

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN**

**“laporan Praktikum Pekan 4”**

Disusun Oleh:

Faiz Fikri Satria

2511533026

Dosen Pengampu : Dr. Wahyudi, S.T, M.T.  
Asisten Praktikum : Rahmad Dwirizki Olders



DAPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS

2025

## **KATA PENGANTAR**

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga dapat menyelesaikan laporan praktikum ini dengan baik. Tidak lupa juga saya ucapkan terima kasih kepada bapak Wahyudi. Dr. S.T.M.T sebagai dosen pembimbing dan Rahmad Dwirizki Olders yang telah membantu dalam pelaksanaan praktikum ini. Laporan ini saya susun untuk memenuhi tugas praktikum Algoritma dan Pemrograman dari praktikum pertemuan pertama mengenai pemahaman dasar pemrograman bahasa Java menggunakan IDE Eclipse. Praktikum ini bertema pembuatan kode dasar, yaitu penggunaan If, Else, dan Switch, Case. Saya berharap laporan ini dapat memberikan manfaat bagi pembaca dalam memahami konsep dasar pemrograman Java.

Laporan ini dibuat dengan harapan dapat memberikan pemahaman dasar mengenai penggunaan Bahasa pemrograman Java, khususnya mengenai hal struktur dasar program dan fungsi.

Saya menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu, kritik dan saran sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang 04 Oktober, 2025

Penulis

## DAFTAR ISI

|                                 |     |
|---------------------------------|-----|
| KATA PENGANTAR .....            | ii  |
| DAFTAR ISI .....                | iii |
| BAB I PENDAHULUAN .....         | 1   |
| 1.1 Latar Belakang .....        | 1   |
| 1.2 Tujuan .....                | 1   |
| 1.3 Manfaat .....               | 2   |
| BAB II PEMBAHASAN .....         | 3   |
| 2.1 If Else .....               | 3   |
| 2.1.1 Langkah-langkah .....     | 3   |
| 2.1.2 Contoh Pemrograman .....  | 3   |
| 2.1.3 Hasil Output .....        | 4   |
| 2.1.4 Analisis .....            | 4   |
| 2.1.5 Teori Mengenai Kode ..... | 5   |
| 2.2 Lat If .....                | 5   |
| 2.2.1 Langkah-langkah .....     | 5   |
| 2.2.2 Contoh Pemrograman .....  | 6   |
| 2.2.3 Hasil Output .....        | 6   |
| 2.2.4 Analisis .....            | 6   |
| 2.2.5 Teori Mengenai Kode ..... | 7   |
| 2.3 Multi If .....              | 7   |
| 2.2.1 Langkah-langkah .....     | 7   |

|                                 |           |
|---------------------------------|-----------|
| 2.3.2 Contoh Pemrograman .....  | 8         |
| 2.3.3 Hasil Output .....        | 9         |
| 2.3.4 Analisis .....            | 9         |
| 2.3.5 Teori Mengenai Kode ..... | 9         |
| <b>2.4 Nama Bulan .....</b>     | <b>9</b>  |
| 2.4.1 Langkah-langkah .....     | 10        |
| 2.4.2 Contoh Pemrograman .....  | 11        |
| 2.4.3 Hasil Output .....        | 12        |
| 2.4.4 Analisis .....            | 12        |
| 2.4.5 Teori Mengenai Kode ..... | 12        |
| <b>2.5 Penilaian.....</b>       | <b>12</b> |
| 2.5.1 Langkah-langkah .....     | 13        |
| 2.5.2 Contoh Pemrograman .....  | 14        |
| 2.5.3 Hasil Output .....        | 14        |
| 2.5.4 Analisis .....            | 15        |
| 2.5.5 Teori Mengenai Kode ..... | 15        |
| <b>BAB III KESIMPULAN .....</b> | <b>16</b> |
| <b>DAFTAR PUSTAKA .....</b>     | <b>17</b> |

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Praktikum pertemuan keempat ini dirancang untuk memperkenalkan mahasiswa pada struktur kontrol alur program, seperti if-else dan variasinya, yang menjadi dasar dalam membangun aplikasi interaktif. Tema ini mencakup penerapan if-else sederhana, single if, multi if, switch untuk nama bulan, dan penilaian nilai, menggunakan Eclipse sebagai IDE untuk memfasilitasi pengkodean dan debugging. Latar belakang ini timbul dari kebutuhan memahami bagaimana program dapat merespons input pengguna secara dinamis sebelum memasuki topik yang lebih lanjut.

### **1.2 Tujuan**

1. Memahami dan menerapkan struktur if-else dalam program Java untuk pengkondisian sederhana.
2. Mengimplementasikan single if untuk kondisi tunggal tanpa alternatif.
3. Menggunakan multi if untuk mengevaluasi beberapa kondisi independen.
4. Menerapkan switch case untuk menangani kasus berbasis nilai, seperti nama bulan.
5. Melatih if-else ladder untuk penilaian bertingkat dan penggunaan Eclipse dalam pengembangan kode.

### **1.3 Manfaat**

1. Meningkatkan kemampuan mahasiswa dalam menulis kode Java dengan logika kondisional yang benar.
2. Memberikan pemahaman praktis tentang alur program berdasarkan input pengguna.
3. Memperkuat keterampilan debugging dan analisis kondisi menggunakan Eclipse.
4. Menjadi fondasi untuk program kompleks seperti sistem pengambilan keputusan atau aplikasi berbasis menu.
5. Mendorong pemikiran logis dalam merancang solusi pemrograman berbasis kondisi.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 If Else**

Program ini mendemonstrasikan penggunaan struktur if-else dalam Java untuk mengevaluasi nilai IPK yang dimasukkan pengguna melalui Scanner, di mana jika IPK lebih besar dari 2.75, program akan mencetak pesan lulus sangat memuaskan beserta nilai IPK, sedangkan jika tidak, akan mencetak pesan tidak lulus, sehingga membantu memahami pengambilan keputusan biner dalam pemrograman dasar yang berguna untuk validasi sederhana.

##### **2.1.1 Pembahasan Langkah**

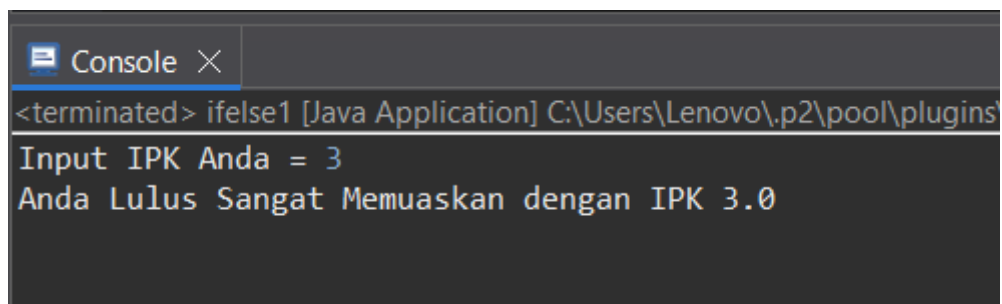
- Buat package baru bernama "Pekan\_empat" di Eclipse.
- Buat class baru bernama "ifelse1" dalam package tersebut.
- Import library java.util.Scanner untuk menerima input dari pengguna.
- Deklarasikan variabel double IPK di dalam method main.
- Inisialisasi objek Scanner dengan nama input.
- Tampilkan prompt "Input IPK Anda = " dan baca nilai IPK menggunakan `input.nextDouble()`.
- Tutup objek Scanner dengan `input.close()`.
- Gunakan if ( $IPK > 2.75$ ) untuk mencetak "Anda Lulus Sangat Memuaskan dengan IPK " + IPK.
- Gunakan else untuk mencetak "Anda Tidak Lulus".
- Jalankan program di Eclipse dan uji dengan input contoh seperti 3.0 atau 2.5.

### 2.1.2 Contoh Pemrograman

```
1 package Pekan_empat;
2
3 import java.util.Scanner;
4 public class ifelse1 {
5     public static void main(String[] args) {
6         double IPK;
7         Scanner input=new Scanner (System.in);
8         System.out.print("Input IPK Anda = ");
9         IPK=input.nextDouble();
10        input.close();
11        if (IPK>2.75) {
12            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
13        } else {
14            System.out.println("Anda TIdak Lulus");
15        }
16    }
17 }
```

### 2.1.3 Hasil Output

Hello World



```
<terminated> ifelse1 [Java Application] C:\Users\Lenovo\p2\pool\plugins
Input IPK Anda = 3
Anda Lulus Sangat Memuaskan dengan IPK 3.0
```

### 2.1.4 Analisis

Program ini berhasil mengevaluasi kondisi biner dengan benar, tetapi tidak menangani input non-numerik yang dapat menyebabkan exception. Analisis menunjukkan efisiensi untuk keputusan sederhana, meskipun typo pada output "TIdak" perlu diperbaiki. Jika IPK tepat 2.75, program masuk ke else, yang mungkin perlu disesuaikan menjadi  $\geq$  untuk inklusif.



### **2.1.5 Teori Mengenai Kode**

Struktur if-else dalam Java digunakan untuk eksekusi kondisional, di mana blok if dieksekusi jika kondisi true, dan else jika false. Teori ini berdasarkan pada logika boolean, dengan operator relasional seperti > untuk perbandingan. If-else meningkatkan fleksibilitas program dalam menangani alternatif.

## **2.2 Lat If**

Program ini mengilustrasikan penggunaan single if dalam Java, di mana pengguna memasukkan nilai IP melalui Scanner, dan jika nilai tersebut lebih besar dari 2.75, program mencetak pesan lulus sangat memuaskan beserta nilai IP, tanpa blok else untuk kasus gagal, sehingga cocok untuk kondisi opsional di mana tindakan hanya diambil jika syarat terpenuhi, membantu pemahaman dasar pengkondisian tunggal dalam pemrograman.

### **2.2.1 Pembahasan Langkah**

- Buat package "Pekan\_empat" di Eclipse jika belum ada.
- Buat class baru bernama "latifl" dalam package tersebut.
- Import library java.util.Scanner untuk input pengguna.
- Deklarasikan variabel double IPK di dalam method main.
- Inisialisasi objek Scanner dengan nama input.
- Tampilkan prompt "Input IP Anda + " dan baca nilai IPK menggunakan input.nextDouble().
- Tutup objek Scanner dengan input.close().
- Gunakan if (IPK > 2.75) untuk mencetak "Anda Lulus Sangat Memuaskan dengan IPK " + IPK.

- Jalankan program dengan input contoh seperti 3.0 atau 2.5 untuk verifikasi.

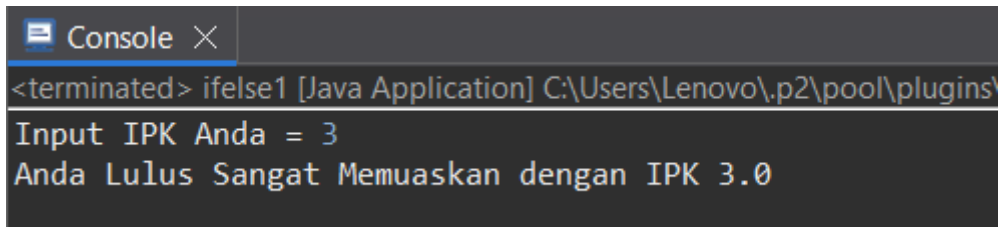
### 2.2.2 Contoh Pemrograman

```

1 package Pekan_empat;
2
3 import java.util.Scanner;
4 public class latif1 {
5
6     public static void main(String[] args) {
7         double IPK;
8         Scanner input=new Scanner(System.in);
9         System.out.print("Input IP 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 }

```

### 2.2.3 Hasil Output



```

<terminated> ifelse1 [Java Application] C:\Users\Lenovo\.p2\pool\plugins\
Input IPK Anda = 3
Anda Lulus Sangat Memuaskan dengan IPK 3.0

```

### 2.2.4 Analisis

Program ini hanya mengeksekusi blok if jika kondisi true, sehingga untuk nilai di bawah 2.75 tidak ada respons, yang berguna untuk aksi opsional. Analisis menemukan potensi improvement dengan penambahan pesan default, dan penanganan input invalid untuk menghindari crash.

### **2.2.5 Teori Mengenai Kode**

Single if adalah bentuk dasar dari struktur kondisional di Java, di mana blok kode hanya dieksekusi jika ekspresi boolean true. Teori ini menekankan efisiensi untuk kondisi tunggal tanpa alternatif, sering digunakan dalam validasi sederhana atau logging.

## **2.3 Multi If**

Program ini menunjukkan penggunaan multiple if statements dalam Java untuk mengevaluasi kombinasi umur dan kepemilikan SIM yang dimasukkan pengguna melalui Scanner, dengan empat kondisi independen yang dicetak berdasarkan operator logika seperti `&&` dan `!=`, sehingga memungkinkan penanganan skenario berbeda tanpa nesting, ideal untuk memahami evaluasi kondisi paralel dalam pemrograman dasar.

### **2.3.1 Pembahasan Langkah**

- Buat package "Pekan\_empat" di Eclipse jika belum ada.
- Buat class baru bernama "multiif" dalam package tersebut.
- Import library `java.util.Scanner` untuk input.
- Deklarasikan variabel `int` umur dan `char` sim di dalam method `main`.
- Inisialisasi objek Scanner dengan nama `a`.
- Tampilkan prompt "Input umur anda: " dan baca nilai umur menggunakan `a.nextInt()`.
- Tampilkan prompt "Apakah Anda Sudah Punya SIM C: " dan baca nilai sim menggunakan `a.next().charAt(0)`.
- Tutup objek Scanner dengan `a.close()`.

- Gunakan if ((umur >= 17) && (sim == 'y')) untuk mencetak "Anda Sudah dewasa dan boleh bawa motor".
- Gunakan if ((umur >= 17) && (sim != 'y')) untuk mencetak "Anda SUDah dewasa tetapi tidak boleh bawa motor".
- Gunakan if ((umur < 17) && (sim != 'y')) untuk mencetak "Anda Belum Cukup Umur bawa motor".
- Gunakan if ((umur < 17) && (sim == 'y')) untuk mencetak "Anda Belum Cukup Umur punya sim".
- Jalankan program dengan input contoh seperti umur 18 dan sim 'y'.

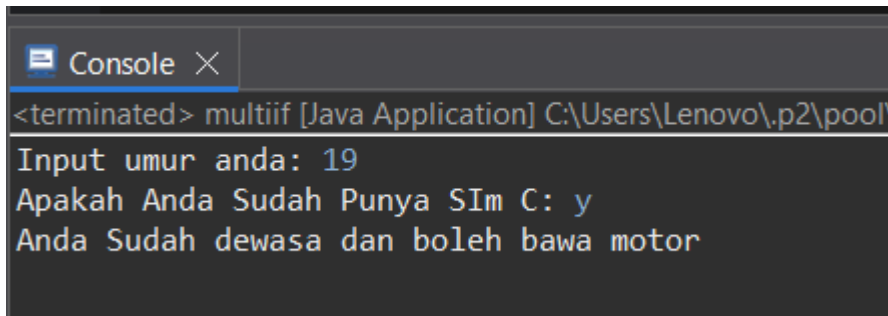
### 2.3.2 Contoh Pemrograman

```

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

```

### 2.3.3 Hasil Output



```
<terminated> multiif [Java Application] C:\Users\Lenovo\p2\pool\
Input umur anda: 19
Apakah Anda Sudah Punya SIm C: y
Anda Sudah dewasa dan boleh bawa motor
```

### 2.3.4 Analisis

Program ini mengevaluasi semua kondisi secara independen, yang efisien untuk kasus mutually exclusive, tetapi bisa dioptimalkan dengan else if untuk menghindari pengecekan berlebih. Analisis menemukan typo seperti "SIm" dan "SUhah", serta potensi error jika input sim bukan char tunggal.

### 2.3.5 Teori Mengenai Kode

Multiple if statements memungkinkan pengecekan kondisi terpisah, menggunakan operator logika seperti && untuk konjungsi. Teori ini berdasarkan pada eksekusi sekuensial, di mana setiap if dievaluasi terlepas dari yang sebelumnya, berguna untuk validasi multi-kondisi.

## 2.4 Nama Bulan

Program ini menggunakan struktur switch case dalam Java untuk mengonversi angka bulan yang dimasukkan pengguna melalui Scanner menjadi nama bulan yang sesuai, dari Januari hingga Desember, dengan default untuk input invalid, sehingga memudahkan penanganan kasus berbasis nilai enumerasi dan meningkatkan keterbacaan kode dibandingkan if-else bertingkat untuk opsi banyak.

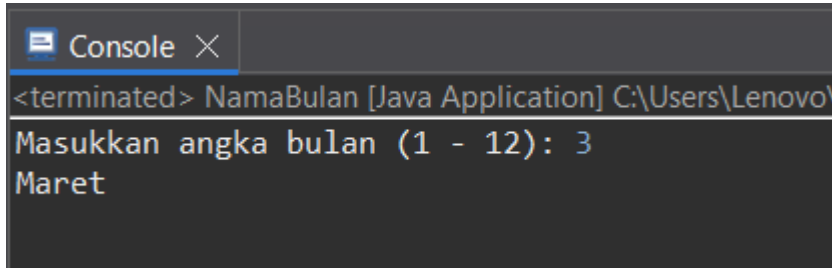
### 2.4.1 Pembahasan Langkah

- Buat package "Pekan\_empat" di Eclipse jika belum ada.
- Buat class baru bernama "NamaBulan" dalam package tersebut.
- Import library `java.util.Scanner` untuk input.
- Inisialisasi objek `Scanner` dengan nama `scanner` di dalam method `main`.
- Tampilkan prompt "Masukkan angka bulan (1 - 12): " dan baca nilai `int` bulan menggunakan `scanner.nextInt()`.
- Gunakan `switch (bulan)` dengan case 1 hingga 12 untuk mencetak nama bulan masing-masing, diikuti `break`.
- Gunakan default untuk mencetak "Angka tidak valid".
- Tutup objek `Scanner` dengan `scanner.close()`.
- Jalankan program dengan input contoh seperti 5 atau 13.

### 2.4.2 Contoh Pemrograman

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

### 2.4.3 Hasil Output



```
<terminated> NamaBulan [Java Application] C:\Users\Lenovo\
Masukkan angka bulan (1 - 12): 3
Maret
```

### 2.4.4 Analisis

Program ini efisien untuk kasus tetap seperti bulan, dengan break mencegah fall-through. Analisis menunjukkan tidak ada error untuk input integer, tapi perlu validasi lebih lanjut untuk non-integer. Switch case lebih readable daripada if-else untuk banyak opsi.

### 2.4.5 Teori Mengenai Kode

Switch case dalam Java digunakan untuk multi-way branching berdasarkan nilai konstan, mendukung int, char, String, dll. Teori ini menekankan penggunaan break untuk keluar dari case, dan default untuk kasus lain, meningkatkan efisiensi dibanding if-else bertingkat.

## 2.5 Penilaian

Program ini menerapkan if-else ladder dalam Java untuk mengonversi nilai angka yang dimasukkan pengguna melalui Scanner menjadi huruf nilai (A hingga E) berdasarkan rentang tertentu, mulai dari  $\geq 81$  untuk A hingga di bawah 50 untuk E, sehingga berguna untuk sistem penilaian bertingkat yang memerlukan evaluasi sekuensial kondisi.



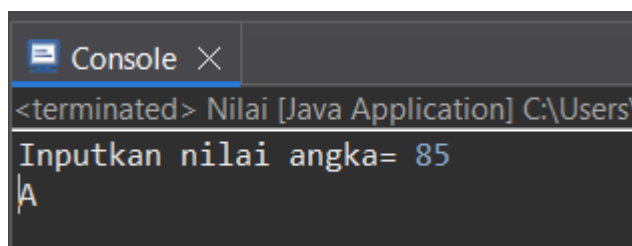
### 2.5.1 Pembahasan Langkah

- Buat package "Pekan\_empat" di Eclipse jika belum ada.
- Buat class baru bernama "Nilai" dalam package tersebut.
- Import library `java.util.Scanner` untuk input.
- Deklarasikan variabel `int nilai` di dalam method `main`.
- Inisialisasi objek `Scanner` dengan nama `input`.
- Tampilkan prompt "Inputkan nilai angka= " dan baca nilai menggunakan `input.nextInt()`.
- Tutup objek `Scanner` dengan `input.close()`.
- Gunakan `if (nilai >= 81)` untuk mencetak "A".
- Gunakan `else if (nilai >= 70)` untuk mencetak "B".
- Gunakan `else if (nilai >= 60)` untuk mencetak "c".
- Gunakan `else if (nilai >= 50)` untuk mencetak "D".
- Gunakan `else` untuk mencetak "E".
- Jalankan program dengan input contoh seperti 85 atau 45.

### 2.5.2 Contoh Pemrograman

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

### 2.5.3 Hasil Output



```
<terminated> Nilai [Java Application] C:\Users\
Inputkan nilai angka= 85
A
```

#### **2.5.4 Analisis**

**Program ini mengevaluasi kondisi secara berurutan hingga menemukan yang true, efisien untuk rentang nilai. Analisis menemukan typo pada "c" yang seharusnya "C" untuk konsistensi, dan perlu penanganan nilai negatif atau di atas 100.**

#### **2.5.5 Teori Mengenai Kode**

**If-else ladder adalah rantai else if untuk menangani kondisi bertingkat, di mana hanya satu blok dieksekusi. Teori ini berdasarkan evaluasi top-down, berguna untuk grading atau kategorisasi, dengan else sebagai fallback.**

## **BAB III**

### **KESIMPULAN**

Dari praktikum ini, dapat disimpulkan bahwa struktur kontrol seperti if-else, single if, multi if, switch case, dan if-else ladder merupakan komponen esensial dalam pemrograman Java untuk mengelola alur program berdasarkan kondisi. Implementasi di Eclipse memungkinkan kami memahami secara praktis bagaimana input pengguna mempengaruhi output, dari pengkondisian sederhana hingga multi-kasus. Hal ini memperkuat dasar logika pemrograman dan menekankan pentingnya penanganan error input.

Secara keseluruhan, praktikum pertemuan keempat berhasil mencapai tujuannya dalam memperkenalkan variasi struktur kontrol. Analisis output konsisten dengan teori, meskipun ada potensi perbaikan seperti validasi input dan koreksi typo. Penggunaan Scanner dan Eclipse memfasilitasi interaksi, sementara konsep ini menjadi pondasi untuk aplikasi lebih kompleks.

Akhirnya, pemahaman ini membuka jalan untuk topik lanjutan seperti loop dan method di Java. Praktikum ini meningkatkan keterampilan analitis dan pemecahan masalah, dengan rekomendasi untuk eksplorasi kasus real-world guna mendalami aplikasi struktur kontrol tersebut.

## DAFTAR PUSTAKA

1. Schildt, Herbert. (2019). *Java: The Complete Reference*. McGraw-Hill Education.
2. Oracle. (2023). *The Java™ Tutorials: Control Flow Statements*.
3. Deitel, Paul & Deitel, Harvey. (2020). *Java How to Program*. Pearson Education.
4. Horstmann, Cay S. (2018). *Core Java Volume I—Fundamentals*. Addison-Wesley.