

LAPORAN PRAKTIKUM

PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

“STRUKTUR PERCABANGAN DALAM BAHASA PEMROGRAMAN JAVA”

disusun oleh

Hafizh Habibullah

2511531002

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

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TAHUN 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat dan karunia-Nya, laporan praktikum ini dapat terselesaikan dengan baik. Laporan ini disusun untuk memenuhi salah satu tugas praktikum Pemrograman Dasar, dengan pokok bahasan mengenai **struktur percabangan dalam bahasa pemrograman Java**.

Melalui praktikum ini, penulis mempelajari bagaimana cara program melakukan pengambilan keputusan berdasarkan kondisi tertentu. Struktur percabangan merupakan konsep penting dalam pemrograman, karena memungkinkan program berjalan secara fleksibel dan menyesuaikan alur eksekusi sesuai syarat yang diberikan.

Semoga laporan ini dapat bermanfaat bagi pembaca dalam memahami dasar-dasar struktur percabangan pada pemrograman.

Padang, 29 September 2025

Hafizh Habibullah

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR LAMPIRAN	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan.....	1
1.3 Manfaat	1
BAB II PEMBAHASAN	2
2.1 Program latIf1 dan IFelse1	2
2.2 Program multiIf.....	4
2.3 Kode Program Nilai	5
2.4 Kode Program Nama Bulan	6
BAB III KESIMPULAN.....	8
3.1 Kesimpulan	8
3.2 Saran Pengembangan	8
DAFTAR PUSTAKA.....	9

DAFTAR LAMPIRAN

Gambar 2. 1 - Kode Program latIf1	2
Gambar 2. 2 - Kode Program IFelse1	3
Gambar 2. 3 - Kode Program multiIf	4
Gambar 2. 4 - Kode Program “Nilai”	5
Gambar 2. 5 - Kode Program NamaBulan	6

Tabel 2. 1 – Contoh Hasil Output dari latIF1	2
Tabel 2. 2 - Contoh Hasil Output dari IFelse1 jika Input sesuai dengan blok else .	3
Tabel 2. 3 – Contoh Hasil Output dari multiIf	5
Tabel 2. 4 - Contoh Hasil Output dari Program “Nilai”	6
Tabel 2. 5 - Contoh Hasil Output dari NamaBulan	7
Tabel 2. 6 - Contoh lain dari Hasil Output dari NamaBulan	7

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, pengaturan alur logika adalah salah satu elemen dasar yang menentukan cara kerja sebuah program. Salah satu cara pengendalian alur tersebut adalah struktur percabangan, yang berfungsi untuk menentukan arah eksekusi berdasarkan kondisi tertentu. Dengan adanya percabangan, program tidak berjalan secara linier lagi, tetapi dapat beradaptasi dengan kondisi yang ada.

Bahasa pemrograman Java menawarkan struktur percabangan berupa **if**, **else-if**, dan **switch-case**. Struktur ini sering dipakai karena dapat mengelola lebih dari satu keadaan secara bertahap. Pemahaman tentang percabangan sangat krusial, karena hampir semua aplikasi interaktif memerlukan proses pengambilan keputusan agar dapat beroperasi dengan baik sesuai keinginan pengguna.

1.2 Tujuan

Tujuan dari praktikum ini adalah:

1. Memahami konsep dasar struktur percabangan pada bahasa pemrograman Java.
2. Menerapkan penggunaan **if-else if-else** untuk menyelesaikan permasalahan logika.
3. Melatih keterampilan menyusun program yang mampu menyesuaikan jalannya eksekusi sesuai kondisi yang diberikan.

1.3 Manfaat

Praktikum ini diharapkan memberikan beberapa manfaat, di antaranya:

1. Memberikan pemahaman yang lebih jelas mengenai penerapan logika percabangan dalam pemrograman.
2. Menjadi dasar dalam pengembangan program yang lebih kompleks dan dinamis.
3. Membantu meningkatkan kemampuan analisis dalam merancang solusi berbasis kode.

BAB II

PEMBAHASAN

2.1 Program latIf1 dan IFelse1

Pada bagian ini, dibuat dua Kode Program yang sangat identik satu sama lain. Kode Program **latIf1** yang dibuat untuk menampilkan bagaimana cara menggunakan **"if"** pada bahasa pemrograman Java, sedangkan pada **IFelse1** ditambahkan penggunaan **"else"** untuk menampilkan cara penggunaannya bersamaan dengan **"if"**.

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

Gambar 2. 1 - Kode Program latIf1

Program ini ditulis dengan bahasa Java untuk menampilkan hasil kelulusan berdasarkan nilai IPK yang diberikan oleh pengguna. Pertama, variabel IPK dengan tipe `double` dideklarasikan untuk menyimpan input. Program selanjutnya menampilkan pesan "Masukkan IPK Anda =" dan menunggu pengguna untuk memasukkan nilai IPK melalui objek *Scanner*. Setelah data dimasukkan, nilainya disimpan ke dalam variabel IPK, kemudian *Scanner* ditutup. Selanjutnya, program mengecek apakah IPK melebihi 2.75. Apabila kriteria ini dipenuhi, maka akan muncul pesan "Anda Lulus Sangat Memuaskan dengan IPK ". Apabila syarat tidak dipenuhi, program tidak menghasilkan output lain karena tidak ada blok *else* yang dibuat.

Hasil Output

Input IPK Anda = 3.2

Anda Lulus Sangat Memuaskan dengan IPK 3.2

Tabel 2. 1 – Contoh Hasil Output dari latIF1

Sedangkan pada versi **IfElse1**, percabangan diperluas dengan menambahkan blok **else**. Dengan tambahan ini, program mampu menampilkan dua kemungkinan hasil. Jika nilai IPK lebih besar atau sama dengan 2.75, maka akan muncul pesan “Anda Lulus Sangat Memuaskan dengan IPK <nilai IPK>”. Sebaliknya, jika nilai IPK lebih kecil dari 2.75, program akan menampilkan pesan “Anda Tidak Lulus”.

```
package pekan4;
import java.util.Scanner;
public class IFelse1 {
    public static void main(String[] args) {
        double IPK;
        Scanner input=new Scanner(System.in);
        System.out.print("Input IPK Anda = ");
        IPK=input.nextDouble();
        input.close();
        if (IPK>2.75) {
            System.out.println("Anda Lulus Sangat Memuaskan dengan IPK "+IPK);
        } else {
            System.out.println("Anda Tidak Lulus");
        }
    }
}
```

Gambar 2. 2 - Kode Program IFelse1

Dari versi diatas, jika nilai IPK yang dimasukkan lebih kecil dari 2.75, maka Hasil Output akan menampilkan output sebagai berikut.

Hasil Output
Input IPK Anda = 2.5
Anda Tidak Lulus

Tabel 2. 2 - Contoh Hasil Output dari IFelse1 jika Input sesuai dengan blok else

2.2 Program multiIf

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

Gambar 2. 3 - Kode Program multiIf

Program **multiIf** merupakan pengembangan dari konsep percabangan, di mana logika keputusan tidak hanya melibatkan satu kondisi sederhana, melainkan kombinasi dari beberapa syarat. Program ini meminta dua input dari pengguna, yaitu **umur** dan **status kepemilikan SIM**. Nilai umur digunakan untuk menentukan apakah seseorang sudah cukup umur atau belum, sedangkan input SIM (y atau n) digunakan untuk mengecek apakah orang tersebut sudah memiliki SIM.

Berdasarkan gabungan kedua kondisi tersebut, program melaksanakan percabangan dengan berbagai kondisi **if**. Jika usia ≥ 17 tahun dan memiliki SIM, individu dianggap layak untuk mengendarai sepeda motor. Jika usia ≥ 17 tahun namun belum memiliki SIM, akan muncul pesan bahwa individu belum diizinkan berkendara. Sementara itu, jika umur kurang dari 17, program memberikan hasil yang berbeda berdasarkan kepemilikan SIM: belum cukup umur untuk berkendara, atau belum cukup umur tetapi sudah memiliki SIM.

Hasil Output

Input umur anda: 18

Apakah Anda Sudah Punya SIM C: y

Anda Sudah dewasa dan boleh bawa motor

Tabel 2. 3 – Contoh Hasil Output dari multiIf

2.3 Kode Program Nilai

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

Gambar 2. 4 - Kode Program “Nilai”

Program **Nilai** berfungsi untuk mengubah angka yang dimasukkan oleh pengguna menjadi sebuah nilai. Proses dimulai dengan memperoleh input nilai integer menggunakan objek *Scanner*. Nilai yang diinput disimpan dalam variabel *nilai*, lalu diproses menggunakan percabangan **if–else if–else**.

Logika percabangan menetapkan grade berdasarkan kisaran nilai. Jika nilai ≥ 81 maka akan muncul “A”. Jika skor berada dalam rentang 70–80 maka hasilnya

“B”. Nilai 61–69 mendapatkan "C", nilai 51–59 mendapatkan "D", sementara nilai ≤ 50 ditunjukkan sebagai "E". Melalui metode ini, program dapat secara otomatis mengelompokkan angka ke dalam kategori huruf yang lebih mudah dimengerti.

Hasil Output

Inputkan nilai angka= 80

B

Tabel 2. 4 - Contoh Hasil Output dari Program “Nilai”

2.4 Kode Program Nama Bulan

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

Gambar 2. 5 - Kode Program NamaBulan

Program **NamaBulan** berfungsi untuk menunjukkan nama bulan sesuai dengan angka input yang berkisar antara 1 hingga 12. Proses dimulai dengan

meminta pengguna untuk memasukkan angka bulan, kemudian nilai tersebut disimpan dalam variabel bulan. Struktur kontrol yang diterapkan adalah **switch-case**, di mana setiap angka dari 1 hingga 12 dihubungkan dengan nama bulan yang tepat, dimulai dari Januari hingga Desember.

Apabila pengguna mengetik angka di luar rentang 1 hingga 12, maka bagian default akan dieksekusi dan memberikan output “Angka tidak valid”. Dengan menggunakan **switch-case**, program ini menjadi lebih terorganisir dan lebih mudah dipahami, terutama saat menghadapi banyak opsi yang bersifat tetap seperti nama-nama bulan.

Hasil Output

masukkan angka bulan (1 - 12): 9
September

Tabel 2. 5 - Contoh Hasil Output dari NamaBulan

Hasil Output

masukkan angka bulan (1 - 12): 13
Angka tidak valid

Tabel 2. 6 - Contoh lain dari Hasil Output dari NamaBulan

BAB III

KESIMPULAN

3.1 Kesimpulan

Secara umum, praktikum ini memberikan wawasan mengenai penerapan struktur kontrol dalam pemrograman Java, terutama penggunaan if-else dan switch-case. Kedua bangunan ini mendukung proses pengambilan keputusan sesuai dengan situasi tertentu. Dengan penerapan program sederhana, tampak bahwa if-else lebih adaptif untuk kondisi yang rumit, sedangkan switch-case lebih singkat dan mudah dipahami untuk pemilihan yang bersifat diskrit seperti angka bulan.

3.2 Saran Pengembangan

Ke depan, program bisa ditingkatkan dengan menambahkan validasi input untuk lebih tahan terhadap kesalahan pengguna dan memperluas fitur seperti menampilkan informasi tambahan (misalnya: jumlah hari dalam bulan atau kategori nilai). Di samping itu, penerapan metode atau fungsi yang terpisah juga dapat dilakukan agar kode menjadi lebih teratur dan mudah untuk dirawat.

DAFTAR PUSTAKA

- [1] H. M. Deitel dan P. J. Deitel, *Java: How to Program*, 11th ed. Pearson, 2018.
- [2] Oracle, “The Java™ Tutorials: Control Flow Statements,” [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/flow.html> [Diakses: 29-Sep-2025].
- [3] R. S. Pressman, *Rekayasa Perangkat Lunak: Pendekatan Praktisi*, Yogyakarta: Andi, 2015.
- [4] W. Stallings, *Computer Science: An Overview*, 13th ed. Pearson, 2020.
- [5] S. A. Kusumadewi, *Algoritma dan Pemrograman dengan Java*, Yogyakarta: Graha Ilmu, 2019.