

**LAPORAN PRATIKUM**  
**ALGORITMA DAN PEMROGRAMAN**  
**“Laporan Praktikum Pekan 4”**

Disusun oleh:

Rifal maulana oskar

2511533024

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

Asisten Praktikum : Rahmad Dwirizki Olders



DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS

2025

## **KATA PENGANTAR**

Puji syukur ke hadirat Allah SWT karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum ini dengan baik.

Laporan ini disusun untuk memenuhi tugas praktikum Algoritma dan Pemrograman pada pertemuan pertama dengan topik dasar pemrograman Java, yaitu pembuatan program sederhana 'Hello World' dan simulasi prosedur 'Membuat Kue'.

Semoga laporan ini dapat memberikan pemahaman dasar mengenai penggunaan bahasa pemrograman Java, khususnya dalam mengenal struktur dasar program, fungsi, serta pemanggilan metode.

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

Padang, 19 September 2025

Penulis

## DAFTAR ISI

<b>BAB I</b> .....	4
1.1 Latar belakang .....	4
1.2 Tujuan .....	4
1.3 Manfaat .....	5
<b>BAB II</b> .....	6
2.1 Langkah kerja program ifelse1. ....	6
2.2 Contoh program ifelse1. ....	6
2.4 Hasil output program ifelse. ....	7
2.5 Analisis hasil dan Teori : .....	7
2.6 Langkah kerja program Latif1. ....	7
2.7 Contoh kode program Latif1. ....	8
2.8 Hasil output : .....	8
2.9 Analisis hasil dan teori : .....	8
2.10 Langkah kerja program multiIf. ....	9
2.11 contoh kode program multiIf. ....	9
2.12 Hasil output .....	10
2.13 Analisis hasil dan teori .....	10
2.14 Langkah kerja program NamaBulan. ....	10
2.15 Contoh kerja program NamaBulan. ....	11
2.16 Hasil output .....	13
2.17 Analisis hasil dan teori. ....	13
2.18 Langkah kerja program Nilai. ....	13
2.19 Contoh kerja program Nilai. ....	13
2.20 Hasil output .....	14
2.21 Analis hasil dan teori. ....	14
<b>BAB III</b> .....	15
<b>DAFTAR PUSTAKA</b> .....	16

## BAB I

### PENDAHULUAN

#### 1.1 Latar belakang

Pemrograman komputer merupakan disiplin ilmu fundamental dalam era digital, yang bertujuan memberikan instruksi terperinci agar komputer dapat menyelesaikan masalah tertentu. Inti dari proses ini adalah algoritma, yaitu serangkaian langkah logis yang tersusun sistematis. Agar algoritma dapat dijalankan oleh mesin, ia harus diimplementasikan menggunakan bahasa pemrograman. Java adalah salah satu bahasa pemrograman berorientasi objek yang paling populer dan serbaguna, dikenal karena sifatnya yang platform-independent (ditulis sekali, dijalankan di mana saja).

Dalam pengembangan aplikasi, suatu program jarang sekali hanya berjalan secara linear dari awal hingga akhir. Seringkali, program perlu membuat keputusan dan mengambil jalur eksekusi yang berbeda-beda berdasarkan kondisi tertentu. Inilah yang dikenal sebagai konsep kontrol alir (control flow) atau struktur kendali.

Materi percabangan atau if-else, switch-case, dan multi-if (yang pada dasarnya adalah rangkaian if-else if-else) merupakan pilar utama dalam membangun kontrol alir di Java. Statement sendiri adalah unit terkecil dari aksi yang dapat dieksekusi dalam program Java, dan struktur kendali ini adalah jenis statement yang menentukan urutan statement lain akan dieksekusi. Tanpa struktur kendali ini, program tidak akan mampu merespons input yang berbeda, memproses data yang bervariasi, atau menangani kasus pengecualian, sehingga fungsinya menjadi sangat terbatas.

Oleh karena itu, penguasaan terhadap mekanisme kontrol alir, khususnya if-else, switch-case, dan variannya dalam Java, sangat krusial bagi setiap pengembang untuk dapat merancang algoritma yang efisien, fleksibel, dan reaktif terhadap berbagai kondisi dunia nyata. Makalah ini akan mengupas tuntas implementasi dan perbedaan fungsionalitas dari berbagai struktur kendali tersebut dalam konteks algoritma pemrograman Java..

#### 1.2 Tujuan

Tujuan dari penulisan makalah ini adalah:

- Menganalisis dan menjelaskan secara rinci konsep dasar statements dalam pemrograman Java.
- Menguraikan cara kerja dan sintaks dari struktur kontrol alir if-else sebagai mekanisme pengambilan keputusan biner (dua pilihan).
- Membandingkan dan menjelaskan implementasi dari struktur switch-case sebagai alternatif yang efektif untuk pengambilan keputusan dengan banyak pilihan (multi-cabang).

-Menjelaskan konsep dan implementasi dari multi-if (rangkaian if-else if-else) dalam menangani kondisi bersyarat yang kompleks dan berlapis.

1

2

-Memberikan contoh algoritma dan kode program sederhana yang mengaplikasikan masing-masing struktur kendali tersebut.

### 1.3 Manfaat

-Menambah wawasan dan pemahaman akademis mengenai pentingnya kontrol alir dalam merancang algoritma yang logis dan terstruktur.

-Menjadi sumber referensi bagi mahasiswa dan pembaca lain yang mempelajari dasar-dasar algoritma pemrograman, khususnya dalam bahasa Java.

-Membantu pembaca dalam menguasai penggunaan if-else, switch-case, dan multi-if sehingga mampu menulis kode Java yang lebih efisien dan mudah dibaca (readable).

-Memberikan dasar yang kuat bagi pengembang pemula dalam memilih struktur kendali yang paling tepat untuk situasi algoritma yang dihadapi, misalnya kapan harus menggunakan switch daripada rangkaian if-else if.

-Meningkatkan kemampuan pemecahan masalah (problem-solving) melalui implementasi struktur kendali dalam menghadapi berbagai skenario dan kondisi program.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Langkah kerja program ifelse1.**

- Membuat kelas operator dalam package pekan4.
- Deklarasikan class utama dengan nama ifelse1.
- Di dalamnya, buat method main sebagai titik awal program.
- Deklarasikan beberapa variabel dengan tipe data int.
- Berikan nilai berupa int(angka);.
- Gunakan System.out.println() untuk menampilkan isi variabel.

#### **2.2 Contoh program ifelse1.**

```
package pekan4;

import java.util.Scanner;

public class Ifelse1 {
    public static void main(String[] args) {
        double IPK;
        Scanner input=new Scanner(System.in);
        System.out.print("input IPK anda = ");
        IPK=input.nextDouble();
        input.close();
        if (IPK>2.75) {
            System.out.println("anda lulus sangat memuaskan dengan IPK "+IPK);
        } else {
            System.out.println("anda tidak lulus");
        }
    }
}
```

}

#### 2.4 Hasil output program ifelse.

input IPK anda = 3.0

anda lulus sangat memuaskan dengan IPK 3.0

input IPK anda = 2.0

anda tidak lulus

#### 2.5 Analisis hasil dan Teori :

Struktur if-else adalah inti dari kontrol alir di Java, berfungsi untuk membuat keputusan biner (dua arah). Ia mengevaluasi satu ekspresi boolean (nilai IPK  $\geq$  batasLulus).

Jika ekspresi menghasilkan true, blok kode if dieksekusi (LULUS).

Jika ekspresi menghasilkan false, blok kode else dieksekusi (TIDAK LULUS).

Hanya salah satu blok yang akan dijalankan (mutually exclusive).

Program berhasil mengimplementasikan logika seleksi. Hasil eksekusi pada kedua skenario konsisten dengan teori if-else: program secara efektif memilih jalur LULUS atau TIDAK LULUS. hanya berdasarkan hasil evaluasi kondisi boolean, membuktikan kemampuan program untuk beradaptasi terhadap data input yang berbeda.

#### 2.6 Langkah kerja program Latif1.

- Membuat kelas operator dalam package pekan4.
- Deklarasikan class utama dengan nama Latif1.
- Di dalamnya, buat method main sebagai titik awal program.
- Deklarasikan beberapa variabel dengan tipe data int.
- Berikan nilai berupa int(angka);.
- GunakanSystem.out.println() untuk menampilkan isi variabel.

## 2.7 Contoh kode program Latifl.

```
package pekan4;

import java.util.Scanner;

public class Latifl {

    public static void main (String[] args) {

        double IPK;

        Scanner input=new Scanner (System.in);

        System.out.print("input IPK anda =");

        IPK=input.nextDouble();

        input.close();

        if (IPK>2.75) {

            System.out.println("anda lulus sangat memuaskan dengan IPK "+IPK);

        }

    }

}
```

## 2.8 Hasil output :

input IPK anda =3.0

anda lulus sangat memuaskan dengan IPK 3.0

## 2.9 Analisis hasil dan teori :

Struktur if adalah inti dari kontrol alir di Java, berfungsi untuk membuat keputusan satu arah. Ia mengevaluasi satu ekspresi boolean (nilai IPK  $\geq$  batasLulus).

Jika ekspresi menghasilkan true, blok kode if dieksekusi (LULUS).

Program berhasil mengimplementasikan logika seleksi. Hasil eksekusi pada kedua skenario konsisten dengan teori if: program secara efektif memilih jalur LULUS. hanya berdasarkan hasil evaluasi kondisi String, membuktikan kemampuan program untuk beradaptasi terhadap data input yang berbeda.



## 2.10 Langkah kerja program multiIf.

- Membuat kelas operator dalam package pekan4.
- Deklarasikan class utama dengan nama multiIf.
- Di dalamnya, buat method main sebagai titik awal program.
- Deklarasikan beberapa variabel dengan tipe data int.
- Berikan nilai berupa char dan var.
- Gunakan System.out.println() untuk menampilkan isi variabel.

## 2.11 contoh kode program multiIf.

```
package pekan4;

import java.util.Scanner;

public class multiIf {

    public static void main(String[] args) {

        int umur;

        int sim;

        Scanner a=new Scanner(System.in);

        System.out.print("input umur anda: ");

        umur= a.nextInt();

        System.out.println("apakah anda punya sim c: ");

        sim=a.next().charAt(0);

        a.close();

        if ((umur >= 17)&&(sim=='y')) {

            System.out.println("anda sudah dewasa dan boleh bawa motor");

        }

        if ((umur >= 17)&&(sim!='y')) {

            System.out.println("anda sudah dewasa tetapi tidak boleh bawa motor");

        }

    }

}
```

```

if ((umur < 17)&&(sim!='y')) {
    System.out.println("anda belum cukup umur bawa motor");
}
if ((umur < 17)&&(sim=='y')) {
    System.out.println("anda belum cukup umur punya sim");
}

```

7

## 2.12 Hasil output

input umur anda: 18

apakah anda punya sim c:

ya

anda sudah dewasa dan boleh bawa motor

## 2.13 Analisis hasil dan teori

Struktur multi-if dalam Java adalah alat penting untuk mengimplementasikan logika bersyarat yang kompleks dan berjenjang. Analisis menunjukkan bahwa:

- Ia menjamin eksekusi tunggal dari berbagai pilihan.
- Efektivitasnya sangat bergantung pada urutan kondisi—kondisi yang paling spesifik atau membatasi harus diletakkan lebih awal.
- Prinsip penghentian dini memastikan efisiensi program dengan membatasi pemeriksaan hanya sampai kondisi pertama yang terpenuhi.

## 2.14 Langkah kerja program NamaBulan.

- Membuat kelas operator dalam package pekan4.
- Deklarasikan class utama dengan nama NamaBulan.
- Di dalamnya, buat method main sebagai titik awal program.
- Deklarasikan beberapa variabel dengan tipe data char.
- Berikan nilai berupa char dan var.
- Gunakan System.out.println() untuk menampilkan isi variabel.

### 2.15 Contoh kerja program>NamaBulan.

```
package pekan4;

import java.util.Scanner;

public class>NamaBulan {

    public static void main (String[] args) {

        Scanner Scanner = new Scanner(System.in);

        System.out.println("masukan angka bulan (1 - 25): ");

        int bulan = Scanner.nextInt();

        switch (bulan) {

            case 1:

                System.out.println("januari");

                break;

            case 2:

                System.out.println("febuari");

                break;

            case 3:

                System.out.println("maret");

                break;

            case 4:

                System.out.println("april");

                break;

            case 5:

                System.out.println("mei");

                break;

            case 6:
```

```
        System.out.println("juni");
        break;
case 7:
    System.out.println("juli");
    break;

case 8:
    System.out.println("agustus");
    break;
case 9:
    System.out.println("september");
    break;
case 10:
    System.out.println("oktober");
    break;
case 11:
    System.out.println("november");
    break;
case 12:
    System.out.println("desember");
    break;
default:
    System.out.println("angka tidak valid");
}
Scanner.close();

}
```

### 2.16 Hasil output

masukan angka bulan (1 - 25): 9

september

### 2.17 Analisis hasil dan teori.

Struktur if adalah inti dari kontrol alir di Java, berfungsi untuk membuat keputusan satu arah. Ia mengevaluasi satu ekspresi boolean (angka bulan (1-25) = nama bulan).

Program berhasil mengimplementasikan logika seleksi. Hasil eksekusi pada kedua skenario konsisten dengan teori if: program secara efektif memilih jalur nama bulan. hanya berdasarkan hasil evaluasi kondisi String, membuktikan kemampuan program untuk beradaptasi terhadap data input yang berbeda.

### 2.18 Langkah kerja program Nilai.

- Membuat kelas operator dalam package pekan4.
- Deklarasikan class utama dengan nama Nilai.
- Di dalamnya, buat method main sebagai titik awal program.
- Deklarasikan beberapa variabel dengan tipe data int dan char.
- Berikan nilai berupa char dan int.
- Gunakan System.out.println() untuk menampilkan isi variabel.

### 2.19 Contoh kerja program Nilai.

```
package pekan4;
import java.util.Scanner;
public class Nilai {
    public static void main(String[] args) {
        int nilai;
        Scanner input=new Scanner(System.in);
        System.out.println("inputkan nilai angkamu = ");
        nilai=input.nextInt();
```

```

        input.close();
        if (nilai >=81) {
            System.out.println("A");
        } else if (nilai >=70) {
            System.out.println("B");

        } else if (nilai >=60) {
            System.out.println("C");
        } else if (nilai >=50) {
            System.out.println("D");
        } else {
            System.out.println("E");
        }
    }
}

```

11

## 2.20 Hasil output

inputkan nilai angkamu = 99

A

## 2.21 Analisis hasil dan teori.

Teori kunci dari multi-if adalah evaluasi berurutan dan penghentian dini (short-circuiting). Kondisi diperiksa dari atas ke bawah, dan eksekusi berhenti begitu satu kondisi ditemukan bernilai true.

Analisis: Program berjalan efisien. Meskipun 85 juga lebih besar dari 70, 60, dan 50, kondisi pertama yang terpenuhi (yaitu nilai  $\geq 81$ ) memicu output "A" dan menghentikan pengecekan lebih lanjut.

### BAB III

## KESIMPULAN

Berdasarkan latar belakang, tujuan, dan pembahasan mengenai implementasi struktur kontrol alir dalam pemrograman Java, khususnya if-else dan multi-if, dapat ditarik beberapa kesimpulan utama:

- Pentingnya Kontrol Alir: Struktur kendali, yang diimplementasikan melalui statements seperti if-else dan multi-if, adalah fundamental bagi setiap algoritma. Struktur ini memungkinkan program untuk membuat keputusan dan mengubah alur eksekusi secara dinamis sebagai respons terhadap kondisi atau data input yang berbeda.
- if-else (Keputusan Biner): Struktur if-else adalah alat dasar untuk menerapkan seleksi biner. Ia menjamin bahwa dari dua pilihan yang ada, hanya satu yang akan dieksekusi berdasarkan hasil evaluasi ekspresi boolean (true atau false), menjadikannya ideal untuk penentuan kondisi dua arah (misalnya, LULUS/GAGAL).
- multi-if (Keputusan Multi-Cabang): Struktur multi-if (if-else if-else) memungkinkan implementasi logika berjenjang yang kompleks. Efektivitasnya bergantung pada evaluasi berurutan dan prinsip penghentian dini (short-circuiting). Struktur ini memastikan bahwa dari banyak kemungkinan kondisi, hanya kondisi pertama yang terpenuhi yang akan dieksekusi, dengan blok else terakhir berfungsi sebagai jalur default atau penangkap kesalahan.
- Fleksibilitas Algoritma: Penguasaan ketiga mekanisme ini memungkinkan pengembang untuk merancang algoritma Java yang efisien, fleksibel, dan mampu menangani skenario dunia nyata dengan beragam kondisi, mulai dari penentuan kelulusan hingga sistem penilaian mutu yang kompleks.

**DAFTAR PUSTAKA**

- Schildt, H. (2019). Java: The Complete Reference (11th Edition). McGraw-Hill Education.
- Oracle. (2023). The Java™ Tutorials – Scanner Class.  
<https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>
- Eckel, B. (2006). Thinking in Java (4th Edition). Prentice Hall.
- Horstmann, C. S., & Cornell, G. (2019). Core Java Volume I–Fundamentals (11th Edition). Prentice Hall.
- Deitel, P., & Deitel, H. (2020). Java How to Program, Early Objects. Edisi ke-12. Pearson Education.
- Gaddis, T. (2018). Starting Out with Java: From Control Structures through Objects. Edisi ke-7. Pearson Education.
- Schildt, H. (2019). Java: The Complete Reference. Edisi ke-11. McGraw-Hill Education.



