

LAPORAN PRAKTIKUM
ALGORITMA PEMOGRAMAN
STATEMENT KONDISIONAL PADA PEMOGRAMAN JAVA

Disusun Oleh :

Aliifah Felda Mufarrihati Salwaa

2511531011

Dosen Pengampu :

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

Asisten Praktikum :

Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Laporan praktikum ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma pemograman yang membahas tentang statement kondisional pada pemograman Java. Melalui laporan ini penulis dapat memahami materi praktikum secara mendalam dan dapat lebih teliti, teratur, serta memiliki kemampuan yang baik dalam penulisan kode pemograman sesuai kaidah akademik. Sehingga laporan ini dapat menjadi sarana belajar, dokumentasi kegiatan, dan referensi praktikum berikutnya.

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

Padang, 29 September 2025

Aliifah Felda Mufarrihati Salwaa

2511531011

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Praktikum	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	2
BAB II PEMBAHASAN	3
2.1 Statement kondisional pada pemograman Java.....	3
2.2 Praktikum lat If 1.....	4
2.3 Praktikum If Else 1	5
2.4 Praktikum Multi If.....	7
2.5 Praktikum Nilai	9
2.6 Praktikum Nama Bulan	11
BAB III KESIMPULAN	14
3.1 Kesimpulan.....	14
3.2 Saran.....	14
DAFTAR PUSTAKA	15

BAB I

PENDAHULUAN

1.1 Latar Belakang Praktikum

Kemampuan suatu program untuk mengambil keputusan berdasarkan suatu kondisi tertentu tentunya menjadi pondasi utama untuk membuat program tersebut menjadi sebuah program yang responsif. Tanpa adanya pengambilan keputusan oleh program, program tersebut hanya akan menampilkan hasil secara linear tanpa bisa membedakan input atau keadaan yang berbeda. Karena hal itu, statement kondisional sangat dibutuhkan ketika kita hendak membuat sebuah program yang baik dan dinamis.

Dalam pemrograman Java ada beberapa pernyataan kondisional, seperti *if*, *if-else*, *if-else-if*, dan *switch*. Pernyataan kondisional ini yang kemudian akan membuat sebuah program menjadi responsif terhadap suatu kondisi. Penguasaan pada statement kondisional ini akan membawa kita pada pemahaman yang lebih lanjut dalam memproses kode pemrograman dalam bahasa Java.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Mengetahui bagaimana program Java dapat bekerja menggunakan statement kondisional.
2. Mengetahui langkah-langkah dalam membuat program Java dengan statement kondisional.
3. Mengetahui hasil dari program Java yang menggunakan statement kondisional.

1.3 Manfaat Praktikum

Manfaat dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Memahami penulisan kode pemrograman Java terutama dalam menggunakan statement kondisional.
2. Mampu mengimplementasikan statement kondisional pada pembuatan kode program Java berikutnya.
3. Mampu mengidentifikasi perbedaan dari setiap statement kondisional dan bagaimana cara mengaplikasikannya pada kode program Java.

BAB II PEMBAHASAN

2.1. Statement kondisional pada pemograman Java

Kemampuan sebuah komputer untuk melakukan sebuah pekerjaan dibuat dengan hanya beberapa menggabungkan komando sederhana menjadi struktur kontrol. Pada pemograman Java, ada enam struktur yang digunakan untuk menentukan alur kontrol normal dalam suatu program. keenam struktur kontrol tersebut adalah: *block*, *while loop*, *do.while loop*, *for loop*, *if statement*, dan *switch statement*. Masing-masing struktur ini dianggap sebagai satu pernyataan yang dapat berisi satu atau lebih pernyataan lain di dalamnya.

Statement kondisional yang kita gunakan pada praktikum kali ini adalah statement *if* dan statement *switch*. Statemnt *if* adalah pernyataan paling dasar dari semua pernyataan alur kontrol. Pernyataan ini memberitahu program untuk mengeksekusi kode tertentu, jika pengujian tertentu bernilai benar. Selain statement *if*, kita juga memiliki statement *if-else*. Statement *if-else* memeberikan opsi kedua jika *if* bernilai salah.

Berbeda dengan statement *if*, statement *switch* bisa memiliki beberapa jalur eksekusi. Badan dari pernyataan *switch* dikenal sebagai *block switch*. Menentukan apakah akan menggunakan statement *if* atau statement *switch* didasarkan pada apa yang akan diuji oleh statement tersebut. Statement *if-else* dapat digunakan ketika kita memiliki rentang nilai, sementara statement *switch* bisa digunakan ketika menguji hanya berdasarkan satu nilai.

2.2. Praktikum “latIf1”

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class latIf1 {
6
7     public static void main(String[] args) {
8         double IPK;
9         Scanner input = new Scanner(System.in);
10        System.out.println("Input IPK anda : ");
11        IPK = input.nextDouble();
12        input.close();
13        if (IPK>2.75) {
14            System.out.println("Anda lulus Sangat Memuaskan dengan IPK "+IPK);
15        }
16    }
17 }
18
19 }

```

Gambar 2.1: Kode program praktikum “latIf1”

Pada kode program ini, kita melakukan suatu proses yang termasuk singkat, dimana kita memasukkan input IPK dan program menganalisis sesuai kondisi yang kita masukkan pada statement *if* yang kita gunakan. Berikut untuk penjelasan lengkap tentang langkah-langkah program diatas:

1. Pada baris kode 3 kita melakukan pemanggilan *scanner* dengan *java.util*
2. Pada baris kode 8 kita mendeklarasikan variabel IPK dengan tipe data *double*.
3. Pada baris kode 9-12 kita meminta *input* IPK dari user berupa jenis data *double*.
4. Pada baris kode 13-14 kita melakukan perbandingan menggunakan statement *if*, dimana jika $IPK > 2.75$ maka kita akan mengeluarkan *output* berupa “Anda lulus sangat memuaskan dengan IPK:” dimana *output* IPK adalah sesuai dengan *input* IPK yang dimasukkan oleh user.
5. Jika IPK yang dimasukkan oleh user < 2.75 , maka program tidak akan menjalankan apapun dan akan stuck pada porses *input*.

Dari seluruh langkah-langkah yang telah kita gunakan untuk menuliskan program, maka kita akan mendapatkan *output* seperti gambar 2.2 dan gambar 2.3

```
<terminated> latIf1 [Java Application] D:\Eclipse\eclipse\plugins
Input IPK anda :
2.76
Anda lulus Sangat Memuaskan dengan IPK 2.76
```

Gambar 2.2: output kode porgram “latIf1”

Output ini akan kita dapatkan apabila IPK yang kita *input*-kan lebih besar dari 2.75. Namun, apabila nilai IPK yang kita *input* lebih kecil dari 2.75, maka kita tidak akan mendapatkan *output* apapun dari program seperti pada gambar berikut:

```
<terminated> latIf1 [Java Application] D:\Eclipse\eclipse\plugins
Input IPK anda :
2
```

Gambar 2.3: output 2 kode porgram “latIf1”

Hal ini terjadi karena program tidak memiliki perintah apapun selain untuk mengeluarkan *output* ketika nilai IPK lebih besar daripada 2.75

2.3. Praktikum “Ifelse1”

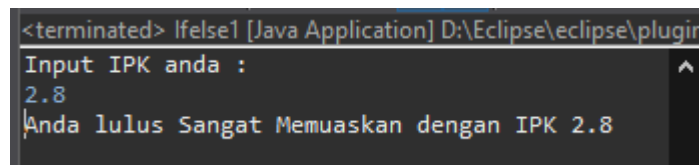
```
1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class Ifelse1 {
6
7     public static void main(String[] args) {
8         double IPK;
9         Scanner input = new Scanner(System.in);
10        System.out.println("Input IPK anda : ");
11        IPK = input.nextDouble();
12        input.close();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

Gambar 2.4: kode program praktikum “Ifelse1”

Berbeda dengan praktikum sebelumnya, pada praktikum kali ini kita menggunakan statement *if-else* yang dimana program akan tetap mengeluarkan *output* apabila *input* oleh user berbeda dari rentang nilai yang kita miliki pada *if*. Berikut untuk langkah-langkah lengkap dari kode program yang kita punya:

1. Pada baris kode 3 kita memanggil *scanner* dari *java.util* karena kita membutuhkan *input* IPK dari user.
2. Pada baris kode 8 kita mendeklarasikan variabel IPK sebagai jenis data *double*.
3. Pada baris kode 9-12 kita meminta *input* dari user dan memasukkannya kedalam variabel IPK
4. Kemudian kita akan membandingkan nilai IPK dengan 2.75
5. Apabila IPK lebih besar daripada 2.75, maka program akan mengeluarkan *output* berupa “Anda lulus sangat memuaskan dengan IPK:” dengan IPK adalah *input* dari user.
6. Apabila IPK lebih kecil dari 2.75, maka program akan mengeluarkan *output* berupa “Anda tidak lulus”.

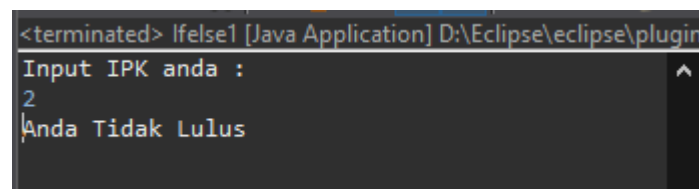
Dari seluruh langkah yang kita tuliskan pada kode program, kita akan menghasilkan dua buah *input* yang berbeda, yakni:



```
<terminated> Ifelse1 [Java Application] D:\Eclipse\eclipse\plugin
Input IPK anda :
2.8
Anda lulus Sangat Memuaskan dengan IPK 2.8
```

Gambar 2.5: output kode program “Ifelse1”

Output ini kita dapatkan apabila kita memasukkan *input* IPK lebih besar daripada 2.75. Apabila *input* IPK kurang daripada 2.75, maka kita akan mendapatkan *output* seperti berikut:



```
<terminated> Ifelse1 [Java Application] D:\Eclipse\eclipse\plugin
Input IPK anda :
2
Anda Tidak Lulus
```

Gambar 2.6: output 2 kode program “Ifelse1”

Dari kedua *input* ini, kita bisa menyimpulkan bahwa statement *if* dan statement *if-else* memiliki fungsi yang jelas berbeda. Statement *if* kita gunakan ketika tidak ingin ada *output* selain dari *range if* yang kita punya, sementara statement *if-else* kita gunakan ketika ingin membuat *output* untuk *input* selain dari *range if*.

2.4. Praktikum “multiIf”

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class multiIf {
6
7     public static void main(String[] args) {
8         int umur;
9         char sim;
10        Scanner a = new Scanner(System.in);
11        System.out.println("Input umur anda : ");
12        umur = a.nextInt();
13        System.out.println("Apakah anda sudah punya sim C : ");
14        sim = a.next().charAt(0);
15        a.close();
16        if ((umur>=17)&&(sim=='y')) {
17            System.out.println("Anda sudah dewasa dan boleh bawa motor");
18        }
19        if ((umur>=17)&&(sim!='y')) {
20            System.out.println("Anda sudah dewasa tetapi tidak boleh bawa motor");
21        }
22        if ((umur<17)&&(sim!='y')) {
23            System.out.println("Anda belum cukup umur bawa motor");
24        }
25        if ((umur<17)&&(sim=='y')) {
26            System.out.println("Anda belum cukup umur punya SIM");
27        }
28    }
29 }
30
31 }
32

```

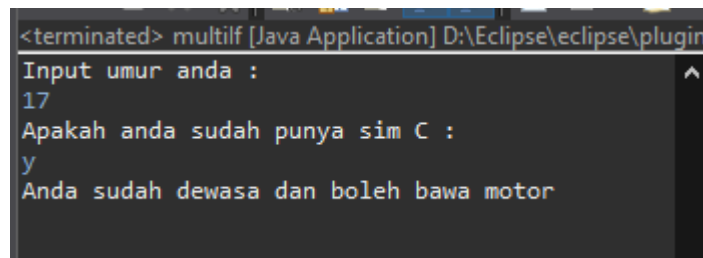
Gambar 2.7: kode program praktikum “multiIf”

Pada kode program ini, kita menggunakan statement *multi If* yang berbeda dari statement *if* dan *if-else* sebelumnya. Pada *multi if*, kita dapat menentukan banyak *output* sesuai dengan berapa banyak *range* yang ingin kita gunakan asalkan *range if* yang satu tidak mengganggu *range if* yang lain. Berikut untuk langkah-langkah lengkap kode program yang kita buat:

1. Pada baris kode 3 kita memanggil *scanner* dari *java.util* karena kita membutuhkan *input* dari user.

2. Pada baris kode 8 kita mendeklarasikan variabel umur sebagai jenis data *integer* dan variabel sim sebagai data *char*.
3. Pada baris kode 10-15 meminta *input* umur dan sim dari user.
4. Pada baris kode 16-25, kita melakukan perbandingan sebagai berikut:
 - a. Jika umur ≥ 17 && sim==y, maka program akan menghasilkan *output* “Anda sudah dewasa dan boleh bawa motor”
 - b. Jika umur ≥ 17 && sim!=y, maka program akan menghasilkan *output* “Anda suda dewasa tetapi tidak boleh bawa motor”
 - c. Jika umur < 17 && sim!=y, maka program akan menghasilkan *output* “Anda belum cukup umur bawa motor”
 - d. Jika umur < 17 && sim==y, maka program akan menghasilkan *output* “Anda belum cukup umur punya sim”

Dari seluruh langkah kerja kode program yang telah dijabarkan, kita pasti akan mendapatkan 4 *output* yang berbeda sesuai dengan *input* apa yang kita masukkan. Berikut salah satu dari *output* yang akan kita hasilkan dari kode program diatas.



```

<terminated> multilf [Java Application] D:\Eclipse\eclipse\plugin
Input umur anda :
17
Apakah anda sudah punya sim C :
y
Anda sudah dewasa dan boleh bawa motor
  
```

Gambar 2.8: salah satu *output* kode program “multiIf”

Output yang dihasilkan akan terus berbeda sesuai dengan umur dan sim yang kita masukkan.

2.5. 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.println("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

```

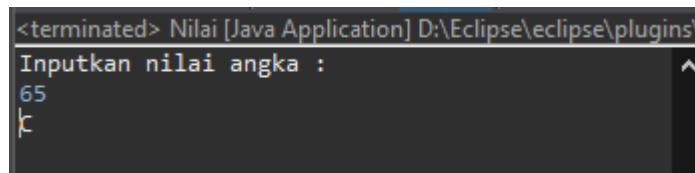
Gambar 2.9: kode program praktikum “Nilai”

Pada kode program ini, kita menggunakan yang namanya statement *nested if-else* yang berarti akan ada *if-else* lain di dalam sebuah *if-else*. *Nested if-else* ini kita gunakan ketika kita ingin mengambil banyak *range* untuk melakukan proses pemrograman dan mengeluarkan *output*. Berikut langkah-langkah lengkap dari kode pemrograman diatas:

1. Pada baris kode 3 kita memanggil *scanner* dari *java.util* karena kita membutuhkan *input* nilai dari user.
2. Pada baris kode 8 kita mendeklarasikan variabel *nilai* sebagai jenis data *integer*.
3. Pada baris kode 9-12 kita meminta *input* nilai dari user.
4. Pada baris kode 14-23, kita menggunakan statement *nested if-else* untuk menentukan *ouput* berdasarkan kondisi:
 - a. Jika nilai ≥ 80 print A, jika tidak,
 - b. Nilai ≥ 70 print B, jika tidak,
 - c. Nilai ≥ 60 print C, jika tidak,
 - d. Nilai ≥ 50 print D, jika tidak,

e. Print E

Dari seluruh langkah-langkah yang kita lakukan, kita akan mendapatkan syarat bertingkat dalam menghasilkan *output* dikarenakan kita menggunakan *nested if-else*. Untuk *output* yang dihasilkan tergantung pada *input* yang dimasukkan oleh user. Berikut salah satu dari contoh *output* dari kode program diatas:



```
<terminated> Nilai [Java Application] D:\Eclipse\eclipse\plugins\  
Inputkan nilai angka :  
65  
|
```

Gambar 2.10: salah satu output kode program “Nilai”

Karena *input* nilai 65 lebih kecil daripada 70 dan lebih besar daripada 60, maka dari statement *nested if-else* yang kita punya, program akan menghasilkan *output* berupa C.

2.6. Praktikum “NamaBulan”

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class NamaBulan {
6
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.println("Masukkan angka bulan (1-12) : ");
10        int bulan = scanner.nextInt();
11        switch (bulan) {
12            case 1 :
13                System.out.println("Januari");
14                break;
15            case 2 :
16                System.out.println("Februari");
17                break;
18            case 3 :
19                System.out.println("Maret");
20                break;
21            case 4 :
22                System.out.println("April");
23                break;
24            case 5 :
25                System.out.println("Mei");
26                break;
27            case 6 :
28                System.out.println("Juni");
29                break;
30            case 7 :
31                System.out.println("Juli");
32                break;
33            case 8 :
34                System.out.println("Agustus");
35                break;
36            case 9 :
37                System.out.println("September");
38                break;

```

Gambar 2.11: kode program praktikum “NamaBulan”

```

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 2.12: kode program 2 praktikum “NamaBulan”

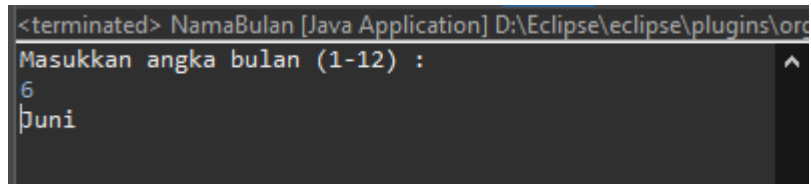
Pada kode program kali ini, dibanding menggunakan statement *if*, *if-else*, ataupun *nested if-else*, kita akan menggunakan statement *switch*. Statement *switch*

ini akan lebih mudah digunakan ketika kita tidak memiliki rentang nilai sebagai penentu percabangan. Statement *switch* juga bisa kita gunakan ketika kita memiliki lebih banyak eksekusi seperti praktikum “Nilai” yang kita lakukan. Berikut langkah-langkah kerja program yang menggunakan statement *switch*.

1. Pada baris kode 3 kita memanggil *scanner* dari *java.util* karena kita membutuhkan *input* dari user.
2. Pada baris kode 8-9 kita meminta *input* berupa angka dari user.
3. Pada baris kode 10 kita deklarasi variabel bulan dengan tipe data *integer*.
4. Pada baris kode 11-49 kita lakukan statement *switch* dengan keterangan sebagai berikut:
 - a. Jika case 1, maka *output* yang dihasilkan adalah Januari,
 - b. Jika case 2, maka *output* yang dihasilkan adalah Februari,
 - c. Jika case 3, maka *output* yang dihasilkan adalah Maret,
 - d. Jika case 4, maka *output* yang dihasilkan adalah April,
 - e. Jika case 5, maka *output* yang dihasilkan adalah Mei,
 - f. Jika case 6, maka *output* yang dihasilkan adalah Juni,
 - g. Jika case 7, maka *output* yang dihasilkan adalah Juli,
 - h. Jika case 8, maka *output* yang dihasilkan adalah Agustus,
 - i. Jika case 9, maka *output* yang dihasilkan adalah September,
 - j. Jika case 10, maka *output* yang dihasilkan adalah Oktober,
 - k. Jika case 11, maka *output* yang dihasilkan adalah November,
 - l. Jika case 12, maka *output* yang dihasilkan adalah Desember

Jika kita lihat pada kode program pada gambar 2.11 dan gambar 2.12, kita menggunakan *break* pada setiap *case*. Penggunaan *break* ini bertujuan untuk menghentikan eksekusi dari *block switch* setelah *case* yang cocok telah ditemukan dan dieksekusi.

Output yang dihasilkan dari kode program diatas dapat menghasilkan 13 variasi sesuai dengan 12 *case* yang kita miliki dan satu *default*. Berikut salah satu contoh *output* dari kode program yang akan kita jalankan:



```
<terminated> NamaBulan [Java Application] D:\Eclipse\ eclipse\plugins\org
Masukkan angka bulan (1-12) :
6
Juni
```

Gambar 2.13: salah satu output kode program “NamaBulan”

Karena *input* yang dimasukkan adalah 6, maka program akan menjalankan *case* 6 sesuai dengan statement *switch* yang kita gunakan.

BAB III

KESIMPULAN

3.1. Kesimpulan

Dari seluruh praktek dan landasan teori yang kita miliki, kita dapat memahami bahwasanya penggunaan statement kondisional pada pemograman Java sangat dibutuhkan agar komputer bisa menganalisis langkah selanjutnya pada pemograman berdasarkan kondisi tertentu. Statement kondisional pada pemograman Java ada enam jenis, namun yang kita gunakan dan pelajari pada praktikum kali ini hanya dua jenis, yakni statement *if* dan statement *switch*.

Statement *if* sendiri memiliki tiga macam, yakni *if*, *if-else*, dan *nested-if-else*. Pemakaian pada setiap jenisnya ini berbeda tergantung apa yang kita butuhkan dalam pemograman. Sementara untuk statement *switch*, kita menggunakan kasus (*case*) dalam menentukan percabangan dan menutup *block switch* dengan *break* setelah *case* dieksekusi. Penggunaan statement *if* dan statement *switch* bergantung pada jenis pemograman apa yang akan kita gunakan.

3.2. Saran

Untuk dapat menguasai bahasa pemograman Java dan membuat suatu program yang responsif, kita harus memahami tentang statement kondisional yang ada pada pemograman Java. Pada praktikum kali ini kita hanya mempelajari dua dari enam jenis statement kondisional yang ada pada bahasa Java. Untuk dapat lebih berkembang, tentunya kita bisa mempelajari statement kondisional yang lain dan mencoba membuat sebuah program dengan menggunakan statement kondisional ini.

DAFTAR PUSTAKA

- [1] David J. Eck, *Introduction to Programming Using Java*, 9th ed., JavaFX Edition. Geneva, NY, USA: Hobart and William Smith Colleges, May 2022 [Daring]. Tersedia pada: <https://math.hws.edu/javanotes/>
- [2] Oracle Corporation, “The if-then and if-then-else Statements,” *Java Tutorials: Learning the Java Language*, 2024. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html> [Diakses: 02-Okt-2025]
- [3] Oracle Corporation, “The switch Statement,” *Java Tutorials: Learning the Java Language*, 2024. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html> [Diakses: 02-Okt-2025]