

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
“LAPORAN CUACA”

Disusun Oleh:

Rifqi Aditya

2511533002

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 atas rahmat dan karunia-Nya saya dapat menyelesaikan laporan praktikum mata kuliah **Algoritma dan Pemrograman** dengan judul "*Laporan Cuaca*".

Laporan ini disusun untuk memenuhi salah satu tugas praktikum sekaligus sebagai sarana pembelajaran dalam memahami dasar-dasar pemrograman menggunakan bahasa Java. Melalui praktikum ini, penulis dapat mempelajari penggunaan tipe data dasar seperti *integer*, *float*, *char*, dan *boolean*, serta bagaimana menampilkan informasi ke layar menggunakan perintah `System.out.println`.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah membantu dalam proses penyusunan laporan ini.

Akhir kata, semoga laporan ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi pembaca, khususnya dalam memperdalam pemahaman tentang pemrograman dasar dengan Java.

Padang, 26 September 2025

Penulis

Daftar Pustaka

BAB I.....	4
1.1 Latar Belakang.....	4
1.2 Tujuan.....	4
1.3 Manfaat.....	4
BAB II	5
2.1 Operator Aritmetika.....	Error! Bookmark not defined.
2.2 Operator Penugasan (Assignment)	Error! Bookmark not defined.
2.3 Operator Relasional	Error! Bookmark not defined.
2.4 Operator Logika.....	Error! Bookmark not defined.
DAFTAR PUSTAKA.....	14

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman Java merupakan landasan penting dalam mempelajari algoritma. Untuk menguasainya, pemahaman mendalam tentang elemen dasar seperti tipe data sangatlah diperlukan. Praktikum ini bertujuan untuk memberikan pengalaman praktis dalam penggunaan tipe data di Java, meliputi *integer*, *float*, *char*, *double* dan *Boolean*.

1.2 Tujuan

1. Melatih keterampilan menampilkan output ke layar dengan format yang rapi.
2. Membiasakan penggunaan penginputan sebagai media tampilan informasi.
3. Menerapkan deklarasi variabel dengan berbagai tipe data dasar di Java.

1.3 Manfaat

1. Mahasiswa mampu membuat program sederhana.
2. Mahasiswa memahami penggunaan tipe data dasar dalam Java.
3. Mahasiswa dapat mengaitkan pembelajaran pemrograman dengan kasus nyata di dunia industri.

BAB II

PEMBAHASAN

Bab ini menyajikan analisis mendalam terhadap kode program yang mempraktikkan penggunaan struktur kontrol percabangan dalam bahasa Java. Struktur ini memungkinkan program untuk membuat keputusan dan menjalankan blok kode yang berbeda berdasarkan kondisi yang dievaluasi. Pembahasan akan mencakup uraian kode, langkah kerja, serta analisis hasil yang didukung oleh teori dan referensi yang relevan untuk setiap struktur yang diimplementasikan, yaitu if-else, if-else if-else, multi if, dan switch-case.

2.1 Struktur Percabangan Dua Arah (If-Else)

Struktur if-else adalah bentuk paling dasar dari kontrol percabangan yang menyediakan dua kemungkinan alur eksekusi.

2.1.1 Uraian Kode Program

```
1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class latif1 {
6
7     public static void main(String[] args) {
8         double ipk;
9         Scanner input=new Scanner(System.in);
10        System.out.println("Input IPK Anda : ");
11        ipk=input.nextDouble();
12        input.close();
13        if (ipk>2.75) {
14            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK " + ipk);
15        }else {
16            System.out.println("Anda Tidak Lulus");//|
17        }
18    }
19
20 }
21
```

Kode Program 1

Program di atas dirancang untuk menentukan status kelulusan seorang mahasiswa berdasarkan IPK. Program akan meminta pengguna memasukkan nilai IPK yang bertipe data double. Selanjutnya, program akan mengevaluasi apakah nilai IPK tersebut lebih besar dari

2.75. Jika kondisi terpenuhi, sebuah pesan kelulusan akan ditampilkan dan jika tidak, pesan yang berbeda akan muncul.

2.1.2 Langkah Kerja Program

1. Program dimulai, lalu mendeklarasikan peubah ipk dengan tipe data double.
2. Objek Scanner dibuat untuk menerima masukan dari pengguna.
3. Program menampilkan pesan "Input IPK Anda : " ke layar.
4. Pengguna memasukkan sebuah nilai desimal, yang kemudian disimpan dalam peubah ipk.
5. Program mengevaluasi kondisi $ipk > 2.75$.
6. Jika kondisi bernilai true, blok kode di dalam if dieksekusi, yaitu menampilkan "Anda Lulus Sangat Memuaskan dengan IPK " diikuti nilai IPK.
7. Jika kondisi bernilai false, blok kode di dalam else dieksekusi, yaitu menampilkan "Anda Tidak Lulus".
8. Program selesai.

2.1.3 Analisis Hasil

Struktur if-else merupakan implementasi dari percabangan dua arah (two-way selection). Secara teoretis, struktur ini memastikan bahwa dari dua blok kode yang tersedia (blok if dan blok else), hanya satu yang akan dieksekusi. Pilihan ini bergantung sepenuhnya pada hasil evaluasi ekspresi boolean (true atau false). Sebagai contoh, jika pengguna memasukkan IPK 3.15, ekspresi $3.15 > 2.75$ akan bernilai true, sehingga keluaran program adalah "Anda Lulus Sangat Memuaskan dengan IPK 3.15". Sebaliknya, jika IPK yang dimasukkan adalah 2.50, ekspresi $2.50 > 2.75$ bernilai false, dan program akan menjalankan blok else yang menghasilkan keluaran "Anda Tidak Lulus". Ini menunjukkan bagaimana alur program dapat "bercabang" ke jalur yang berbeda berdasarkan satu kondisi tunggal.

2.2 Struktur Percabangan Multi-Arah (If-Else If-Else)

Struktur if-else if-else, digunakan ketika terdapat lebih dari dua kemungkinan alur eksekusi. Program ini adalah contoh implementasi dari struktur ini.

2.2.1 Uraian Kode Program

```
1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class nilai {
6
7     public static void main(String[] args) {
8         int nilai;
9         Scanner input=new Scanner(System.in);
10        System.out.println("Inputkan Nilai Angka : ");
11        nilai=input.nextInt();
12        input.close();
13
14        if(nilai>=80) {
15            System.out.println("A");
16        }else if(nilai >=70) {
17            System.out.println("B");
18        }else if(nilai >=60) {
19            System.out.println("C");
20        }else if(nilai >=50) {
21            System.out.println("D");
22        }else {
23            System.out.println("E");//
24        }
25    }
26
27 }
28
```

Kode Program 2

Program ini berfungsi untuk mengonversi nilai angka menjadi nilai huruf. Program meminta pengguna memasukkan sebuah nilai integer yang disimpan dalam peubah nilai. Kemudian, nilai tersebut dievaluasi secara berjenjang untuk menentukan predikat yang sesuai.

2.2.2 Langkah Kerja Program

1. Program dimulai dan meminta pengguna memasukkan sebuah nilai angka integer.
2. Program pertama kali memeriksa kondisi nilai ≥ 80 . Jika true, program mencetak "A" dan seluruh sisa struktur else if dan else akan dilewati.

3. Jika kondisi pertama false, program akan melanjutkan ke pemeriksaan berikutnya: nilai ≥ 70 . Jika true, program mencetak "B" dan keluar dari struktur.
4. Proses ini berlanjut secara berurutan untuk kondisi nilai ≥ 60 (C) dan nilai ≥ 50 (D).
5. Jika tidak ada satu pun dari kondisi-kondisi di atas yang terpenuhi, blok else terakhir akan dieksekusi, yang mencetak "E".

2.2.3 Analisis Hasil

Struktur if-else if-else sangat efektif untuk menangani kondisi yang **saling eksklusif**. Menurut teori struktur kontrol, evaluasi kondisi dilakukan secara sekuensial dari atas ke bawah. Segera setelah salah satu kondisi ditemukan bernilai true, blok kode yang bersesuaian akan dieksekusi, dan sisa dari rantai else if akan diabaikan. Hal ini membuat struktur ini lebih efisien dibandingkan menggunakan beberapa pernyataan if terpisah untuk kasus yang sama.

Contohnya, jika pengguna memasukkan nilai = 75:

- $75 \geq 80$ bernilai false.
- Program lanjut ke $75 \geq 70$, yang bernilai true.
- Program mencetak "B" dan langsung melompati sisa pengecekan untuk C, D, dan E.

2.3 Struktur Multi If (Kondisi Independen)

Berbeda dari if-else if, penggunaan beberapa pernyataan if secara berurutan menciptakan serangkaian pemeriksaan kondisi yang independen. Program multiif.java menunjukkan kasus ini.

2.3.1 Uraian Kode Program


```

1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class multiif {
6
7     public static void main(String[] args) {
8         int umur;
9         char sim;
10        Scanner a= new Scanner(System.in);
11        System.out.println("Input Umur Anda : ");
12        umur=a.nextInt();
13        System.out.println("Apakah Anda Sudah Punya Sim C : ");
14        sim=a.next().charAt(0);
15        a.close();
16        if((umur >= 17)&&(sim=='y')) {
17            System.out.println("Anda Sudah Dewasa Dan Boleh Bawa Motor");
18        }
19        if ((umur >= 17)&&(sim!='y')) {
20            System.out.println("Anda Sudah Dewasa Tetapi Tidak Boleh Bawa Motor");
21        }
22        if ((umur < 17)&&(sim!='y')) {
23            System.out.println("Anda Belum Cukup Umur Bawa Motor");
24        }
25        if((umur < 17)&&(sim=='y')) {
26            System.out.println("Anda Belum Cukup Umur Punya Sim");//
27        }
28    }
29 }
30

```

Kode Program 3

Program ini bertujuan untuk memberikan keterangan berdasarkan dua kriteria: umur dan kepemilikan Surat Izin Mengemudi (SIM). Program meminta masukan berupa umur (integer) dan status kepemilikan SIM C. Terdapat empat pernyataan if terpisah yang masing-masing mengevaluasi kombinasi kondisi yang berbeda.

2.3.2 Langkah Kerja Program

1. Program meminta masukan untuk umur dan status SIM.
2. **Pengecekan pertama:** Program mengevaluasi `if ((umur >= 17)&&(sim=='y'))`. Jika true, pesan yang sesuai akan dicetak.
3. **Pengecekan kedua:** Tanpa memedulikan hasil pengecekan pertama, program mengevaluasi `if (((umur >= 17)&&(sim!='y'))`. Jika true, pesan yang sesuai akan dicetak.

4. **Pengecekan ketiga dan keempat:** Proses yang sama berlanjut untuk dua kondisi if berikutnya. Setiap if dievaluasi secara mandiri.

2.3.3 Analisis Hasil

Secara teoretis, struktur multi if berarti setiap kondisi if adalah sebuah unit percabangan yang independen. Tidak seperti if-else if yang berhenti setelah satu kondisi terpenuhi, di sini semua kondisi akan diperiksa satu per satu dari atas ke bawah.

Dalam kasus ini, logikanya dirancang agar hanya satu dari empat kondisi yang bisa true pada satu waktu. Misalnya, jika pengguna memasukkan umur = 20 dan sim = 't', hanya kondisi kedua (umur >= 17 DAN sim != 'y') yang akan terpenuhi, dan hanya satu pesan yang tercetak.

2.4 Struktur Percabangan Switch-Case

Struktur switch-case menyediakan alternatif yang lebih rapi dan sering kali lebih mudah dibaca dibandingkan rantai if-else if yang panjang, terutama ketika mengevaluasi satu peubah terhadap beberapa nilai konstanta.

2.4.1 Uraian Kode Program

```

1 package Pekan4;
2
3 import java.util.Scanner;
4
5 public class namabulan {
6
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.println("Masukkan angka bulan (1 - 12) : ");
10        int bulan = scanner.nextInt();
11        switch (bulan) {
12            case 1:
13                System.out.println("Januari");
14                break;
15            case 2:
16                System.out.println("Februari");
17                break;
18            case 3:
19                System.out.println("Maret");
20                break;
21            case 4:
22                System.out.println("April");
23                break;
24            case 5:
25                System.out.println("May");
26                break;
27            case 6:
28                System.out.println("Juni");
29                break;
30            case 7:
31                System.out.println("july");
32                break;
33            case 8:
34                System.out.println("Agustus");
35                break;
36            case 9:
37                System.out.println("September");
38                break;
39            case 10:
40                System.out.println("Oktober");
41                break;
42            case 11:
43                System.out.println("November");
44                break;
45            case 12:
46                System.out.println("Desember");
47                break;
48            default:
49                System.out.println("Angka Tidak Valid");//
50        }
51        scanner.close();
52    }
53

```

Kode Program 4

Program di atas berfungsi untuk menampilkan nama bulan berdasarkan angka yang dimasukkan oleh pengguna. Program menggunakan peubah bulan sebagai ekspresi dalam switch.

2.4.2 Langkah Kerja Program

1. Program meminta pengguna memasukkan angka bulan (1-12).
2. Nilai yang dimasukkan disimpan dalam peubah bulan.
3. Struktur switch mengevaluasi nilai dari bulan.

4. Program akan "melompat" ke blok case yang labelnya cocok dengan nilai bulan.
Misalnya, jika bulan adalah 5, eksekusi akan dimulai dari case 5:.
5. Pernyataan di dalam case tersebut (`System.out.println`) dieksekusi.
6. Pernyataan `break`; kemudian menghentikan eksekusi di dalam switch dan melanjutkan ke baris kode setelah switch.
7. Jika tidak ada case yang cocok, program akan melompat ke blok default: dan mengeksekusi kodenya.

2.4.3 Analisis Hasil

Struktur switch-case secara konseptual adalah struktur percabangan multi-arah yang berbasis pada pencocokan nilai. Keunggulannya terletak pada keterbacaan kode ketika berhadapan dengan banyak pilihan nilai. Penggunaan switch-case pada program ini terbukti efektif dan membuat kode lebih terstruktur dibandingkan jika ditulis dengan 12 tingkat if-else if.

BAB III

KESIMPULAN

Berdasarkan pembahasan dan analisis program pada bab sebelumnya, dapat ditarik kesimpulan mengenai penggunaan dan karakteristik dari berbagai struktur kontrol percabangan dalam bahasa Java:

1. Struktur if-else merupakan mekanisme untuk percabangan dua arah. Struktur ini efektif untuk menangani satu kondisi logis dengan dua kemungkinan hasil yang saling eksklusif.
2. Struktur if-else if-else adalah pengembangan untuk percabangan multi-arah. Keunggulannya terletak pada efisiensi, di mana evaluasi kondisi dilakukan secara sekuensial dan akan berhenti segera setelah salah satu kondisi terpenuhi.
3. Penggunaan multi if cocok untuk situasi di mana terdapat beberapa kondisi yang tidak saling terikat dan perlu dievaluasi secara mandiri. Berbeda dengan if-else if, semua kondisi if akan diperiksa tanpa terkecuali, yang memungkinkan lebih dari satu blok kode dapat dieksekusi jika kondisinya terpenuhi.
4. Struktur switch-case menawarkan alternatif yang lebih terstruktur dan sering kali lebih mudah dibaca dibandingkan rantai if-else if yang panjang, terutama saat membandingkan sebuah peubah tunggal terhadap serangkaian nilai konstanta yang spesifik.

DAFTAR PUSTAKA

- [1] Deitel, P. J. (2018). *Java: How to Program, Early Objects*.
- [2] Horstmann, C. S. (2019). *Core Java Volume I--Fundamentals*.
- [3] Rosihan Ari Yuana, S. M. (2022). *Pemrograman Java*.