

**LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**

LAPORAN PRAKTIKUM PEKAN 9

**Disusun Oleh:
Muhammad Aufa Rafiki
2511531012**

**Dosen Pengampu: Dr. Wahyudi S.T.M.T
Asisten Praktikum: Muhammad Zaki Al Hafiz**



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Pedoman ini disusun sebagai rujukan resmi bagi mahasiswa Departemen Informatika dalam penyusunan laporan praktikum pada mata kuliah *Pemrograman Dasar dengan Java*. Dokumen ini tidak hanya memberikan gambaran umum mengenai format penulisan, tetapi juga menguraikan secara rinci sistematika laporan, tata cara penyajian isi, serta contoh penulisan kode program yang dilengkapi dengan referensi ilmiah. Melalui panduan ini, mahasiswa diharapkan mampu menyusun laporan yang tidak sekadar memenuhi aspek administratif, tetapi juga mencerminkan ketelitian, keteraturan, dan penerapan kaidah penulisan akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam jenjang studi selanjutnya.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR PUSTAKA	ii
BAB I PENDAHULUAN	1
1.1 Pengertian Pratikum.....	1
1.2 Tujuan Pratikum	1
1.3 Persyaratan Pratikum	1
1.4 Waktu dan Tempat Pratikum.....	2
BAB II PEMBAHASAN	3
2.1 Calculator	3
BAB III KESIMPULAN	8
DAFTAR PUSTAKA	9

BAB I PENDAHULUAN

1.1 Pengertian Pratikum

Praktikum Java adalah kegiatan pembelajaran yang dilakukan di laboratorium komputer untuk mengasah keterampilan mahasiswa dalam memahami serta menerapkan konsep pemrograman Java. Kegiatan ini tidak hanya menekankan pada penguasaan teori, tetapi juga pada latihan penyusunan kode program, pengujian, hingga analisis hasil eksekusi. Praktikum dipandang sebagai wahana latihan yang menjembatani pemahaman konseptual dengan kemampuan teknis pemrograman.

1.2 Tujuan Pratikum

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.** Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
- 2.** Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
- 3.** Meningkatkan keterampilan dalam memecahkan masalah (*problem solving*) dengan pendekatan algoritmik.
- 4.** Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
- 5.** Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.

1.3 Persyaratan Pratikum

Agar praktikum berjalan lancar, mahasiswa perlu memenuhi beberapa persyaratan berikut:

- 1.** Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.
- 2.** Membawa perlengkapan yang diperlukan, antara lain laptop atau komputer yang sudah terpasang Java Development Kit (JDK) dan Integrated Development Environment (IDE) yang direkomendasikan.

3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

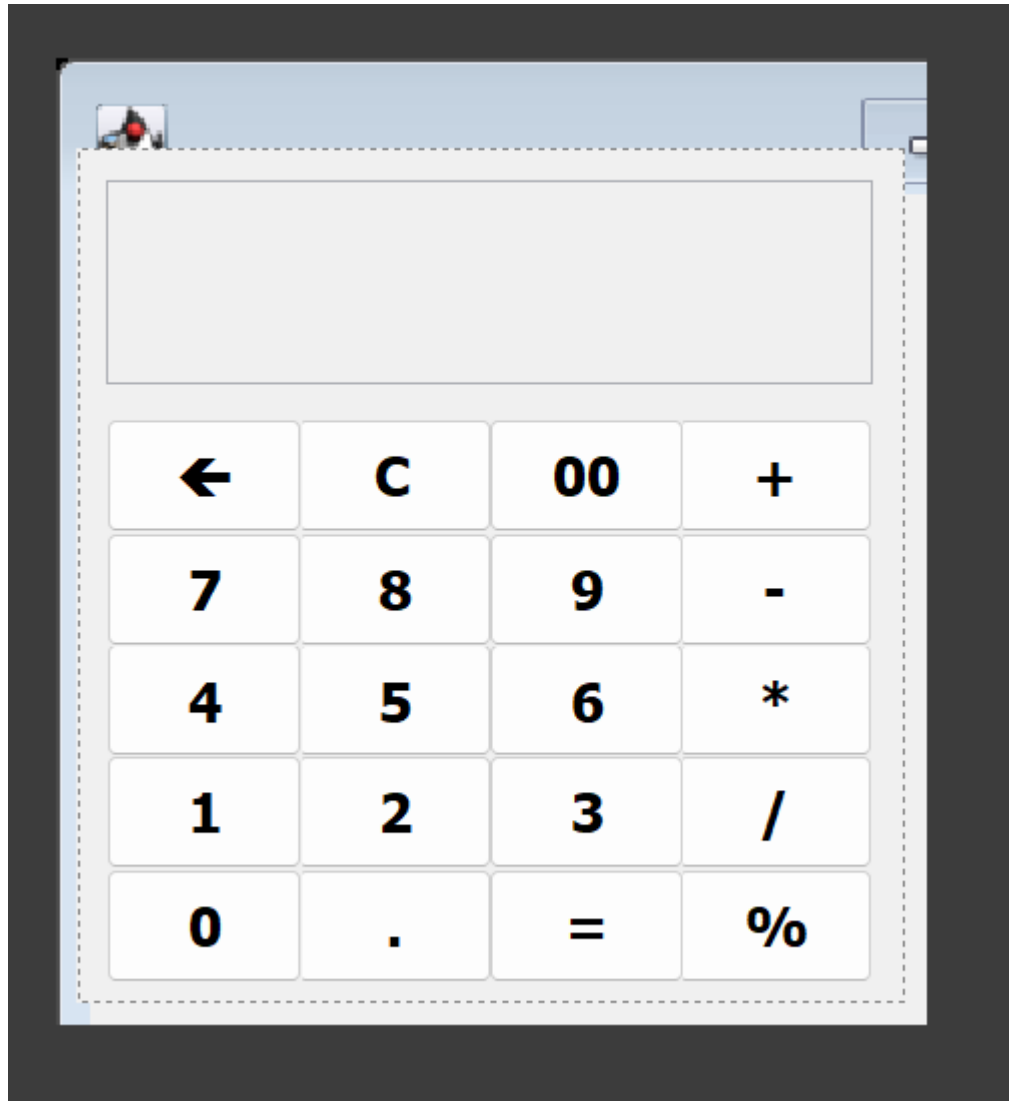
1.4 Waktu dan Tempat Pratikum

Pelaksanaan praktikum Java mengikuti kalender akademik yang berlaku pada program studi. Setiap sesi praktikum dilaksanakan sesuai jadwal yang ditentukan oleh dosen pengampu. Tempat kegiatan umumnya berlangsung di laboratorium komputer, namun pada kondisi tertentu dapat dilaksanakan secara mandiri dengan perangkat masing-masing, selama memenuhi syarat teknis yang ditetapkan.

BAB II PEMBAHASAN

2.1 Calculator

1. Design



Gambar 2.1

2. Kode Program

1. Untuk angka (0-9), titik(.), dan angka (00) menggunakan JButton

```
JButton btn1 = new JButton("1");
btn1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String number=textField.getText()+btn1.getText();
        textField.setText(number);
    }
});
```

Gambar 2.2

2. Untuk operation (+, -, *, /, %)

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

Gambar 2.3

3. Backspace (←)

```
JButton btnBackspace = new JButton("\uF0E7");
btnBackspace.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);
        }
    }
});
```

Gambar 2..4

4. Clear (C)

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

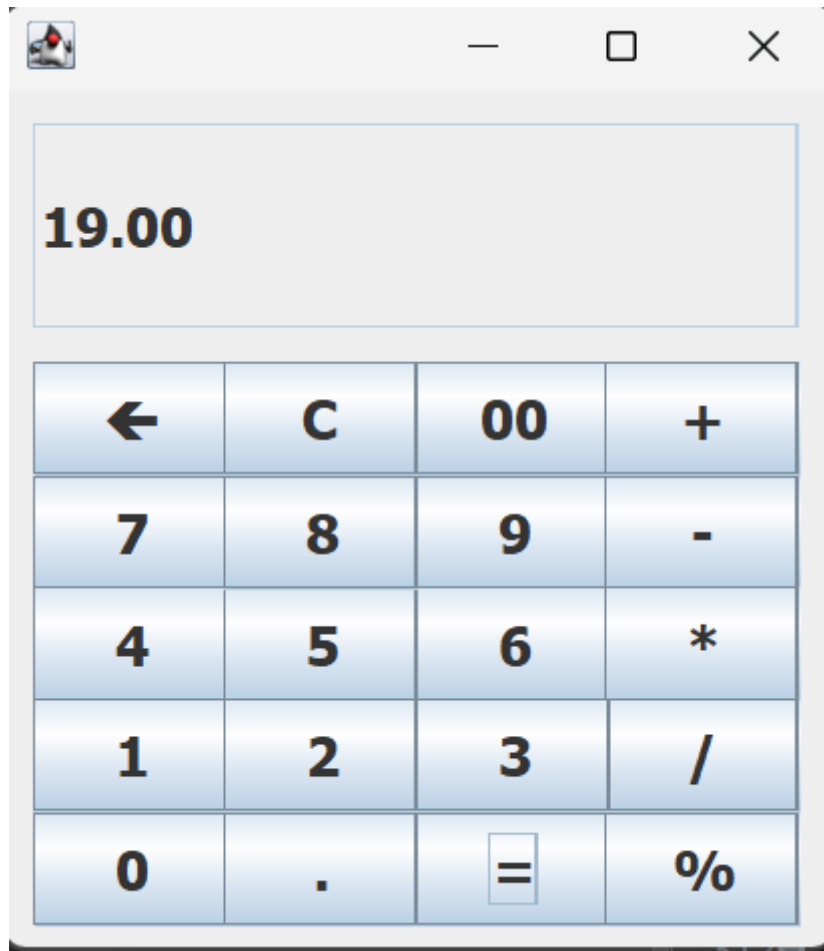
Gambar 2.5

5. Equal (=)

```
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 2.6

3. Output



Gambar 2.7

4. Penjelasan logika program

Logika dasar kalkulator ini dibangun di atas konsep penyimpanan keadaan (state) sementara dan manipulasi teks. Kode dimulai dengan menyiapkan beberapa variabel global sebagai "memori" untuk menampung angka pertama (first), angka kedua (second), hasil hitungan (result), dan jenis operasi matematika yang dipilih (operation). Saat pengguna menekan tombol angka (0-9), program sebenarnya belum melakukan perhitungan matematika apa pun. Sebaliknya, program menjalankan logika penggabungan kata (string concatenation), di mana teks yang sedang tampil di layar diambil, lalu angka baru yang ditekan "ditempelkan" di belakangnya, sehingga angka-angka tersebut berderet membentuk bilangan yang diinginkan.,

Alur logika berubah ketika pengguna menekan salah satu tombol operasi (seperti +, -, *, atau /). Pada momen ini, program melakukan tiga langkah penting sekaligus: pertama, ia mengubah teks angka yang ada di layar menjadi tipe data bilangan desimal (double) dan menyimpannya ke variabel first. Kedua, program mencatat simbol operasi yang ditekan ke dalam variabel operation sebagai penanda untuk nanti. Ketiga, program membersihkan layar teks menjadi kosong. Langkah ini krusial karena memberi ruang bagi pengguna untuk mulai mengetik angka kedua tanpa

tercampur dengan angka pertama.

Proses eksekusi utama terjadi saat tombol "sama dengan" (=) ditekan. Program mengambil angka yang baru saja diketik di layar yang kosong tadi dan menyimpannya sebagai variabel second. Selanjutnya, serangkaian logika percabangan (if-else) akan memeriksa "catatan" yang ada di variabel operation. Jika catatannya adalah tanda tambah, maka first ditambah second, begitu pula untuk operasi lainnya. Setelah hasil matematika didapatkan, nilainya diformat khusus agar menampilkan dua angka di belakang koma, lalu diubah kembali menjadi teks untuk ditampilkan ke layar pengguna sebagai hasil akhir. Sementara itu, fitur tambahan seperti Backspace bekerja dengan memanipulasi string menggunakan StringBuilder untuk menghapus satu karakter terakhir, dan tombol Clear (C) hanya bertugas mereset layar kembali menjadi kosong.

BAB III

KESIMPULAN

Kalkulator sederhana yang diimplementasikan menggunakan Java Swing beroperasi berdasarkan prinsip manajemen keadaan (state management) yang diatur oleh variabel-variabel global (first, second, operation, result). Logika kunci dari aplikasi ini adalah memisahkan proses Input Angka dari proses Eksekusi Operasi. Pengguna memasukkan angka melalui manipulasi string (penggabungan teks) di JTextField. Ketika tombol operasi (+, -, *, /, %) ditekan, kalkulator beralih fungsi menjadi perekam data: ia menyimpan angka pertama (first), mencatat jenis operasi (operation), dan mengosongkan layar untuk input angka kedua. Proses perhitungan yang sebenarnya hanya dieksekusi setelah tombol "sama dengan" (=) ditekan. Pada titik ini, program menggunakan struktur kontrol bersyarat (if-else) untuk menentukan operasi yang benar berdasarkan variabel operation yang tersimpan, menghitung hasilnya, dan menampilkannya kembali ke antarmuka pengguna setelah diformat menjadi string yang rapi.

DAFTAR PUSTAKA

- [1] Programmer Expert, "GUI Calculator in JAVA using eclipse ide | tech Projects," YouTube. 15 Jul. 2023. Diakses pada: 24 Nov. 2025.
[<https://www.youtube.com/watch?v=mZVD5OssHQM>]