ‘6’, ‘7’, ‘8’, ‘9’, ‘0’, ‘00’, ‘.’).
- Untuk tombol clear tambahkan code:

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

- Guna code ini Adalah untuk menghilangkan semua teks yang ada di textfield.
- Untuk tombol backspace “yang gambar panah”, Kalian mungkin bingung bagaimana caranya membuat symbol panah kebelakang tersebut. Caranya hanya perlu menukar code di bagian setFont menjadi seperti ini:

```
btnBackspace.setFont(new Font("Wingdings", Font.BOLD, 14));
```

Ini akan mengubahnya menjadi symbol panah.

- lalu tambahkan kode ini

```
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);  
    }  
}
```

Kode ini berfungsi untuk mengambil 1 karakter di dalam textfield lalu menghapusnya

- Selanjutnya kita mulai membuat coding untuk operasi buat 4 variable untuk seluruh fungsi dengan format seperti ini:

```
double first;  
double second;  
double result;  
String Operation;  
String Answer;
```

- Lalu pergi ke menu design, Double click di tombol operasi, sebagai contoh disini kita pilih di 'btnPlus':

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

Lakukan hal yang sama ke semua operasi dan ubah yang berhubungan dengan symbol plus dengan symbol sesuai button('+', '-', '\*', '/', '%').

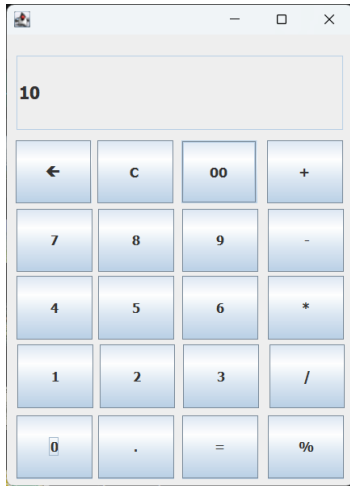
- Untuk tombol '=' buatlah code seperti dibawah ini:

```
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);  
    }  
}
```

Dan siap, program sudah bisa di jalankan.

### 2.2.2 Contoh Output

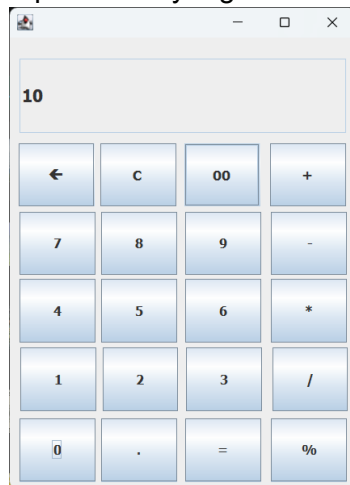
Input pertama yang diberikan:



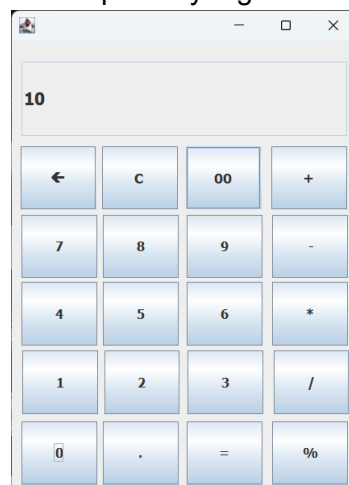
Input operator '+' yang diberikan:



Input kedua yang diberikan:



Hasil operasi yang diberikan:





## **BAB III**

### **KESIMPULAN**

Dari praktikum ini, kita bisa memahami dan mengimplementasikan, pemahaman kita pada praktikum2 sebelumnya, sehingga kita dapat membuat program dengan Front-end dan Back-end yang mencukupi.

Secara keseluruhan, tujuan dari praktikum pertemuan kesembilan berhasil mencapai tujuannya yaitu memperkenalkan “Window Builder” Eclipse IDE kepada mahasiswa, sehingga mahasiswa dapat membuat program yang tidak hanya bekerja di console, melainkan dapat lebih mudah dipahami dan di mengerti oleh khalayak ramai.

Akhirnya, pemahaman ini membuka jalan untuk topik lanjutan di pemograman java. Dengan praktikum ini berhasil melatih para mahasiswa agar dapat mengembangkan program dengan User Interface. Program yang dibuat mahasiswa kelak tidak hanya dilakukan menggunakan text, tapi bisa menggunakan User interface sesuai dengan program yang mereka inginkan, dan mempermudah para Pengguna untuk memahami program.

Sekian Laporan Praktikum Pekan 9 ini saya susun

## DAFTAR PUSTAKA

1. Schildt, Herbert. (2019). *Java: The Complete Reference*. McGraw-Hill Education.
2. Oracle. (2023). *The Java™*.
3. Deitel, Paul & Deitel, Harvey. (2020). *Java How to Program*. Pearson Education.
4. Horstmann, Cay S. (2018). *Core Java Volume I—Fundamentals*. Addison Wesley.S

