

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“STATEMENT KONDISIONAL DI JAVA”

Disusun Oleh :

Dzhillan Dzhilila

2511531001

Dosen Pengampu:

Dr. Wahyudi, S.T, M.T

Asisten Praktikum:

Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, sehingga penulis dapat menyelesaikan Laporan Praktikum Algoritma Pemrograman dan Pemrograman dengan baik. Laporan ini disusun sebagai dokumentasi dan refleksi atas kegiatan praktikum yang telah dilaksanakan, dengan fokus utama pada pemahaman konsep statement kondisional yang ada pada bahasa pemrograman Java.

Melalui praktikum ini, penulis mempelajari berbagai pernyataan kondisional seperti *if*, *if-else*, *if-else-if*, serta *switch-case* yang merupakan konsep utama dalam pengambilan keputusan dalam bahasa pemrograman Java. Dalam praktikum ini, implementasi dari berbagai kendali percabangan pada proyek sederhana membantu penulis untuk memahami logika di balik alur kendali program berdasarkan kondisi tertentu.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat di harapkan demi perbaikan dan pengembangan di masa mendatang.

Semoga laporan ini dapat memberikan manfaat, baik sebagai bahan pembelajaran bagi penulis maupun referensi bagi pembaca dalam memahami dasar-dasar pemrograman komputer.

Padang, 2025

Dzhillan Dzhaila

251531001

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB 1.....	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Manfaat	2
BAB 2.....	3
PEMBAHASAN	3
2.1 Program If.....	3
2.1.1 Uraian Langkah Kerja	3
2.1.2 Penjelasan Teori	4
2.2 Program Multiple If	5
2.2.1 Uraian Langkah Kerja	5
2.2.2 Penjelasan Teori	6
2.3 Program If- else	7
2.3.1 Uraian Langkah Kerja	7
2.3.2 Penjelasan Teori	8
2.4 Program If-else if -else.....	10
2.4.1 Uraian Langkah Kerja	10
2.4.2 Penjelasan Teori	11
2.5 Program Stiwch-Case.....	12
2.5.1 Uraian Langkah Kerja	13
2.5.2 Penjelasan Teori	15
BAB III.....	17
KESIMPULAN	17
DAFTAR PUSTAKA	18

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman terkhusus pemrograman dengan bahasa Java, pernyataan kondisional menjadi konsep penting yang harus dipahami. Percabangan merupakan sebuah cara yang digunakan dalam suatu program untuk mengambil keputusan. Pemrograman yang menggunakan pernyataan kondisional akan mampu mengambil suatu keputusan berdasarkan kondisi tertentu. Pada saat suatu program tidak memiliki pernyataan kondisional, maka pemrograman itu hanya beroperasi secara linear dan tidak mampu menilai berbagai percabangan yang menyimbolkan berbagai situasi yang berbeda dalam satu eksekusi. Oleh karena itu, memahami konsep pernyataan kondisional menjadi salah satu tahap inti dalam pembelajaran algoritma pemrograman dan pemrograman.

Pernyataan kondisional (Statement Kondisional) pada bahasa pemrograman bahasa Java diantaranya adalah statement *if*, *if-else*, *if-else-if*, dan *switch-case*. Statemen kondisional inilah yang bertugas untuk menganalisis dan mengevaluasi suatu kondisi sehingga pemrograman dapat menentukan aksi apa yang harus dikerjakan ketika sebuah kondisi terpenuhi. Berbagai jenis pernyataan kondisional tentunya memiliki karakteristik penggunaan yang berbeda-beda sesuai dengan konteks kebutuhan. Pemahaman dari cara kerja masing-masing struktur percabangan, penggunaannya, serta implementasinya pada proyek sederhana menjadi harapan dari terlaksananya praktikum ini.

1.2 Tujuan

1. Memahami konsep dasar statement kondisional dalam bahasa pemrograman Java.
2. Mampu memahami dan mengimplementasikan jenis-jenis statement kondisional, cara kerja, serta penggunaannya dalam kode program yang menggunakan struktur kendali kondisional.
3. Mampu mengembangkan pemahaman terhadap alur eksekusi program berdasarkan kondisi tertentu.

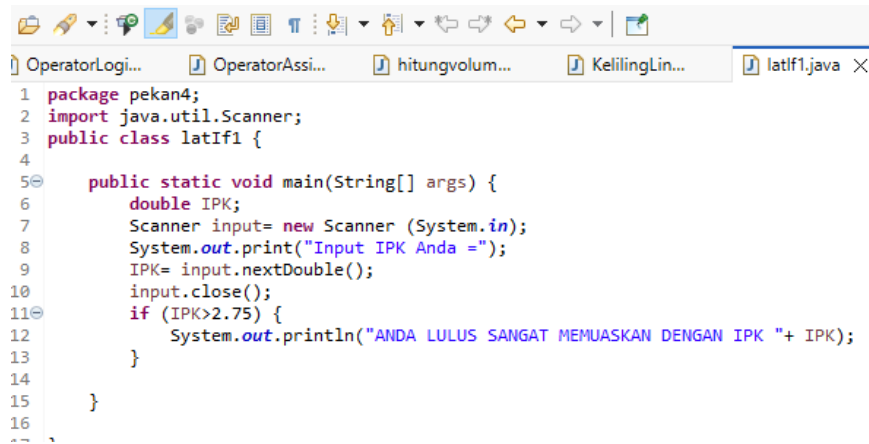
1.3 Manfaat

1. Memberikan pengalaman kepada mahasiswa dalam menulis program dalam bahasa Java dengan menggunakan statement kondisional atau logika percabangan.
2. Memperkuat fondasi dan menambah bekal awal dalam memahami struktur kendali program dalam pemrograman dalam bahasa Java.

BAB 2

PEMBAHASAN

2.1 Program If



```
1 package pekan4;
2 import java.util.Scanner;
3 public class latIf1 {
4
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        }
14    }
15 }
16
17 }
```

Gambar 2.1 Source Code Program latIf1.java

2.1.1 Uraian Langkah Kerja

1. Import Library = `import java.util.Scanner;` (program mengimpor kelas *Scanner* dari paket *java.util* untuk membaca input dari pengguna melalui keyboard.
2. Deklarasi Kelas = `public class latIf1 {`
3. Deklarasi Main Method = `public static void main (String [] args)`
4. Deklarasi Variabel = `double IPK;` (mendeklarasikan variabel IPK dengan tipe data *double*).
5. Objek *Scanner*
`Scanner input = new Scanner (System.in)` : membuat objek input dari kelas *Scanner* untuk membaca data dari aliran masukan standar (*System.in*)
6. Menampilkan pesan input = `System.out.print("Input IPK Anda =`
`"`)
7. Membaca Nilai IPK dari Pengguna = `IPK= input.nextDouble();`
8. Menutup *Scanner* = `input.close();`

9. Pengecekan dengan kondisi *if* = *If* ($IPK > 2.75$){
 - a. Jika **Ya**, pesan “ANDA LULUS SANGAT MEMUASKAN DENGAN IPK [nilai IPK]” akan muncul sebagai output.
 - b. Jika **Tidak**, maka menghasilkan output baris kosong karena tak berisi apa-apa.
10. Eksekusi dan Output Program
 - a. Jika nilai IPK yang diinputkan oleh pengguna >2.75

```
Input IPK Anda = 3
ANDA LULUS SANGAT MEMUASKAN DENGAN IPK 3.0
```

Gambar 2.2 Output Program latIf1

- b. Jika nilai IPK yang diinputkan oleh pengguna <2.75

```
terminated> latIf1 [Java Application] C:\Users\HI
nput IPK Anda = 2
```

Gambar 2.3 Output Program latIf1

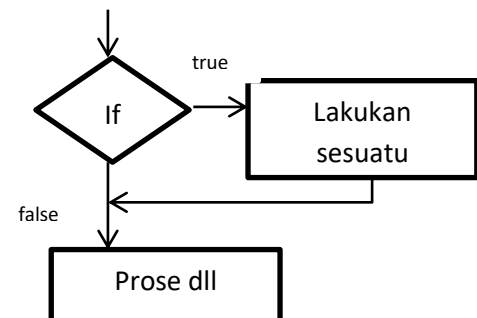
2.1.2 Penjelasan Teori

Statement IF atau percabangan IF adalah suatu percabangan yang hanya memiliki satu pilihan. Blok kode setelah pernyataan IF hanya akan dikerjakan jika kondisi terpenuhi atau bernilai benar (*true*). Jika suatu kondisi tidak terpenuhi atau bernilai salah (*false*), maka blok kode setelah pernyataan IF akan dilewati.

Bentuk umum :

```
If (kondisi){
    Pernyataan 1;
    Pernyataan 2;
}
```

Contoh flowchart IF



2.2 Program Multiple If

```

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

```

Gambar 2.4 : Source Code Program multiIf.java

2.2.1 Uraian Langkah Kerja

1. Import Library = `import java.util.Scanner;` (program mengimpor kelas *Scanner* dari paket *java.util* untuk membaca input dari pengguna melalui keyboard.
2. Deklarasi Kelas = `public class multiIf{`
3. Deklarasi Main Method = `public static void main (String [] args)`
4. Deklarasi Variabel
 - a. `int umur ;` (mendeklarasikan variabel umur dengan tipe data *int*).
 - b. `char sim;` (mendeklarasikan variabel sim dengan tipe data *char*)
5. Objek *Scanner*

Scanner a = new Scanner (System.in) : membuat objek a dari kelas *Scanner* untuk membaca data dari aliran masukan standar (System.in)

6. Meminta input usia = System.out.print("Input umur anda: ");
7. Membaca nilai usia yang telah diinputkan = umur = a.nextInt();
8. Meminta input kepemilikan SIM = System.out.print("Apakah Anda Sudah Punya SIM C?: ");
9. Membaca input = sim = a.next().charAt(0); (membaca input sebagai *string*, lalu (charAt(0)) mengambil huruf pertama dari input.
10. Menutup Scanner = a.close();
11. Pengecekan kondisi dengan Multi If
 - a. Sudah dewasa dan punya SIM.
 - b. Sudah dewasa dan tidak punya SIM.
 - c. Belum cukup umur dan tidak punya SIM.
 - d. Belum cukup umur tapi punya SIM.
12. Eksekusi dan Hasil Output

```
Input umur anda: 17
Apakah Anda Sudah Punya SIM C?: n
Anda Sudah dewasa tetapi tidak boleh bawa motor
```

Gambar 2.5 : Output Salah Satu Kondisi Program multiIf.java

2.2.2 Penjelasan Teori

Statement Multi IF adalah struktur percabangan dalam pemrograman berupa gabungan dari beberapa statement If. Statement Multi IF akan mengevaluasi beberapa kondisi secara independen dan semuanya akan dievaluasi oleh program. Sehingga, walaupun kondisi yang bernilai benar sudah ditemukan, program akan terus melakukan evaluasi terhadap kondisi kondisi setelahnya.

Lalu, statement ini juga bisa mengeksekusi lebih dari satu blok if jika kondisinya memang terpenuhi.

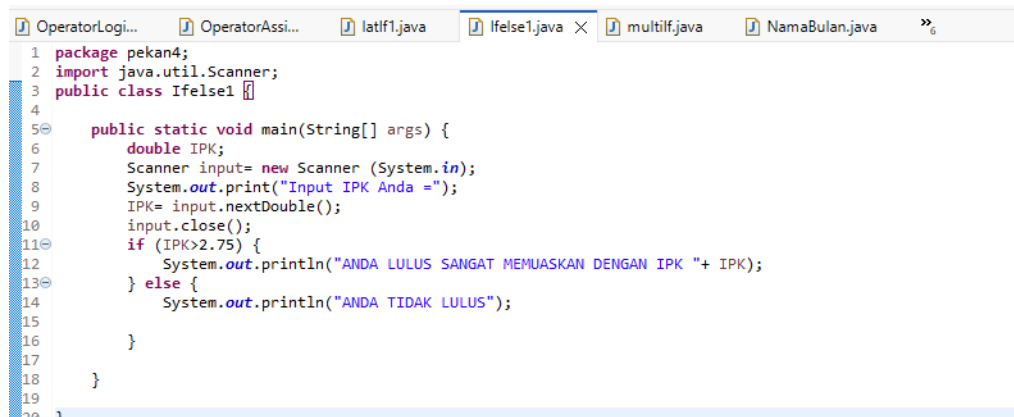
Bentuk umum :

```
if (kondisi1) {
}

if (kondisi2) {
}

if (kondisi3) {
}
```

2.3 Program If- else



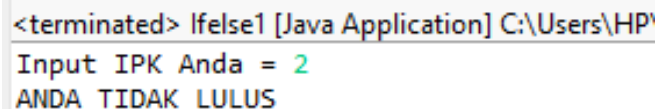
```
1 package pekan4;
2 import java.util.Scanner;
3 public class Ifelse1 {
4
5     public static void main(String[] args) {
6         double IPK;
7         Scanner input= new Scanner (System.in);
8         System.out.print("Input IPK Anda =");
9         IPK= input.nextDouble();
10        input.close();
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 }
```

2.6 Source Code Program Ifelse1.java

2.3.1 Uraian Langkah Kerja

1. Import Library = `import java.util.Scanner;` (program mengimpor kelas *Scanner* dari paket *java.util* untuk membaca input dari pengguna melalui keyboard.
2. Deklarasi Kelas = `public class Ifelse1{`

3. Deklarasi Main Method = `public static void main (String [] args)`
4. Deklarasi Variabel = `double IPK;` (mendeklarasikan variabel IPK dengan tipe data *double*).
5. Objek *Scanner*
`Scanner input = new Scanner (System.in)` : membuat objek input dari kelas *Scanner* untuk membaca data dari aliran masukan standar (*System.in*)
6. Menampilkan pesan input = `System.out.print("Input IPK Anda = ")`
7. Membaca Nilai IPK dari Pengguna = `IPK= input.nextDouble();`
8. Menutup *Scanner* = `input.close();`
9. Pengecekan dengan kondisi *if-else* = `If (IPK > 2.75){`
 - a. Jika **Ya**, pesan "ANDA LULUS SANGAT MEMUASKAN DENGAN IPK [nilai IPK]" akan muncul sebagai output
 - b. Jika **Tidak**, maka menghasilkan output "ANDA TIDAK LULUS"
10. Eksekusi dan Output Program



```
<terminated> Ifelse1 [Java Application] C:\Users\HP\
Input IPK Anda = 2
ANDA TIDAK LULUS
```

2.7 Output Kode Program Ifelse1.java

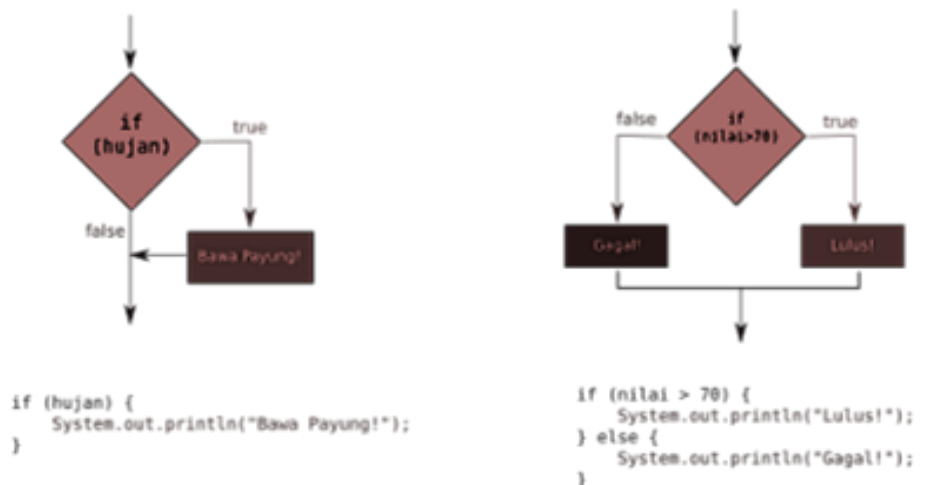
2.3.2 Penjelasan Teori

Statement If-else adalah pengambil keputusan yang digunakan untuk mengontrol alur program berdasarkan satu kondisi. Statement ini akan mengeksekusi suatu blok kode dan apabila salah satu blok kode memenuhi kondisi benar, maka proses eksekusi atau evaluasi

akan berhenti pada blok kode tsb. Namun jika kondisi tidak terpenuhi maka pernyataan pada bagian ELSE yang akan dieksekusi.

Bentuk umum:

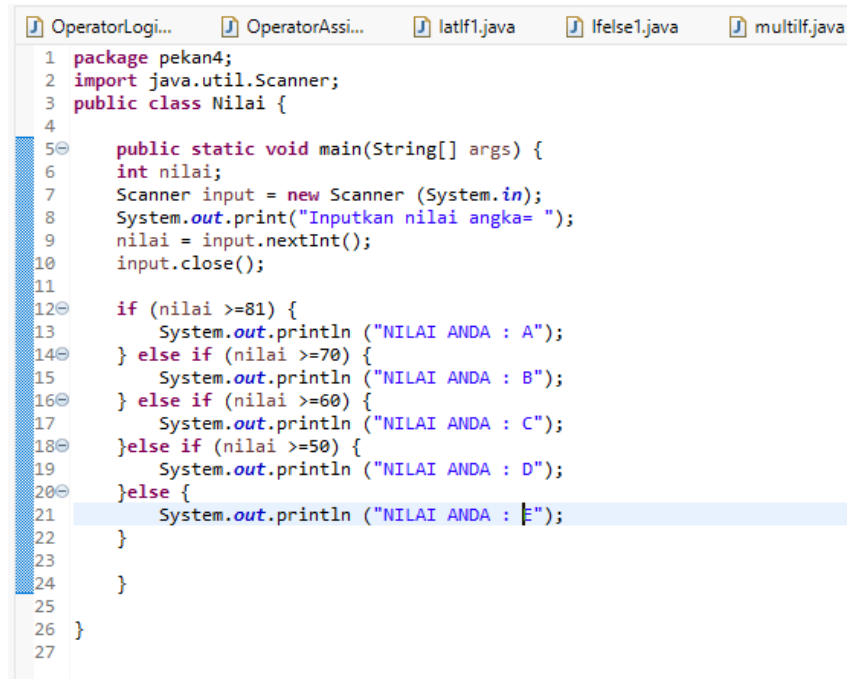
```
If (kondisi){
    Pernyataan 1;
    Pernyataan 2;
} else {
    alternative_pernyataan1;
    alternative_pernyataan2;
}
```



Gambar 2.8 : Contoh alur flowchart yang menggunakan if-else

https://1.bp.blogspot.com/-EBFXR8YanHg/ViyfSNX80-I/AAAAAAAAACEI/BVBV3EX_r3Q/s1600/Percabangan%2Bdi%2Bpemrograman%2Bjava.png

2.4 Program If-else if -else



```

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

```

Gambar 2.9 : Source Code dari Program Nilai.java

2.4.1 Uraian Langkah Kerja

1. Import Library = `import java.util.Scanner;` (program mengimpor kelas *Scanner* dari paket *java.util* untuk membaca input dari pengguna melalui keyboard.
2. Deklarasi Kelas = `public class Nilai{`
3. Deklarasi Main Method = `public static void main (String [] args)`
4. Deklarasi Variabel = `int nilai;` (mendeklarasikan variabel IPK dengan tipe data *int*).
5. Objek *Scanner*
`Scanner input = new Scanner (System.in)` : membuat objek input dari kelas *Scanner* untuk membaca data dari aliran masukan standar (*System.in*)
6. Menampilkan Pesan Input = `System.out.print ("Inputkan nilai angka= ")`
7. Membaca nilai dari Pengguna = `nilai = input.nextInt();`

8. Menutup *scanner* = `input.close();`
9. Pengecekan kondisi dengan *if-else if-else*
 - a. Kondisi 1 : Nilai ≥ 81 = Grade A
 - b. Kondisi 2 : Nilai > 70 (dan < 81) = Grade B
 - c. Kondisi 3 : Nilai > 60 (dan ≤ 70) = Grade C
 - d. Kondisi 4 : Nilai ≥ 50 (dan ≤ 60) = Grade D
 - e. Kondisi 5: Nilai < 50 = Grade E
10. Eksekusi dan Output Program

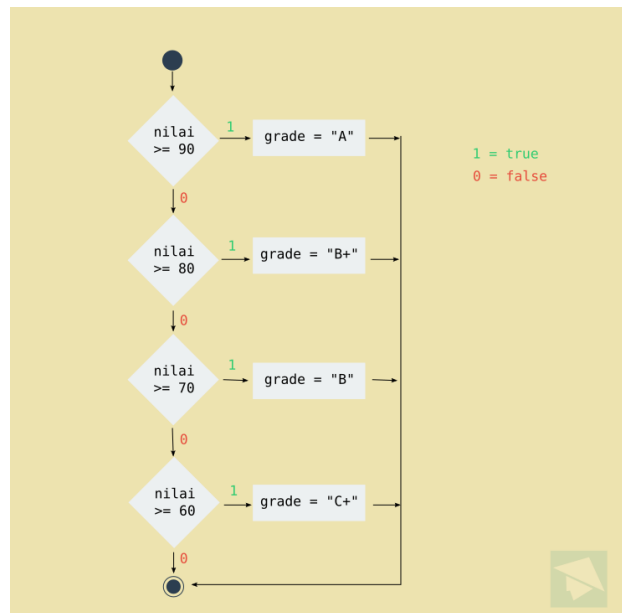


```
<terminated> Nilai [Java Application] C:\Users\HP\.p2
Inputkan nilai angka= 45
NILAI ANDA : E
```

Gambar 2.10 : Ouput Kondisi ELSE dari program Nilai.java

2.4.2 Penjelasan Teori

Saat percabangan IF ELSE hanya memiliki dua pilihan saja. IF ELSE IF ELSE memiliki lebih dari dua pilihan. Ketika menggunakan statement IF ELSE IF, program akan membaca kondisi dari atas ke bawah secara sekuesnsial. Membaca secara sekuensial berarti kondisi 1 akan dicek apakah bernilai benar atau tidak. Jika kondisi 1 bernilai benar maka blok kode setelahnya akan dijalankan, Jika kondisi 1 salah, maka pengecekan akan berlanjut ke kondisi ke-2 dan begitu seterusnya. Hingga pada semua kondisi tidak ada yang bernilai benar atau kondisi if dan else if bernilai salah, maka program akan mengeksekusi pernyataan alternatif pada bagian ELSE. Berikut contoh algoritma Flowchart dari program yang menggunakan statement *if- else if-else*.



Gambar 2.11 : Contoh Flowchart Program dengan *if-else if-else*

<https://3.bp.blogspot.com/-9uS-c73seec/VkBISDA0AXI/AAAAAAAAACLU/AsWctwfCN0/s1600/flowchart%2Bpercabangan%2Byang%2Bmemiliki%2Bbanyak%2Bpilihan%2Bkeputusan.png>

2.5 Program Stiwch-Case

```

package pekan4;
import java.util.Scanner;
public class NamaBulan {

    public static void main(String[] args) {
        Scanner scanner = new Scanner (System.in);
        System.out.print("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");
                break;
            case 11 :
                System.out.println("November");
                break;
            case 12 :
                System.out.println("Desember");
                break;
            default :
                System.out.println("Angka bulan tidak valid");
        }
    }
}
  
```

Gambar 2.12 :Source Code Program Namabulan.java

```

        case 8 :
            System.out.println("Agustus");
            break;
        case 9 :
            System.out.println("September");
            break;
        case 10 :
            System.out.println("Oktober");
            break;
        case 11 :
            System.out.println("November");
            break;
        case 12 :
            System.out.println("Desember");
            break;
        default :
            System.out.println("Angka tidak valid");

    }
    scanner.close();
}

```

Gambar 2.13 : Source Code Lanjutan dari Program Namabulan.java

2.5.1 Uraian Langkah Kerja

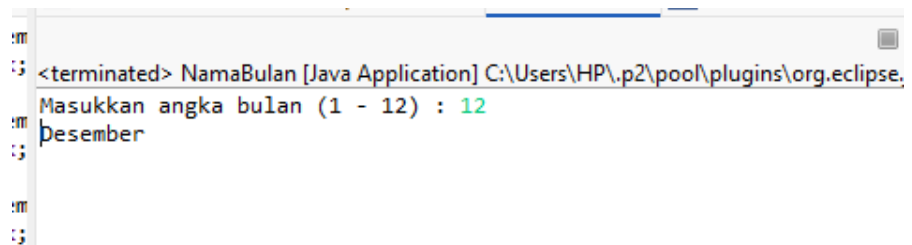
1. Import Library = `import java.util.Scanner;` (program mengimpor kelas *Scanner* dari paket *java.util* untuk membaca input dari pengguna melalui keyboard.
2. Deklarasi Kelas = `public class Namabulan{`
3. Deklarasi Main Method = `public static void main (String [] args)`
4. Objek *Scanner*
`Scanner scanner = new Scanner (System.in)` : membuat objek input dari kelas *Scanner* untuk membaca data dari aliran masukan standar (*System.in*)
5. Meminta Input Angka Bulan.
6. Menyimpan nilai input ke dalam variabel bulan.
7. Menggunakan Struktur *switch-case* untuk menentukan nama bulan, lalu menjalankan blok/ *case* yang sesuai. Berikut penggunaan struktur *if-else if – else* pada program menentukan nama bulan.


```
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");
        break;
    case 11:
        System.out.println("November");
        break;
    case 12:
        System.out.println("Desember");
        break;
    default:
        System.out.println("Angka tidak valid");
    }
}
8. Menutup Scanner = scanner.close();
9. Eksekusi dan Output Program

```



Gambar 2.14 : Output dari Program Namabulan.java

2.5.2 Penjelasan Teori

Struktur percabangan *switch-case* sebetulnya hampir sama dengan percabangan IF ELSE dimana jika di percabangan IF terdapat beberapa kondisi pada pemilihan *switch* akan ada beberapa case yang dapat dibuat. Jika nilai yang menjadi nilai pembanding , sama dengan case yaang ada atau kata lainnya, jika suatu *case* pada pemilihan *switch* bernilai *true* maka pernyataan pada *case* tersebut yang akan dieksekusi .

Bentuk Umum :

```
switch (ekspresi) {  
  case1;  
    pernyataan 1;  
    break;  
  case 2;  
    pernyataan 2;  
    break;  
  default :  
    pernyataan_default;  
}
```

Catatan :

- a. **Break** : artinya berhenti. Berfungsi untuk memberhentikan mengecek case yang lainnya
- b. **Default** : artinya jika nilai variabel tidak ada yang sama dengan pilihan *case*, maka kode yang akan dikerjakan adalah kode yang ada dalam *default*. Pilihan *default* juga tidak memiliki *break*, karena dia adalah pilihan terakhir. Akhirnya pengecekan akan berakhir di situ.

BAB III

KESIMPULAN

Statement Kondisional secara *general* digunakan untuk mengontrol alur program berdasarkan kondisi tertentu. Masing-masing struktur percabangan memiliki cara penggunaan dan ketentuan yang berbeda-beda. Berdasarkan praktikum, penjelasan singkat mengenai penggunaan statemen kondisional adalah sebagai berikut;

1. Statement If

Digunakan saat kita hanya ingin melihat output satu kondisi yang telah terpenuhi. Statement ini tidak akan membuat program tidak menampilkan apa apa ketika suatu kondisi bernilai salah / *false*.

2. Statement Multiple If

Merupakan gabungan dari beberapa statement IF, dan digunakan ketika kita ingin menguji semua kondisi yang ada. Penggunaan statement ini membuat program bisa mengeksekusi lebih dari satu blok IF jika kondisinya memang terpenuhi.

3. Statement If-else

Statement ini merupakan pengembangan dari statement IF. Berbeda dengan IF yang tidak akan mengeksekusi blok kode manapun ketika suatu kondisi bernilai salah, statement IF ELSE akan mengeksekusi pernyataan dalam blok ELSE saat suatu kondisi bernilai salah.

4. Statement If-else If- else

Statement ini akan membuat program mengevaluasi semua kondisi dari beberapa pernyataan sampai menemukan suatu kondisi yang bernilai benar(ELSE IF) . Namun, jika tidak ditemukan, kondisi yang akan dieksekusi adalah pernyataan yang berada dalam konsep ELSE.

5. Statement switch-case

Pada statement ini, jika nilai yang menjadi nilai pembanding , sama dengan case yaang ada atau kata lainnya, jika suatu *case* pada pemilihan *switch* bernilai *true* maka pernyataan pada *case* tersebut yang akan dieksekusi .

DAFTAR PUSTAKA

- [1] Dr. Dedek Indra Gunawan Hts, S.Kom, M.Kom., Abdul Meizar, S.Kom, M.Kom., *Pemrograman Berorientasi Objek Dengan Java*, 1st ed, Indramayu, 2024.
- [2] Petani Kode, “Memahami 3 Bentuk Percabangan dalam Java,” 2015. Tersedia pada: <https://www.petanikode.com/java-percabangan/>. [Diakses : 1 Okt 2025]
- [3] Baeldung, “Format Multiple ‘or’ Conditions in an If Statement in Java,” 2024. Tersedia pada: <https://www.baeldung.com/java-multiple-or-conditions-if-statement>. [Diakses : 1 Okt 2025]
- [3] Geeksforgeeks, “Java if-else Statement,” 2024. [Daring}. Tersedia pada: <https://www.geeksforgeeks.org/java/java-if-else-statement-with-examples/> . [Diakses : 2 Okt 2025].