

# **LAPORAN PRATIKUM ALGORITMA PEMROGRAMAN**

## **STATEMENT CONDISIONAL**

Disusun Oleh:

Endy Pardilian 2511531017

Dosen Pengampu:

Wahyudi. Dr., S.T,M.T

Asisten Pratikum:

Aufan Taufiqurrahman



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG**

**2025**

## **KATA PENGANTAR**

Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma pemrograman yang membahas mengenai Statement Condisional dalam bahasa pemrograman java. Melalui laporan ini, penulis dapat memahami materi pratikum secara mendalam. Penulisan laporan ini juga dapat melatih ketelitian, keteraturan, serta kemampuan menulis sesuai kaidah akademik. Dengan demikian, laporan pratikum ini dapat berfungsi sebagai sarana belajar, dokumentasi kegiatan, dan referensi untuk praktikum atau pembelajaran jenjang berikutnya.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan, baik dari isi maupun penyajiannya. Oleh sebab itu, saran dan kritik sangat penulis harapkan untuk laporan berikutnya.

Padang, 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>i</b>
<b>DAFTAR ISI .....</b>	<b>ii</b>
<b>BAB I PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan.....	1
1.3 Manfaat.....	1
<b>BAB II PEMBAHASAN .....</b>	<b>2</b>
2.1 Program IPK Lulus ( <i>If</i> ) <sup>2</sup> .....	2
2.2 Program IPK Lulus atau Tidak ( <i>If-Else</i> ) .....	3
2.3 Program SIM & Umur ( <i>If</i> Ganda / <i>Multi If</i> ).....	4
2.4 Program Penilaian Huruf ( <i>If-Else-If</i> ).....	5
2.5 Program Menentukan Nama Bulan ( <i>Switch-Case</i> ) .....	7
<b>BAB III KESIMPULAN .....</b>	<b>9</b>
3.1 Kesimpulan.....	9
3.2 Saran.....	9
<b>DAFTAR PUSTAKA.....</b>	<b>10</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam pemrograman, pengambilan keputusan merupakan salah satu konsep dasar. Statement Conditional digunakan Ketika suatu instruksi harus dijalankan hanya jika memenuhi kondisi tertentu.

Beberapa bentuk statement condisional yang digunakan dalam Java antara lain *if*, *if-else*, *if-else-if*, dan *switch-case*. Setiap bentuk memiliki kegunaan masing masing, missal *if-else* cocok untuk percabangan sederhana, sedangkan *switch-case* digunakan saat banyak pilihan kondisi.

Melalui praktikum ini, mahasiswa diharapkan mampu memahami konsep dasar statement condisional dan mampu mengimplementasikannya ke dalam kode program Java sesuai dengan kebutuhan.

### **1.2 Tujuan**

1. Mengetahui konsep dasar statement conditional di Java.
2. Mempelajari perbedaan antara *if*, *if-else*, *if-else-if*, dan *switch-case*.
3. Mengimplementasikan statement conditional dalam bentuk program.
4. Melatih kemampuan analisis dalam menentukan struktur percabangan yang tepat untuk menyelesaikan suatu permasalahan.

### **1.3 Manfaat**

1. Memberikan pemahaman mengenai penggunaan statement conditional di Java.
2. Membantu mahasiswa dalam mengasah logika pemrograman melalui percabangan kondisi.
3. Menjadi dasar untuk mempelajari struktur program yang lebih kompleks seperti perulangan dan algoritma.

## BAB II

### PEMBAHASAN

#### 2.1 Program IPK Lulus (*If*)

Statement *if* digunakan untuk melakukan pengujian terhadap suatu kondisi tertentu, kemudian menjalankan perintah hanya jika kondisi tersebut bernilai benar (true). Jika kondisi tidak terpenuhi, maka perintah di dalam blok *if* akan diabaikan dan program melanjutkan eksekusi ke baris berikutnya. Contoh sederhana adalah ketika kita ingin mengecek apakah seorang mahasiswa lulus berdasarkan nilai ujian. Jika nilai yang diperoleh lebih besar atau sama dengan 60, maka program akan menampilkan pesan “Lulus”. Struktur ini sangat berguna ketika hanya ada satu kondisi yang ingin diperiksa tanpa perlu mempertimbangkan alternatif lain. Dengan demikian, *if* biasanya dipakai untuk keputusan sederhana dan langsung.

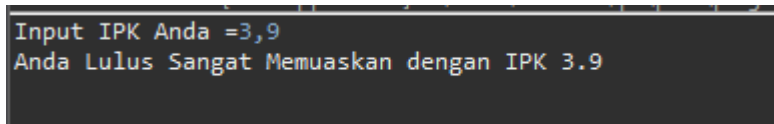
```
1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class latIf1 {
6     public static void main (String [] args) {
7         double IPK;
8         Scanner input = new Scanner(System.in);
9         System.out.print("Input IPK Anda =");
10        IPK = input.nextDouble();
11        input.close();
12        if (IPK>2.75) {
13            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK " + IPK);
14        }
15    }
16 }
17 }
18 }
```

Code 2.1

Nama file: latIf1.java

1. Program meminta input berupa IPK (angka desimal) dari pengguna.
2. Nilai IPK disimpan ke variabel IPK.
3. Program menggunakan *if* tunggal untuk memeriksa apakah  $IPK > 2.75$ .
4. Jika benar, maka akan tampil pesan "Anda Lulus Sangat Memuaskan dengan IPK ...".
5. Jika salah, tidak ada output tambahan (karena hanya menggunakan *if*).

Output:

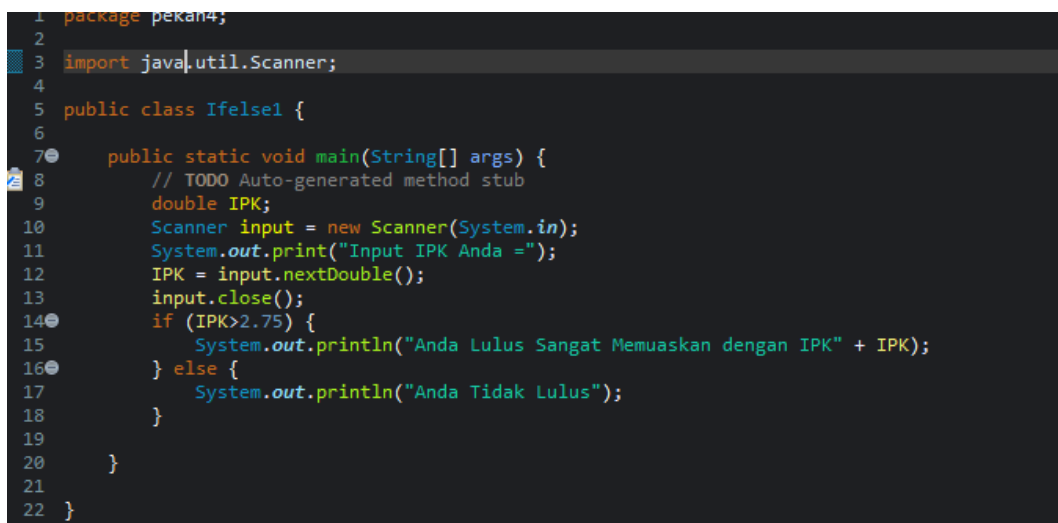


```
Input IPK Anda =3,9
Anda Lulus Sangat Memuaskan dengan IPK 3.9
```

Gambar 2.1

## 2.2 Program IPK Lulus atau Tidak (*If-Else*)

Berbeda dengan *if* tunggal, struktur *if-else* memberikan dua kemungkinan jalur eksekusi. Jika kondisi yang diuji bernilai benar, maka blok kode di dalam *if* akan dijalankan, namun jika kondisi bernilai salah, maka blok kode pada bagian *else* akan dijalankan sebagai alternatif. Misalnya, saat memeriksa nilai mahasiswa, jika nilainya lebih besar atau sama dengan 60 maka ditampilkan “Lulus”, sedangkan jika nilainya kurang dari 60 maka ditampilkan “Tidak Lulus”. Struktur ini sangat berguna ketika program harus memilih antara dua kondisi yang saling bertolak belakang. Dengan *if-else*, program menjadi lebih fleksibel dalam memberikan respon yang berbeda terhadap input pengguna.



```
1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class Ifelse1 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         double IPK;
10        Scanner input = new Scanner(System.in);
11        System.out.print("Input IPK Anda =");
12        IPK = input.nextDouble();
13        input.close();
14        if (IPK>2.75) {
15            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK" + IPK);
16        } else {
17            System.out.println("Anda Tidak Lulus");
18        }
19    }
20 }
21
22 }
```

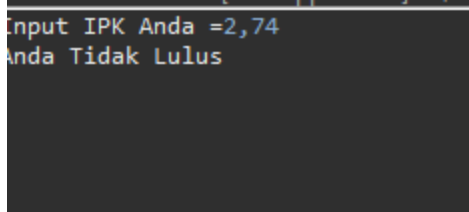
Code 2.2

Nama file: Ifelse1.java

1. Program meminta input IPK dari pengguna.
2. Nilai IPK disimpan dalam variabel IPK.
3. Program menggunakan if-else untuk membuat dua kemungkinan:
  - a. Jika  $IPK > 2.75$ , maka menampilkan "Anda Lulus Sangat Memuaskan dengan IPK ...".

- b. Jika  $IPK \leq 2.75$ , maka menampilkan "Anda Tidak Lulus".
4. Program ini lebih lengkap daripada sebelumnya karena menangani dua kondisi sekaligus.

Output:



```
Input IPK Anda =2,74
Anda Tidak Lulus
```

Gambar 2.2

### 2.3 Program SIM & Umur ( *Multi If* )

```
1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class multiIf {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         int umur;
10        char sim;
11        Scanner a = new Scanner(System.in);
12        System.out.print("Input umur anda:");
13        umur = a.nextInt();
14        System.out.print("Apakah Anda Sudah Punya SIM C : ");
15        sim=a.next().charAt(0);
16        a.close();
17        if((umur >= 17)&&(sim=='y')) {
18            System.out.println("Anda Sudah dewasa dan boleh bawa motor");
19        }
20        if((umur >= 17)&&(sim!='y')) {
21            System.out.println("Anda Sudah dewasa tetapi tidak boleh bawa motor");
22        }
23        if((umur <= 17)&&(sim!='y')) {
24            System.out.println("Anda Belum Cukup Umur bawa motor");
25        }
26        if((umur <= 17)&&(sim=='y')) {
27            System.out.println("Anda Belum Cukup Umur punya SIM");
28        }
29    }
30 }
31 }
```

Code 2.3

Nama file: multif.java

1. Program meminta dua input dari pengguna:
  - a. Umur (bilangan bulat).
  - b. Kepemilikan SIM C (karakter y untuk ya, Y untuk tidak).
2. Input disimpan dalam variabel umur dan sim.

3. Program menggunakan if terpisah untuk mengecek kondisi kombinasi:
  - a. Jika umur  $\geq 17$  dan sudah punya SIM  $\rightarrow$  "Anda Sudah dewasa dan boleh bawa motor".
  - b. Jika umur  $\geq 17$  dan tidak punya SIM  $\rightarrow$  "Anda Sudah dewasa tetapi tidak boleh bawa motor".
  - c. Jika umur  $< 17$  dan punya SIM  $\rightarrow$  "Anda Belum Cukup Umur bawa motor".
  - d. Jika umur  $< 17$  dan tidak punya SIM  $\rightarrow$  "Anda Belum Cukup Umur punya SIM".
4. Program ini menunjukkan penggunaan operator logika  $\&\&$  (AND) untuk menggabungkan dua syarat.

Output:

```
Input umur anda:18
Apakah Anda Sudah Punya SIM C : Y
Anda Sudah dewasa tetapi tidak boleh bawa motor
```

Gambar 2.3

#### 2.4 Program Penilaian Huruf (*If-Else-If*)

Struktur *if-else-if* digunakan untuk menangani situasi yang lebih kompleks, ketika ada lebih dari dua kemungkinan kondisi yang harus diperiksa. Pada struktur ini, kondisi diperiksa secara bertingkat dari atas ke bawah. Jika kondisi pertama tidak terpenuhi, maka program akan memeriksa kondisi berikutnya, dan seterusnya hingga salah satu kondisi bernilai benar. Jika tidak ada kondisi yang benar, maka blok else (jika ada) akan dijalankan. Contoh penerapan umum adalah pada konversi nilai angka menjadi nilai huruf, misalnya 80 ke atas menjadi "A", 70–79 menjadi "B", 60–69 menjadi "C", dan seterusnya. Dengan adanya *if-else-if*, program dapat mengatur logika pengambilan keputusan yang lebih rumit secara berurutan dan sistematis.



```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class Nilai {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         int nilai;
10        Scanner input=new Scanner(System.in);
11        System.out.print("Inputkan nilai angka =" );
12        nilai=input.nextInt();
13        input.close();
14
15        if(nilai >= 81) {
16            System.out.println("A");
17        } else if (nilai >= 70) {
18            System.out.println("B");
19        } else if (nilai >= 60) {
20            System.out.println("C");
21        } else if (nilai >= 50) {
22            System.out.println("D");
23        } else {
24            System.out.println("E");
25        }
26
27    }
28
29 }

```

Code 2.4

Nama file: Nilai.java

1. Program ini meminta input berupa angka nilai dari pengguna.
2. Nilai yang dimasukkan akan disimpan dalam variabel nilai.
3. Program menggunakan struktur *if-else-if* untuk menentukan nilai huruf:
  - a. Jika nilai  $\geq 81$  maka output "A".
  - b. Jika nilai  $\geq 70$  maka output "B".
  - c. Jika nilai  $\geq 60$  maka output "C".
  - d. Jika nilai  $\geq 50$  maka output "D".
  - e. Selain itu, maka output "E".
4. Tujuan program adalah mengonversi nilai angka menjadi nilai huruf sesuai standar tertentu.

Output:

```

Inputkan nilai angka =75
B

```

Gambar 2.4

## 2.5 Program Menentukan Nama Bulan (*Switch-Case*)

Selain menggunakan *if-else-if*, Java juga menyediakan struktur *switch-case* untuk menangani percabangan dengan banyak pilihan. Perbedaannya, *switch-case* lebih efisien dan rapi digunakan ketika program harus memilih salah satu dari banyak kemungkinan berdasarkan nilai tetap, seperti angka atau string. Misalnya, ketika pengguna memasukkan angka 1 sampai 12 untuk mewakili bulan, maka *switch-case* dapat langsung menentukan nama bulan yang sesuai. Jika input tidak valid, maka bagian default akan dijalankan untuk memberikan pesan kesalahan. Keunggulan *switch-case* dibanding *if-else-if* adalah lebih mudah dibaca dan dipelihara, terutama jika jumlah kondisi yang diperiksa cukup banyak dan berbentuk konstan.

```
1 package pekan4;
2 import java.util.Scanner;
3 public class NamaBulan {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner scanner = new Scanner(System.in);
8         System.out.print("Masukan angka bulan (1 - 12): ");
9         int bulan = scanner.nextInt();
10        switch (bulan) {
11            case 1:
12                System.out.println("Januari");
13                break;
14            case 2:
15                System.out.println("Februari");
16                break;
17            case 3:
18                System.out.println("Maret");
19                break;
20            case 4:
21                System.out.println("April");
22                break;
23            case 5:
24                System.out.println("Mei");
25                break;
26            case 6:
27                System.out.println("Juni");
28                break;
29            case 7:
30                System.out.println("Juli");
31                break;
32            case 8:
33                System.out.println("Agustus");
34                break;
35            case 9:
36                System.out.println("September");
37                break;
38            case 10:
39                System.out.println("Oktober");
40                break;
41            case 11:
42                System.out.println("November");
43                break;
44            case 12:
45                System.out.println("Desember");
46                break;
47            default:
48                System.out.println("Angka tidak valid");
49        }
50        scanner.close();
51    }
52 }
53
54 }
```

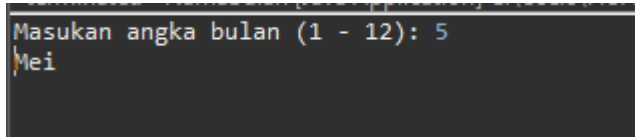
Code 2.5

Nama file: NamaBulan.java

1. Program meminta input berupa angka dari 1–12.
2. Angka tersebut mewakili bulan, lalu disimpan dalam variabel bulan.
3. Program menggunakan struktur switch-case untuk menentukan nama bulan:

- a. 1 untuk Januari, 2 untuk Februari, hingga 12 untuk Desember.
  - b. Jika angka yang dimasukkan tidak valid (misalnya 13), maka blok default akan dijalankan dan menampilkan "Angka tidak valid".
4. Program ini lebih efisien dibandingkan *if-else* ketika ada banyak pilihan kondisi.

Output:



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

Gambar 2.5

## BAB III

### KESIMPULAN

#### 3.1 Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa statement conditional di Java berfungsi untuk mengatur alur program sesuai kondisi tertentu. Struktur *if*, *if-else*, *if-else-if*, dan *switch-case* memiliki kegunaan masing-masing: *if* untuk pengecekan sederhana, *if-else* untuk dua kondisi, *if-else-if* untuk banyak syarat bertingkat, dan *switch-case* untuk pilihan yang lebih banyak dan terstruktur. Selain itu, operator logika juga dapat digunakan untuk menggabungkan beberapa kondisi sekaligus. Dengan memahami percabangan, mahasiswa dapat membuat program yang lebih dinamis, logis, dan sesuai kebutuhan.

#### 3.2 Saran

Sebagai saran, sebaiknya penjelasan materi saat praktikum bisa lebih detail dan perlahan, supaya mahasiswa yang belum terlalu paham coding bisa mengikuti dengan baik. Selain itu, dosen atau asisten praktikum sebaiknya memberikan kriteria pelaksanaan tugas yang lebih jelas, agar mahasiswa mengetahui apa saja yang harus dikerjakan dan bagaimana penilaiannya.

## DAFTAR PUSTAKA

- [1] Oracle, “The Java™ Tutorials: Control Flow Statements,” Oracle. [Online].  
Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html>