

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
CONDITIONAL STATEMENT



disusun Oleh:

Tiara Amalia Insani

NIM 2511532019

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

Asisten Praktikum : Jovantri Immanuel Gulo

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG

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.

Padang, 2 Oktober 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
BAB I	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum	1
1.3 Manfaat	1
BAB II	2
PEMBAHASAN	2
2.1 Conditional Statement	2
2.1.1 If Statement	2
2.1.2 If-Else Statement	2
2.1.3 If-Else If-Else Statement.....	2
2.1.4 Multi If (Beberapa If Terpisah).....	2
2.1.5 Switch-Case Statement	2
2.1.6 Nested If Statement.....	2
2.2 Praktikum	3
2.3.1 Alat dan Bahan	3
2.3.2 Program.....	3
BAB III	7
KESIMPULAN DAN SARAN	7
3.1 Kesimpulan	7
3.2 Saran	7
DAFTAR PUSTAKA	8

DAFTAR GAMBAR

Gambar 2.1	3
Gambar 2.2	4
Gambar 2.3	4
Gambar 2.4	5
Gambar 2.5	6

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam membuat program, kadang kita butuh program bisa mengambil keputusan sendiri sesuai kondisi yang ada. Contohnya menentukan lulus atau tidak dari nilai, menampilkan nama bulan dari angka, atau mengecek umur dan SIM seseorang. Di Java, hal ini bisa dilakukan dengan *conditional statement* seperti *if*, *if-else*, *nested if*, dan *switch-case*. Dengan mempelajari bagian ini, kita jadi tahu cara membuat program yang lebih pintar, fleksibel, dan bisa menyesuaikan diri dengan situasi yang berbeda

1.2 Tujuan Praktikum

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.
6. Belajar bagaimana membuat program yang bisa mengambil keputusan otomatis sesuai kondisi

1.3 Manfaat

Mahasiswa diharapkan semakin terbiasa menggunakan percabangan dalam pemrograman, dan bisa lebih mudah memahami logika pengambilan keputusan, melatih cara berpikir sistematis, dan mampu membuat program sesuai kebutuhan. Juga berguna saat mengerjakan program yang lebih kompleks nantinya.

BAB II

PEMBAHASAN

2.1 Conditional Statement

adalah perintah dalam bahasa pemrograman yang digunakan untuk mengambil keputusan berdasarkan kondisi tertentu. Jika kondisi terpenuhi (*true*), maka suatu blok kode akan dijalankan. Jika tidak (*false*), maka blok kode yang lain bisa dijalankan.

2.1.1 If Statement

Digunakan untuk menjalankan satu blok kode apabila kondisi bernilai benar (*true*)

2.1.2 If-Else Statement

Digunakan jika ada dua kemungkinan kondisi:

- Jika kondisi *true*, blok pertama dijalankan.
- Jika kondisi *false*, blok kedua dijalankan.

2.1.3 If-Else If-Else Statement

Digunakan jika terdapat lebih dari dua kondisi yang mungkin terjadi.

2.1.4 Multi If (Beberapa If Terpisah).

Berbeda dengan if-else, pada multi if semua kondisi akan diperiksa satu per satu, bukan berhenti di kondisi yang benar pertama.

2.1.5 Switch-Case Statement

Digunakan ketika ada banyak pilihan berdasarkan nilai tunggal (misalnya angka atau string). Switch-case lebih rapi dibanding if-else Panjang

2.1.6 Nested If Statement

Nested if adalah percabangan bersarang, yaitu if di dalam if. Blok if bagian dalam hanya dijalankan jika kondisi if bagian luar terpenuhi.

2.2 Praktikum

2.3.1 Alat dan Bahan

1. Laptop/computer dengan eclipse terinstal.
2. Materi praktikum mengenai Conditional Statement.

2.3.2 Program

1. Latihan If

```
package pekan4;

import java.util.Scanner;

public class latIf1 {
    public static void main (String[] args) {
        double IPK;
        Scanner input=new Scanner(System.in);
        System.out.println("Input IPK Anda=");
        IPK=input.nextDouble();
        input.close();
        if(IPK>2.75) {
            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK"+IPK);
        }
    }
}
```

Gambar 2.1

Langkah-langkah :

- a. Buat *Class* baru dengan latIf1.java
- b. *Import library Scanner* untuk membaca input dari keyboard
- c. Di dalam *method main*, buat variabel IPK dengan tipe *double*
- d. Buat objek *Scanner* untuk mengambil input dari keyboard
- e. Gunakan percabangan *if* untuk memberikan keputusan berdasarkan nilai IPK
- f. Tampilkan hasil operasi dengan *System.out.println()*
- g. Jalankan program dan amati hasilnya

2. Ifelse

```
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.println("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");
        }
    }
}
```

Gambar 2.2

Langkah-langkah:

- Buat *class* baru dengan nama *Ifelse1.java* pada *package* *pekan4*.
- Deklarasikan variabel *IPK* dengan tipe data *double*.
- Buat objek *Scanner* untuk membaca input dari keyboard.
- Tampilkan pesan ke layar: "Input IPK Anda=" dan ambil nilai input dari user menggunakan *nextDouble()*.
- Terapkan percabangan *if-else*.
- Jalankan program dan lakukan uji coba.

3. MultiIf

```
package pekan4;

import java.util.Scanner;

public class multiIf {
    public static void main (String[] args) {
        int umur;
        char sim;
        Scanner a=new Scanner(System.in);
        System.out.print("Input umur anda:");
        umur=a.nextInt();
        System.out.print("Apakah Anda sudah 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");
        }
    }
}
```

Gambar 2.3

Langkah-langkah :

- a. Buat *class* baru dengan nama *multif.java* pada *package* *pekan4*.
- b. Deklarasikan variabel umur dengan tipe *int* dan sim dengan tipe *char*.
- c. Buat objek *Scanner* untuk membaca input dari keyboard.
- d. Input umur dari user menggunakan *nextInt()*.
- e. Input status SIM menggunakan *next().charAt(0)*.
- f. Gunakan beberapa *if* untuk menentukan kondisi
- g. Jalankan program dan lakukan uji coba

4. 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.print("Inputkan nilai angka=");
        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");
        }
    }
}
```

Gambar 2.4

Langkah-langkah:

- a. Buat *class* baru dengan nama *Nilai.java* pada *package* *pekan4*.
- b. Deklarasikan variabel nilai dengan tipe data *int*.
- c. Buat objek *Scanner* untuk membaca input dari keyboard.
- d. Tampilkan pesan ke layar: "Inputkan nilai angka=" dan ambil input dari user menggunakan *nextInt()*.
- e. Terapkan percabangan *if-else if-else* untuk menentukan *grade*
- f. Jalankan program dan masukkan berbagai nilai untuk menguji hasil

5. Nama Bulan

```
package pekan4;
import java.util.Scanner;

public class NamaBulan {
    public static void main (String[] args) {
        Scanner scanner=new Scanner(System.in);
        System.out.println("masukkan nama bulan (1-12)=");
        int bulan=scanner.nextInt();
        switch(bulan) {
            case 1:
                System.out.println("Januari");
                break;
            case 2:
                System.out.println("Februari");
                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;
        }
        scanner.close();
    }
}
```

Gambar 2.5

Langkah-langkah

- Buat *class* baru dengan nama NamaBulan.java pada *package* pekan4.
- Import class *Scanner* dari *library* *java.util.Scanner* untuk membaca input dari keyboard.
- Deklarasikan *method* *public static void main(String[] args)*.
- Buat objek *Scanner* untuk membaca input dari user:
- Tampilkan pesan ke layar: "masukkan nama bulan (1-12)=" dan baca input angka dari user dengan *nextInt()*.
- Gunakan struktur *switch-case* untuk mengecek nilai input user (1–12)
- Beri *break*; pada setiap case agar program berhenti setelah menemukan case yang sesuai.
- Tutup objek *Scanner* dengan *scanner.close()*;
- Jalankan program dan masukkan angka dari 1–12 untuk melihat hasil nama bulan yang sesuai.

BAB III

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa penggunaan *conditional statement* sangat membantu dalam menyelesaikan berbagai kasus pemrograman yang membutuhkan pengambilan keputusan. Setiap bentuk percabangan memiliki fungsi masing-masing, mulai dari if yang sederhana hingga switch-case yang lebih terstruktur. Dengan memahami cara kerja percabangan ini, kita bisa menyusun program yang lebih jelas alurnya, mudah dipahami, serta dapat menangani berbagai kondisi dengan tepat.

3.2 Saran

1. Praktikum berikutnya sebaiknya dilengkapi dengan lebih banyak contoh program nyata (kasus sehari-hari) supaya lebih mudah memahami penerapan tipe data dalam menyelesaikan masalah.
2. Materi pembelajaran sebaiknya diberikan sebelum praktikum dimulai, agar dapat dipahami terlebih dahulu, sehingga pada saat praktikum lebih mudah mengerti.

DAFTAR PUSTAKA

- Dokumen Institusi

[1] Departemen Informatika, *Pedoman Penulisan Laporan Praktikum Java*. Padang: Universitas Andalas, 2025.

- Buku

[2] H. M. Deitel and P. J. Deitel, *Java: How to Program*, 11th ed. Pearson, 2018.

[3] W. J. Savitch, *Java: An Introduction to Problem Solving and Programming*, 7th ed. Pearson, 2014.

[4] Wahana Komputer, *Belajar Java untuk Pemula*. Yogyakarta: Andi, 2015

- Sumber daring

[5] Oracle, “The Java Tutorials,” 2023. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/>. [Diakses: 1-Okt-2025].