

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
STATEMENT CONDISIONAL DI JAVA



Oleh :

MUHAMMAD ARIF

NIM 2511532017

MATA KULIAH ALGORITMA PEMROGRAMAN

DOSEN PENGAMPU : DR. WAHYUDI, S.T, M.T

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, 29 SEPTEMBER 2025

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, laporan praktikum **“Statement Condisional di Java ”** ini dapat diselesaikan dengan baik.

Laporan ini disusun sebagai dokumentasi hasil kegiatan praktikum yang bertujuan untuk memahami konsep statement condisional di java dalam pemrograman. Penulis menyadari bahwa pemahaman mendalam terhadap topik ini merupakan fondasi essential bagi pengembangan keterampilan pemrograman yang lebih advanced.

Penulis mengucapkan terima kasih sebesar-besarnya kepada:

1. Bapak DR. Wahyudi, S.T, M.T selaku pembimbing mata kuliah algoritma pemograman, yang telah memberikan bimbingan, ilmu, dan motivasi selama proses pembelajaran.
2. Asisten praktikum yang telah membimbing dan membantu penulis selama sesi praktikum berlangsung.

Padang, 29 September 2025

Muhammad Arif

DAFTAR PUSTAKA

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN.....	i
KATA PENGANTAR	2
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang.....	4
1.2 Tujuan	4
1.3 Manfaat	5
BAB II PEMBAHASAN	6
2.1 Landasan Teori.....	6
2.2 Langkah Pengerjaan.....	7
BAB III KESIMPULAN.....	18
3.1 Kesimpulan	18
3.2 Saran	18
DAFTAR PUSTAKA.....	19

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sebagai mahasiswa pemula di pemrograman, saya menyadari bahwa mempelajari teori algoritma terasa abstrak tanpa praktek langsung. Teori tentang pemecahan masalah baru terasa nyata ketika kita mulai mengetik kode dan melihat hasilnya secara langsung. Dalam dunia nyata, program jarang berjalan secara linear. Program perlu membuat keputusan berdasarkan kondisi tertentu, seperti aplikasi perbankan yang harus memeriksa kecukupan saldo sebelum mengizinkan penarikan uang, atau sistem nilai yang mengkonversi angka menjadi huruf. Kemampuan bercabang inilah yang membuat program menjadi dinamis dan interaktif.

Statement kondisional dalam Java seperti `'if'`, `'if-else'`, dan `'switch-case'` menjawab kebutuhan tersebut dengan memungkinkan pengontrolan alur eksekusi kode berdasarkan evaluasi kondisi. Melalui konstruksi ini, blok kode tertentu hanya akan dijalankan ketika kondisi yang ditetapkan terpenuhi.

Laporan praktikum ini bertujuan menjembatani teori dan praktek melalui percobaan berbagai jenis statement kondisional. Tujuannya untuk memuaskan rasa penasaran tentang bagaimana program membuat keputusan logis, mengubah konsep teoretis menjadi pemahaman aplikatif, serta melatih cara berpikir logis dan terstruktur dalam pemecahan masalah pemrograman. Pemahaman ini diharapkan menjadi fondasi untuk mengembangkan program yang lebih cerdas dan responsif di masa mendatang.

1.2 Tujuan

Tujuan dari dilakukannya praktikum ini adalah

1. Mengetahui dan mengaplikasikan penggunaan statement kondisional di java.
2. Melatih pola pikir logis, sistematis, dan terstruktur dalam merancang solusi untuk permasalahan yang membutuhkan pengambilan keputusan di dalam kode program.
3. Dapat membedakan penggunaan masing-masing jenis percabangan.

1.3 Manfaat

Manfaat yang diperoleh dari praktikum ini adalah

1. Dapat mengaplikasikan penggunaan statement condisional
2. Mendapatkan ilmu tentang statement condisional

BAB II

PEMBAHASAN

2.1 Landasan Teori

Pemrograman adalah kegiatan membuat program komputer dengan menulis instruksi-instruksi menggunakan bahasa pemrograman seperti Java. Instruksi-instruksi ini berisi perintah yang harus dijalankan komputer untuk menyelesaikan tugas tertentu. Secara sederhana, pemrograman seperti memberi tahu komputer langkah demi langkah apa yang harus dilakukan, sama seperti ketika kita memberikan arahan detail kepada seseorang untuk menyelesaikan suatu pekerjaan. Hasil akhirnya adalah program atau aplikasi yang dapat digunakan untuk memecahkan masalah sehari-hari.

Statement kondisional merupakan struktur kendali dalam pemrograman yang memungkinkan program untuk mengeksekusi blok kode tertentu berdasarkan kondisi yang diberikan. Menurut WSCube Tech (2023), statement kondisional dalam Java berfungsi untuk mengontrol alur eksekusi program dengan mengevaluasi ekspresi boolean yang menghasilkan nilai true atau false. Mekanisme ini memungkinkan program untuk membuat keputusan secara dinamis sesuai dengan situasi yang dihadapi.

Jenis-Jenis Statement Kondisional dalam Java

- Statement if

Statement if merupakan bentuk paling dasar dari percabangan. statement if akan mengeksekusi blok kode di dalamnya hanya jika kondisi yang diberikan bernilai true.

```
if(kondisi) {  
    // blok kode yang dijalankan jika kondisi true  
}
```

- Statement if else

Statement if-else menyediakan alternatif eksekusi ketika kondisi tidak terpenuhi. Statement ini memberikan dua keputusan eksekusi berdasarkan hasil kondisi.

```
if(kondisi) {  
    // blok kode jika kondisi true  
} else {  
    // blok kode jika kondisi false  
}
```

- Statement if else if

Statement if-else-if digunakan ketika terdapat multiple kondisi yang perlu dievaluasi. Ini biasanya digunakan ketika ada lebih dari dua kondisi yang harus dieksekusi.

```
if(kondisi1) {  
    // blok kode 1  
} else if(kondisi2) {  
    // blok kode 2  
} else {  
    // blok kode default  
}
```

- Statement switch case

Statement switch-case merupakan alternatif untuk if-else-if ketika berhadapan dengan banyak kondisi. Statement switch cocok digunakan ketika nilai yang dievaluasi memiliki kemungkinan tetap dan diketahui sebelumnya.

```
switch (ekspresi) {  
    case value1:  
        // blok kode 1  
        break;  
    case value2:  
        // blok kode 2  
        break;  
    default:  
        // blok kode default  
}
```

2.2 Langkah Pengerjaan

a. Statement IF

1. Buat package dan *class* baru untuk pembuatan file yang bisa dilakukan untuk operasi pemrograman Java.
2. Setelah *class* dibuat, ketikkan **import** *java.util.Scanner*. agar java mengetahui bahwa kita menggunakan *Scanner* pada pemrograman.
3. *Scanner* keyboard = **new** *Scanner* (*System.in*); digunakan untuk membaca input dari keyboard.
4. Disini saya membuat menentukan kelulusan berdasarkan IPK yang diinputkan user. Jika IPK lebih dari 2.5 maka output “anda lulus sangat memuaskan dengan IPK + IPK yang diinputkan.
5. Karena kondisi yang diberikan Cuma satu ,kita hanya menggunakan If sebagai statement.

```

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 2.2. 1

6. Output yang dihasilkan adalah :

```

Input IPK anda =
4
Anda Lulus Sangat Memuaskan dengan IPK = 4.0

```

Gambar 2.2. 2

7. Jika kita memasukkan IPK dengan nilai 2.5 atau lebih kecil maka outputnya akan seperti gambar di bawah ini :

```

input IPK anda =
2
|

```

Gambar 2.2. 3

8. Hal ini terjadi karena kita hanya menggunakan statement If pada program. Jika nilai yang dimasukkan lebih kecil yang berarti itu false dari kondisi yang diberikan maka program langsung selesai.

b. Statement IF-ELSE

1. Buat *class* baru untuk pembuatan file yang bisa dilakukan untuk operasi pemrograman Java.
2. Setelah *class* dibuat, ketikkan **import** java.util.Scanner. agar java mengetahui bahwa kita menggunakan *Scanner* pada pemrograman.
3. *Scanner* keyboard = **new** *Scanner* (System.in); digunakan untuk membaca input dari keyboard.
4. Disini saya membuat menentukan kelulusan berdasarkan IPK yang diinputkan user. Jika IPK lebih dari 2.5 maka output “anda lulus sangat memuaskan dengan

IPK + IPK yang diinputkan dan jika IPK lebih kecil dari 2.5 maka output “anda tidak lulus”

5. Karena kondisi ada dua maka kita menggunakan If-Else sebagai statement.

```
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. 4

6. Jika kita input nilai lebih besar dari 2.5 maka output :

```
Input IPK anda =
4
Anda Lulus Sangat Memuaskan dengan IPK = 4.0
```

Gambar 2.2. 5

7. Jika menginputkan nilai lebih kecil dari 2.5 maka tidak terpenuhi, program akan melanjutkan ke baris berikutnya

```
Input IPK anda =
2
Anda tidak Lulus
```

Gambar 2.2. 6

c. Statement IF-ELSE-IF

1. Buat *class* baru untuk pembuatan file yang bisa dilakukan untuk operasi pemrograman Java.
2. Setelah *class* dibuat, ketikkan **import** java.util.Scanner. agar java mengetahui bahwa kita menggunakan Scanner pada pemrograman.
3. Scanner keyboard = **new** Scanner (System.in); digunakan untuk membaca input dari keyboard.
4. Disini saya membuat menentukan nilai berdasarkan angka yang diinputkan

```

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 angka= ");
        nilai=input.nextInt();
        input.close();

        if(nilai >=80) {
            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 2.2. 7

5. Jika saya memasukkan angka ≥ 80 maka output :

```

Inputkan nilai angka=
100
A

```

Gambar 2.2. 8

6. Jika saya memasukkan angka ≥ 70 maka output :

```

Inputkan nilai angka=
70
B

```

Gambar 2.2. 9

7. Jika saya memasukkan angka ≥ 60 maka output :

```

Inputkan nilai angka=
60
C

```

Gambar 2.2. 10

8. Jika saya memasukkan angka ≥ 50 maka output :

```
Inputkan nilai angka=  
50  
D
```

Gambar 2.2. 11

9. Jika saya memasukkan angka ≤ 50 maka output :

```
Inputkan nilai angka=  
40  
E
```

Gambar 2.2. 12

d. Statement multiIF

1. Buat *class* baru untuk pembuatan file yang bisa dilakukan untuk operasi pemrograman Java.
2. Setelah *class* dibuat, ketikkan **import** java.util.Scanner. agar java mengetahui bahwa kita menggunakan *Scanner* pada pemrograman.
3. *Scanner* keyboard = **new** *Scanner* (System.in); digunakan untuk membaca input dari keyboard.
4. Disini saya membuat menentukan kelayakan mengemudi berdasarkan umur dan SIM
5. Program ini menggunakan empat kondisi IF independen yang mengevaluasi semua kombinasi usia dan kepemilikan SIM.

```

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.println("Input umur anda:");
        umur = a.nextInt();
        System.out.println("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 membawa 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 2.2. 13

6. Jika kita memasukkan umur ≥ 17 dan SIM = y maka output :

```

Input umur anda:
18
Apakah anda sudah punya sim C:
y
Anda sudah dewasa dan boleh bawa motor

```

Gambar 2.2. 14

7. Jika kita memasukkan umur ≥ 17 dan SIM != y maka output :

```

Input umur anda:
18
Apakah anda sudah punya sim C:
n
Anda sudah dewasa tetapi tidak boleh membawa motor

```

Gambar 2.2. 15

8. Jika kita memasukkan umur ≤ 17 dan SIM = y maka output :

```
Input umur anda:  
16  
Apakah anda sudah punya sim C:  
y  
Anda belum cukup umur punya SIM
```

Gambar 2.2. 16

9. Jika kita memasukkan umur ≤ 17 dan SIM $\neq y$ maka output :

```
Input umur anda:  
16  
Apakah anda sudah punya sim C:  
n  
Anda belum cukup umur bawa motor
```

Gambar 2.2. 17

10. Pada program tersebut menggunakan multiIF karena setiap kondisi memiliki output yang berbeda, bisa saja menggunakan if-else tetapi lebih kompleks.

e. Switch – Case

1. Buat *class* baru untuk pembuatan file yang bisa dilakukan untuk operasi pemrograman Java.
2. Setelah *class* dibuat, ketikkan **import** *java.util.Scanner*. agar java mengetahui bahwa kita menggunakan *Scanner* pada pemrograman.
3. *Scanner* keyboard = **new** *Scanner* (*System.in*); digunakan untuk membaca input dari keyboard.
4. Disini saya membuat menentukan nama bulan berdasarkan angka yang diinputkan dari 1-12
5. Program ini menggunakan statement Switch-Case

```

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 angka 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");

```

Gambar 2.2. 18

```

11         break;
12     case 11:
13         System.out.println("November");
14         break;
15     case 12 :
16         System.out.println("Desember");
17         break;
18     default :
19         System.out.println("Angka tidak valid");
20     }
21     scanner.close();
22 }
23
24 }
25

```

Gambar 2.2. 19

6. Output yang dihasilkan jika memasukkan angka 1 :

```
Masukkan angka bulan (1-12):  
1  
Januari
```

Gambar 2.2. 20

7. Output yang dihasilkan jika memasukkan angka 2 :

```
Masukkan angka bulan (1-12):  
2  
Februari
```

Gambar 2.2. 21

8. Output yang dihasilkan jika memasukkan angka 3 :

```
Masukkan angka bulan (1-12):  
3  
Maret
```

Gambar 2.2. 22

9. Output yang dihasilkan jika memasukkan angka 4 :

```
Masukkan angka bulan (1-12):  
4  
April
```

Gambar 2.2. 23

10. Output yang dihasilkan jika memasukkan angka 5 :

```
Masukkan angka bulan (1-12):  
5  
Mei
```

Gambar 2.2. 24

11. Output yang dihasilkan jika memasukkan angka 6 :

```
Masukkan angka bulan (1-12):  
6  
Juni
```

Gambar 2.2. 25

12. Output yang dihasilkan jika memasukkan angka 7 :

```
Masukkan angka bulan (1-12):  
7  
Juli
```

Gambar 2.2. 26

13. Output yang dihasilkan jika memasukkan angka 8 :

```
Masukkan angka bulan (1-12):  
8  
Agustus
```

Gambar 2.2. 27

14. Output yang dihasilkan jika memasukkan angka 9 :

```
Masukkan angka bulan (1-12):  
9  
September
```

Gambar 2.2. 28

15. Output yang dihasilkan jika memasukkan angka 10 :


```
Masukkan angka bulan (1-12):  
10  
Oktober
```

Gambar 2.2. 29

16. Output yang dihasilkan jika memasukkan angka 11 :

```
Masukkan angka bulan (1-12):  
11  
November
```

Gambar 2.2. 30

17. Output yang dihasilkan jika memasukkan angka 12 :

```
Masukkan angka bulan (1-12):  
12  
Desember
```

Gambar 2.2. 31

18. Output yang dihasilkan jika memasukkan angka 13 :

```
Masukkan angka bulan (1-12):  
13  
Angka tidak valid
```

Gambar 2.2. 32

19. Switch-Case digunakan karena banyak kondisi yang harus ditentukan, selain itu Switch-Case lebih rapi.

BAB III

KESIMPULAN

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan mengenai statement kondisional di Java, dapat disimpulkan bahwa:

1. **Statement IF** merupakan dasar percabangan yang hanya mengeksekusi blok kode ketika kondisi bernilai true, namun tidak menangani kondisi false sehingga program dapat berakhir tanpa output tertentu.
2. **Statement IF-ELSE** memberikan solusi lebih komprehensif dengan menyediakan alternatif eksekusi untuk kedua kemungkinan kondisi (true dan false), membuat program lebih responsif terhadap berbagai skenario input.
3. **Statement IF-ELSE-IF** efektif untuk menangani multiple kondisi yang saling berkaitan, memungkinkan evaluasi berjenjang hingga menemukan kondisi yang terpenuhi.
4. **Statement Multi IF** cocok digunakan ketika terdapat beberapa kondisi independen yang masing-masing memerlukan penanganan khusus, seperti pada program pengecekan kelayakan mengemudi berdasarkan usia dan kepemilikan SIM.
5. **Statement SWITCH-CASE** merupakan pilihan ideal untuk mengevaluasi satu variabel terhadap banyak nilai tetap yang diketahui sebelumnya, memberikan struktur kode yang lebih rapi dan mudah dibaca dibandingkan multiple IF-ELSE.

3.2 Saran

Untuk meningkatkan pemahaman yang lebih mendalam tentang statement kondisional, maka dari itu ada beberapa saran yang bisa dilakukan:

- **Eksperimen Lanjutan:** Mencoba implementasi statement kondisional dalam skenario yang lebih kompleks seperti sistem login multi-level atau kalkulator scientific dengan berbagai operasi.

Masih banyak kekurangan dalam pembuatan laporan praktek ini, maka dari itu penulis sangat terbuka untuk menerima saran dari pembaca sekalian.

DAFTAR PUSTAKA

- [1] WSCube Tech, "Conditional Statements in Java," 2023. [Online]. Available: <https://www.wscubetech.com/resources/java/conditional-statements>. [Accessed: Sept. 29, 2025].
- [2] Smart Programming, "Java Conditional Statements: If-Else & Switch," 2024. [Online]. Available: <https://smartprogramming.in/tutorials/java/java-conditional-statements-if-else-switch>. [Accessed: Sept. 29, 2025].