

**LAPORAN PRATIUM**  
**PEMROGRAMAN ALGORITMA PEMROGRAMAN**  
**CONDITIONAL STATEMENT**

**Disusun Oleh:**

Gina Ramadhani

Nim: 2511533014

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

Asisten Pratikum: Rahmad Dwi Rizki Olders



**DAPERTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**TAHUN 2025**

## **KATA PENGANTAR**

Laporan praktikum ini dibuat sebagai bentuk pertanggungjawaban atas kegiatan praktikum mata kuliah Algoritma Pemrograman, khususnya mengenai Conditional Statement dalam Bahasa Pemrograman Java. Melalui penyusunan laporan ini, penulis dapat memperdalam pemahaman terhadap materi praktikum sekaligus melatih ketelitian, keteraturan, serta kemampuan menulis sesuai dengan kaidah akademik dasar. Dengan demikian, laporan ini diharapkan tidak hanya menjadi dokumentasi kegiatan praktikum, tetapi juga dapat berfungsi sebagai sarana pembelajaran dan latihan keterampilan menulis ilmiah yang bermanfaat untuk proses belajar berikutnya.

Penulis menyadari bahwa laporan ini masih memiliki banyak kekurangan. Oleh sebab itu, penulis sangat mengharapkan adanya kritik dan saran yang bersifat membangun agar kualitas laporan dan pemahaman penulis dapat lebih baik pada kesempatan selanjutnya.

Padang, 2025

Penulis

## **DAFTAR ISI**

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN .....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
<b>BAB II .....</b>	<b>3</b>
<b>PEMBAHASAN .....</b>	<b>3</b>
2.1 Praktikum “Ifelse1” .....	3
2.2 Praktikum “multiIf” .....	5
2.3 Praktikum “latIf1” .....	7
2.4 Praktikum “Nilai” .....	9
2.5 Praktikum ‘NamaBulan’ .....	11
<b>BAB III.....</b>	<b>15</b>
<b>KESIMPULAN .....</b>	<b>15</b>
3.1 Kesimpulan .....	15
<b>DAFTAR PUSTAKA .....</b>	<b>16</b>

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam pemrograman, pengambilan keputusan merupakan salah satu aspek penting yang menentukan bagaimana alur logika sebuah program berjalan. Conditional statement atau pernyataan bersyarat digunakan untuk mengatur jalannya program berdasarkan kondisi tertentu yang bernilai benar (*true*) atau salah (*false*). Dengan adanya conditional statement, program dapat merespons berbagai situasi secara dinamis, sehingga lebih fleksibel dan sesuai dengan kebutuhan pengguna.

Conditional statement sangat penting dipelajari karena hampir semua aplikasi maupun sistem membutuhkan logika pengambilan keputusan, misalnya untuk menentukan hasil ujian, mengecek status login, memvalidasi data, hingga mengendalikan proses pada perangkat lunak yang kompleks. Tanpa penerapan conditional statement, program hanya akan berjalan secara linier tanpa kemampuan beradaptasi terhadap kondisi yang berbeda.

Melalui praktikum ini, mahasiswa diharapkan mampu memahami konsep dasar conditional statement serta dapat mengimplementasikannya dalam bahasa pemrograman. Pemahaman ini akan menjadi landasan untuk mempelajari struktur kontrol lain yang lebih kompleks, sehingga mendukung kemampuan analisis dan penyusunan algoritma secara logis dalam pemrograman.

#### **1.2 Tujuan Praktikum**

1. Memahami konsep dasar conditional statement dalam pemrograman.
2. Mampu mengimplementasikan conditional statement menggunakan Bahasa pemrograman

3. Mengetahui perbedaan antara penggunaan if, if-else, dan switch case.
4. Melatih kemampuan dalam menyusun logika pemrograman yang bersifat dinamis.

### **1.3 Manfaat Praktikum**

1. Memberikan pemahaman tentang peran conditional statement dalam membangun program yang interaktif.
2. Meningkatkan keterampilan analisis logika dalam pemecahan masalah melalui pemrograman.
3. Menjadi dasar untuk mempelajari struktur kontrol lain seperti perulangan dan fungsi.
4. Membekali mahasiswa dengan kemampuan praktis dalam merancang algoritma yang sesuai dengan kondisi yang bervariasi.

## BAB II

### PEMBAHASAN

#### 2.1 Praktikum “Ifelse1”

```

1 package pekan4;
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 }
18 }
19 }
20 }
21 }

```

**Gambar 2.1** Kode Program Praktikum Ifelse1

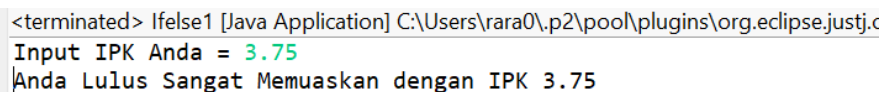
Program ini menggunakan struktur kontrol percabangan if-else untuk menentukan apakah seseorang lulus dengan sangat memuaskan atau tidak lulus, berdasarkan nilai IPK (Indeks Prestasi Kumulatif) yang diinput oleh pengguna.

Berikut penjelasan langkah-langkah kerjanya:

1. package pekan4;  
→ Menunjukkan bahwa program ini berada di dalam paket bernama *pekan4*.
2. import java.util.Scanner;  
→ Mengimpor kelas Scanner dari pustaka Java untuk membaca input dari pengguna melalui keyboard.
3. public class Ifelse1 { ... }  
→ Mendefinisikan kelas utama bernama Ifelse1.
4. public static void main(String[] args) { ... }  
→ Merupakan metode utama tempat eksekusi program dimulai.

5. `double IPK;`  
→ Mendeklarasikan variabel bertipe double untuk menyimpan nilai IPK.
6. `Scanner input = new Scanner(System.in);`  
→ Membuat objek Scanner bernama input untuk mengambil input dari pengguna.
7. `System.out.print("Input IPK Anda = ");`  
→ Menampilkan teks agar pengguna memasukkan IPK.
8. `IPK = input.nextDouble();`  
→ Membaca nilai IPK yang dimasukkan oleh pengguna.
9. `input.close();`  
→ Menutup objek Scanner setelah digunakan.
10. `if (IPK > 2.75) { ... } else { ... }`  
→ Struktur percabangan yang memeriksa apakah IPK lebih besar dari 2.75.
11. `System.out.println("Anda Lulus Sangat Memuaskan dengan IPK " + IPK);`  
→ Jika kondisi benar ( $IPK > 2.75$ ), maka program menampilkan pesan bahwa pengguna lulus dengan sangat memuaskan.
12. `System.out.println("Anda Tidak Lulus");`  
→ Jika kondisi salah ( $IPK \leq 2.75$ ), maka program menampilkan pesan bahwa pengguna tidak lulus.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.2 dibawah ini.



```
<terminated> Ifelse1 [Java Application] C:\Users\rara0\p2\pool\plugins\org.eclipse.justj.c
Input IPK Anda = 3.75
Anda Lulus Sangat Memuaskan dengan IPK 3.75
```

**Gambar 2.2** Output dari kode program Ifelse1

### 2.1.1 Analisis

Pada percobaan ini, program Ifelse1 dijalankan untuk menguji penerapan struktur percabangan if-else dalam menentukan hasil kelulusan berdasarkan nilai IPK yang diinput oleh pengguna.

Dari hasil tersebut dapat disimpulkan bahwa program berjalan dengan benar dan sesuai dengan logika percabangan yang digunakan. Program mampu mengambil keputusan secara otomatis berdasarkan nilai IPK yang dimasukkan pengguna.

## 2.2 Praktikum “multiIf”

```

1 package pekan4;
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 }
29
30

```

**Gambar 2.3** Kode Program Praktikum multiIf

Program ini termasuk ke dalam materi struktur kendali percabangan (conditional statement), khususnya menggunakan beberapa pernyataan if terpisah (*multiple if*).

Tujuannya adalah untuk memeriksa dua kondisi logis, yaitu:

1. Umur pengguna (apakah sudah 17 tahun ke atas atau belum).
2. Kepemilikan SIM C (dijawab dengan huruf 'y' untuk *ya* atau 't' untuk *tidak*).

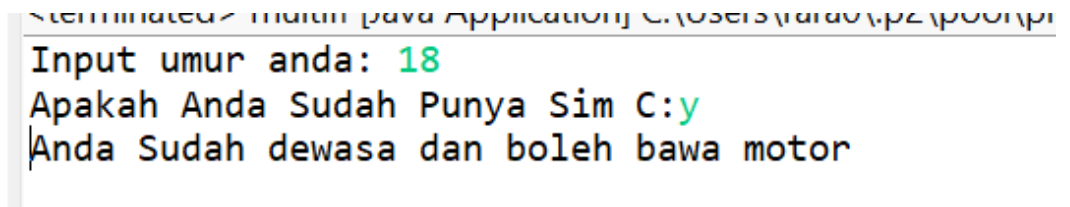


Berdasarkan kedua kondisi tersebut, program memberikan pesan berbeda tergantung hasil pemeriksaan.

Berikut penjelasan langkah-langkah kerjanya:

1. `package pekan4;`  
→ Menandakan bahwa file ini termasuk dalam package bernama *pekan4*.
2. `import java.util.Scanner;`  
→ Mengimpor kelas Scanner untuk membaca input dari pengguna.
3. `public class multiIf { ... }`  
→ Mendefinisikan kelas utama program bernama `multiIf`.
4. `int umur; char sim;`  
→ Mendeklarasikan dua variabel:
  - umur untuk menyimpan umur pengguna (tipe `int`).
  - sim untuk menyimpan jawaban apakah memiliki SIM C (tipe `char`).
5. `Scanner a = new Scanner(System.in);`  
→ Membuat objek Scanner untuk menerima input dari keyboard.
6. `System.out.print("Input umur anda: ");`  
→ Meminta pengguna untuk memasukkan umur.
7. `umur = a.nextInt();`  
→ Menyimpan input umur ke variabel umur.
8. `System.out.print("Apakah Anda Sudah Punya Sim C:");`  
→ Menanyakan apakah pengguna sudah memiliki SIM C.
9. `sim = a.next().charAt(0);`  
→ Membaca satu karakter dari input (misalnya 'y' untuk *ya* atau 't' untuk *tidak*).
10. `a.close();`  
→ Menutup objek Scanner.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.4 dibawah ini.



```

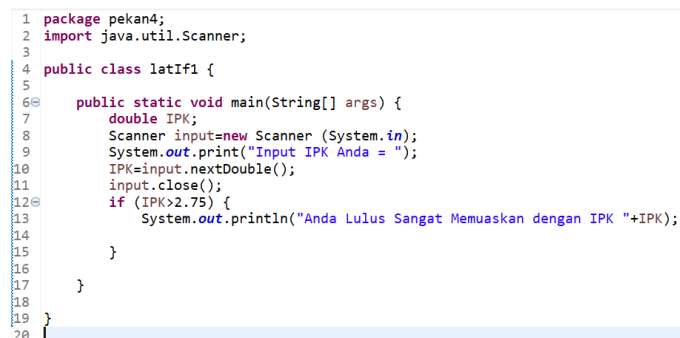
terminated? main [Java Application] C:\Users\linda\AppData\Local\Programs\Java\jdk-11.0.2\bin\java.exe
Input umur anda: 18
Apakah Anda Sudah Punya Sim C:y
Anda Sudah dewasa dan boleh bawa motor
  
```

**Gambar 2.4** Output dari kode program multiIf

### 2.1.2 Analisis

- Program dapat menganalisis dua kondisi sekaligus dengan operator logika && (AND).
- Karena pengguna sudah cukup umur ( $\geq 17$ ) dan memiliki SIM, maka program menampilkan pesan kelulusan logis — boleh membawa motor.
- Tidak ada pesan ganda atau konflik, menandakan logika if pada bagian ini berjalan sesuai harapan
- Program berhasil memproses input dengan benar dan memberikan output sesuai logika percabangan.

## 2.3 Praktikum “latIf1”



```

1 package pekan4;
2 import java.util.Scanner;
3
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 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
19 }
20
  
```

**Gambar 2.5** Kode Program Praktikum latIf1

Berikut penjelasan Langkah-langkah kerjanya:

1. Baris 1–2:

Program mengimpor package `java.util.Scanner` yang digunakan untuk membaca input dari pengguna melalui keyboard.

2. Baris 4:

Dideklarasikan kelas `latIf1` sebagai kelas utama program.

3. Baris 6–18:

Di dalam metode `main`, program melakukan langkah-langkah berikut:

- Mendeklarasikan variabel `IPK` bertipe `double` untuk menyimpan nilai `IPK`.
  - Membuat objek `Scanner` bernama `input` untuk membaca data dari pengguna.
  - Menampilkan pesan ke layar:
4. Program ini berfungsi untuk mengevaluasi hasil studi mahasiswa berdasarkan nilai `IPK` yang dimasukkan. Jika `IPK` lebih dari 2.75, maka mahasiswa dianggap lulus sangat memuaskan.
5. Program belum memiliki blok `else`, sehingga jika `IPK`  $\leq 2.75$  tidak akan menampilkan pesan apa pun.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.6 dibawah ini.

```
Input IPK Anda = 3
Anda Lulus Sangat Memuaskan dengan IPK 3.0
```

**Gambar 2.6** Output dari kode program `latIf1`

### 2.1.3 Analisis

- Program berjalan dengan benar dan menampilkan output sesuai logika percabangan `if`.
- Tidak ada error dalam eksekusi.
- Input berupa angka desimal (3) diterima dengan baik oleh program karena menggunakan tipe data

double.

## 2.4 Praktikum “Nilai”

```

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.print("Inputkan nilai angka=");
11        nilai=input.nextInt();
12        input.close();
13
14        if(nilai >=81) {
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 }
29

```

**Gambar 2.7** Kode Program Praktikum Nilai

Program meminta pengguna untuk memasukkan nilai angka (tipe data int), kemudian menggunakan struktur percabangan if-else if-else untuk menentukan hasil berupa huruf mutu (A–E) sesuai rentang nilai tertentu.

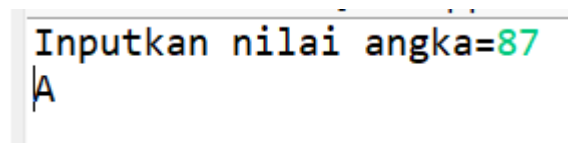
Berikut penjelasan langkah-langkah kerjanya:

- Baris 1: Menentukan bahwa program berada dalam paket bernama *pekan4*.
- Baris 2: Mengimpor library Scanner untuk membaca input dari pengguna.
- Baris 4: Mendefinisikan kelas utama dengan nama *Nilai*.
- Baris 6: Mendeklarasikan metode utama main yang menjadi titik awal eksekusi program.
- Baris 7: Membuat variabel bertipe integer untuk menyimpan nilai yang dimasukkan pengguna.

- Baris 8: Membuat objek Scanner untuk mengambil input dari keyboard.
- Baris 9: Menampilkan perintah kepada pengguna agar memasukkan nilai angka.
- Baris 10: Menerima input nilai dari pengguna dan menyimpannya ke variabel.
- Baris 11: Menutup objek Scanner setelah digunakan.
- Baris 13–15: Mengecek apakah nilai lebih besar atau sama dengan 81, jika ya maka hasilnya adalah “A”.
- Baris 16–17: Mengecek apakah nilai lebih besar atau sama dengan 70, maka hasilnya adalah “B”.
- Baris 18–19: Mengecek apakah nilai lebih besar atau sama dengan 60, maka hasilnya adalah “C”.
- Baris 20–21: Mengecek apakah nilai lebih besar atau sama dengan 50, maka hasilnya adalah “D”.
- Baris 22–24: Jika semua kondisi di atas tidak terpenuhi, maka hasilnya adalah “E”.
- Baris 27: Menutup blok metode utama.
- Baris 28: Menutup blok kelas program.

Secara keseluruhan, program ini digunakan untuk mengonversi nilai angka menjadi nilai huruf (A–E) dengan menggunakan struktur percabangan if-else-if.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.8 dibawah ini.



**Gambar 2.8** Output dari kode program Nilai

#### 2.1.4 Analisis

- Program menggunakan if-else-if secara berurutan, sehingga setiap kondisi akan diperiksa dari atas ke bawah.
- Hanya satu kondisi yang benar yang akan dijalankan; kondisi lain diabaikan.
- Dengan demikian, tidak terjadi tumpang tindih hasil.
- Program berjalan dengan benar dan efisien untuk menentukan nilai huruf berdasarkan skor angka. Struktur logikanya sudah tepat, mudah dibaca, dan sesuai dengan prinsip dasar penggunaan kondisi bertingkat dalam Java.

## 2.5 Praktikum ‘NamaBulan’

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class NamaBulan {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner (System.in);
8         System.out.print("Masukkan 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
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 }
54 }

```

**Gambar 2.9** Kode program praktikum NamaBulan

Program ini adalah contoh penggunaan switch-case dalam Java untuk mengonversi angka menjadi nama bulan dalam bahasa Indonesia. Program ini menangani input yang valid (1–12) dan memberikan pesan error untuk input yang tidak valid.

1. Package dan Import

- Program berada dalam package bernama pekan4.
- Mengimpor kelas Scanner dari java.util untuk membaca input dari keyboard.

2. Kelas dan Metode Utama

- Nama kelas: NamaBulan.
- Metode main adalah titik awal eksekusi program.

3. Input Pengguna

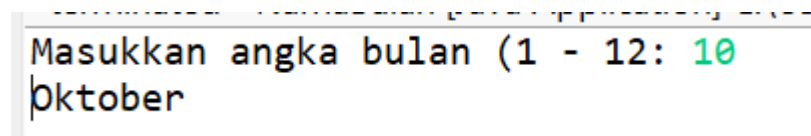
- Program mencetak pesan: "Masukkan angka bulan (1 - 12): ".
- Menggunakan Scanner untuk membaca input angka dari pengguna dan menyimpannya dalam variabel bulan.

4. Penggunaan Struktur switch-case

- Program mengecek nilai dari variabel bulan.
- Jika nilainya antara 1 hingga 12, program mencetak nama bulan yang sesuai dalam Bahasa Indonesia:
  - → Januari
  - → Februari
  - → Maret
  - → April
  - → Mei
  - → Juni
  - → Juli
  - → Agustus
  - → September
  - → Oktober
  - → November
  - 12 → Desember

- Jika angka di luar rentang 1–12, program mencetak: "Angka tidak valid".
5. Penutupan Scanner
- Setelah input selesai digunakan, `scanner.close()` dipanggil untuk menutup input stream dan membebaskan resource.

Dari Langkah-langkah diatas kita akan mendapatkan output seperti gambar 2.10 dibawah ini.



**Gambar 2.10** Output dari kode program NamaBulan

### 2.1.5 Analisis

- Variabel **bulan** bertipe `int` dan dideklarasikan sebagai variabel lokal di dalam method `main()`, sehingga hanya dapat diakses dalam lingkup method tersebut.
- Objek **Scanner** digunakan untuk membaca input dari pengguna melalui konsol (`System.in`), dan harus ditutup setelah digunakan untuk menghindari memory leak, seperti yang dilakukan dengan `scanner.close()`.
- Struktur kontrol **switch-case** digunakan untuk memetakan nilai numerik (1–12) ke nama-nama bulan dalam Bahasa Indonesia, yang membuat kode lebih terstruktur dan mudah dibaca.



- case digunakan untuk menangani input yang tidak valid (di luar rentang 1–12), yang merupakan praktik baik untuk menangani kemungkinan error dari input pengguna.
- Setiap case diakhiri dengan break; untuk mencegah eksekusi berlanjut ke case berikutnya (fall-through), sesuai dengan standar penggunaan switch di Java.
- Penamaan variabel mengikuti konvensi camelCase (scanner, bulan) yang merupakan standar penamaan di Java.

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Berdasarkan hasil praktikum yang telah dilakukan pada materi conditional statement, dapat disimpulkan bahwa pemahaman terhadap struktur kendali kondisi merupakan hal yang sangat penting dalam dunia pemrograman. Conditional statement digunakan untuk mengatur alur logika program agar dapat merespon berbagai kondisi atau situasi yang mungkin terjadi selama program dijalankan. Beberapa bentuk pernyataan kondisi yang umum digunakan adalah if, if else, dan switch case. Masing-masing memiliki fungsi dan kegunaan tersendiri, tergantung pada kompleksitas logika yang dibutuhkan dalam program.

Dalam praktikum ini, mahasiswa mempelajari bagaimana cara menggunakan conditional statement untuk mengarahkan jalannya program berdasarkan input yang diberikan oleh pengguna. Salah satu implementasi yang digunakan adalah pemilihan nama bulan berdasarkan angka, yang diselesaikan dengan menggunakan struktur switch case. Struktur ini sangat efektif ketika kondisi yang diperiksa merupakan nilai tetap dalam jumlah banyak. Selain itu, penggunaan break pada setiap case diperlukan untuk mencegah terjadinya kesalahan logika akibat eksekusi yang tidak diinginkan.

Praktikum ini juga menekankan pentingnya melakukan validasi terhadap input pengguna. Hal ini bertujuan untuk memastikan bahwa input yang diterima sesuai dengan harapan program, serta untuk menghindari error atau keluaran yang tidak sesuai. Tanpa validasi yang tepat, program dapat menghasilkan output yang salah atau bahkan gagal dijalankan.

Secara keseluruhan, praktikum ini memberikan pemahaman yang mendalam tentang cara kerja dan penerapan conditional statement dalam membangun logika program yang dinamis, efisien, dan sesuai dengan kebutuhan. Dengan menguasai materi ini, mahasiswa diharapkan mampu membuat program yang tidak hanya berfungsi dengan baik, tetapi juga mudah dipahami, dirawat, dan dikembangkan di masa depan.

## DAFTAR PUSTAKA

- [1] Horstmann, C. S. (2021). *Core Java volume I – fundamentals* (12th ed.). Prentice Hall.
- [2] Oracle. (2024). *The Java™ tutorials – control flow statements*. Retrieved October 5, 2025, from <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html>. [Diakses: 4-Okt-2025]
- [3] Schildt, H. (2022). *Java: The complete reference* (12th ed.). McGraw-Hill Education.