

LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA PEMROGRAMAN

IMPLEMENTASI STRUKTUR PERCABANGAN (*If- Else* dan *Switch-Case*)

disusun oleh:

Khairunnisa M.

NIM 2511532005

Dosen Pengampu:

DR. Wahyudi, S. T., M. T.

Asisten Pratikum:

Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis penjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya laporan praktikum dengan judul “Laporan Praktikum Pemrograman Java: Implementasi Struktur Percabangan (*If-Else* dan *Switch-Case*)” dapat diselesaikan dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu tugas praktikum mata kuliah pemrograman, khususnya pada materi mengenai struktur percabangan dalam Bahasa pemrograman Java. Melalui praktikum ini, penulis dapat memahami dan mengimplementasikan logika percabangan menggunakan perintah *if-else*, *if else if*, maupun *switch-case* untuk menyelesaikan berbagai permasalahan pemrograman.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi isi maupun penyajian. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan laporan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah membantu sehingga laporan ini dapat terselsaikan. Semoga laporan ini dapat bermanfaat bagi penulis sendiri maupun pembaca dalam memahami implementasi percabangan dalam pemrograman Java.

Padang, 29 September 2025

Khairunnisa M.

DAFTAR ISI

KATA PENGANTAR

DAFTAR ISI

BAB I

PENDAHULUAN

- 1.1. Latar Belakang
- 1.2. Tujuan Praktikum
- 1.3. Manfaat Praktikum

BAB II

PEMBAHASAN

- 2.1. Percabangan If Tunggal
- 2.2. Percabangan If Else
- 2.3. Percabangan Multi If
- 2.4. Percabangan if-else if
- 2.5. Percabangan switch-case

BAB III

KESIMPULAN

DAFTAR PUSTAKA

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam pemrograman, terdapat konsep dasar yang sangat penting yaitu struktur kendali atau *control structure*. Struktur kendali memungkinkan sebuah program untuk mengambil keputusan berdasarkan kondisi tertentu. Salah satu bentuk struktur kendali adalah percabangan (branching), yang digunakan agar program dapat mengeksekusi instruksi yang berbeda sesuai dengan syarat atau kondisi yang diberikan.

Bahasa pemrograman Java menyediakan beberapa bentuk percabangan, antara lain *if*, *if-else*, *if-else-if*, serta *switch-case*. Struktur *if* digunakan untuk mengeksekusi pernyataan apabila kondisi bernilai benar, sedangkan *if-else* memberikan alternatif apabila kondisi bernilai salah. Sementara itu, *switch-case* memudahkan pemilihan salah satu dari banyak pilihan dari berdasarkan nilai tertentu, sehingga program menjadi lebih sederhana dan terstruktur.

Pemahaman mengenai percabangan sangat penting bagi seorang programmer, karena hampir semua aplikasi dan system membutuhkan proses pengambilan keputusan. Contohnya, menentukan kelulusan mahasiswa berdasarkan IPK, menentukan izin membawa kendaraan berdasarkan umur dan kepemilikan SIM, mengonversi angka menjadi nama, bulan, serta mengubah nilai angka menjadi nilai huruf.

Melalui praktikum ini, mahasiswa diharapkan dapat memahami cara kerja struktur percabangan dalam Java serta mampu mengimplementasikannya dalam berbagai permasalahan pemrograman. Dengan demikian, mahasiswa tidak hanya memahami teori tetapi juga dapat menguasai penerapannya secara langsung.

1.2. Tujuan Praktikum

1. Memahami konsep dasar struktur percabangan dalam pemrograman Java.
2. Mampu mengimplementasikan percabangan *if*, *if-else*, dan *if-else if* untuk menyelesaikan permasalahan sederhana.
3. Mampu menggunakan struktur *switch-case* sebagai alternatif pengambilan keputusan dengan banyak kondisi.
4. Melatih kemampuan logika pemrograman dalam membangun program yang dapat mengambil keputusan secara otomatis berdasarkan input pengguna.
5. Mengembangkan keterampilan praktis dalam menulis, menguji, dan menganalisis program Java yang menggunakan percabangan.

1.3. Manfaat Praktikum

1. Mahasiswa memperoleh pemahaman cara kerja struktur percabangan dalam pemrograman Java serta perbedaan penggunaan *if-else* dan *switch-case*.
2. Mahasiswa dilatih untuk berfikir logis dalam menyelesaikan masalah dengan menggunakan percabangan yang sesuai dengan kebutuhan.
3. Mahasiswa mampu menuliskan kode program secara langsung, menjalankannya, serta menganalisis output yang dihasilkan.
4. Mahasiswa dapat memilih struktur percabangan yang paling tepat untuk kasus tertentu, sehingga program menjadi lebih efisien, terstruktur, dan mudah dipahami.

BAB II

PEMBAHASAN

2.1. Percabangan If Tunggal

Percabangan if tunggal mengevaluasi kondisi tertentu. Jika syarat bernilai benar, maka blok kode dijalankan. Jika tidak, program langsung berakhir tanpa alternatif lain.

```
1 package Pekan4;
2 import java.util.Scanner;
3
4 public class LatIf1 {
5     public static void main(String[] args) {
6         double IPK;
7         Scanner input=new Scanner (System.in);
8         System.out.println("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 }
18
```

Gambar 2.1. Program Java dengan Percabangan If Tunggal.

Program ini meminta input berupa IPK. Jika nilai IPK lebih besar dari 2.75, maka akan ditampilkan pesan lulus. Namun jika nilai urang atau sama dengan 2.75, program tidak menampilkan apa-apa.

Langkah kerja:

1. Membuat class baru dengan nama LatIf1.
2. Mendeklarasikan variabel IPK bertipe double.
3. Membuat objek *Scanner* untuk membaca input pengguna.
4. Meminta pengguna memasukkan nilai IPK.
5. Program memeriksa kondisi if (IPK > 2.75).

6. Jika kondisi benar, program menampilkan pesan kelulusan.
7. Jika kondisi salah, program tidak menampilkan pesan apapun.
8. Menutup objek *Scanner*.
9. Melakukan uji coba dengan memasukkan nilai IPK berbeda untuk melihat hasil.

Struktur *if* tunggal berguna untuk kondisi sederhana yang tidak memerlukan pernyataan alternatif. Program ini cocok jika hanya ada aksi Ketika kondisi terpenuhi.

2.2. Percabangan If-Else

Percabangan *if-else* digunakan untuk memberikan dua pilihan kondisi. Jika syarat terpenuhi, maka program menjalankan blok *if*, sedangkan jika tidak terpenuhi maka blok *else* yang akan dijalankan. Berikut contoh programnya:

```
1 package Pekan4;
2 import java.util.Scanner;
3
4 public class IfElse1 {
5     public static void main(String[] args) {
6         double IPK;
7         Scanner input=new Scanner (System.in);
8         System.out.println("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 }
```

Gambar 2.2. Program Java dengan Percabangan *If-Else*.

Program ini meminta input berupa IPK dari pengguna. Kemudian, program membandingkan apakah IPK tersebut lebih besar dari 2.75. Jika benar, pengguna dinyatakan lulus sangat memuaskan. Jika salah, pengguna dinyatakan tidak lulus.

Langkah kerja:

1. Membuat *project* baru pada Java.
2. Membuat *class* baru dengan nama *IfElse1*.

3. Mendeklarasikan variabel IPK bertipe *double*.
4. Menggunakan objek *Scanner* untuk menerima *input* dari pengguna.
5. Meminta pengguna memasukkan nilai IPK.
6. Membandingkan nilai IPK dengan kondisi if ($IPK > 2.75$).
7. Jika kondisi benar, program menampilkan pesan kelulusan sangat memuaskan.
8. Jika kondisi salah, program menampilkan pesan tidak lulus.
9. Menutup objek *Scanner* setelah selesai digunakan.
10. Menjalankan program untuk melihat hasil *output* sesuai *input*.

Program ini menunjukkan cara penggunaan *if-else*. Hasil *output* selalu memilih salah satu kondisi, sehingga jelas perbedaan antara *input* IPK tinggi dan rendah.

2.3. Percabangan Multi-If

Multi-if digunakan Ketika kita ingin mengevaluasi lebih dari satu kondisi tanpa menggunakan *else if*. Pada program ini, kombinasi umur dan kepemilikan SIM diperiksa menggunakan beberapa if terpisah.

```

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 membawa 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 }

```

Gambar 2.3. Program Java dengan Percabangan *Multi-if*.

Program meminta input umur dan status SIM pengguna. Setelah itu, program mengevaluasi kombinasi keduanya untuk menentukan apakah pengguna sudah boleh membawa motor atau belum.

Langkah kerja:

1. Membuat *class* baru dengan nama *MultiIf*.
2. Mendeklarasikan variabel umur (*int*) dan sim (*char*).
3. Menggunakan Scanner untuk meminta *input* umur dari pengguna.
4. Meminta *input* status SIM pengguna.
5. Mengevaluasi beberapa kondisi menggunakan *if* secara terpisah:
 - 5.1. Jika umur \geq dan SIM = y, menampilkan pesan boleh membawa motor.
 - 5.2. Jika umur \geq dan SIM \neq y, menampilkan pesan tidak boleh membawa motor.
 - 5.3. Jika umur $<$ dan SIM = y, menampilkan pesan belum cukup umur untuk membawa motor.
 - 5.4. Jika umur $<$ dan SIM \neq y, menampilkan pesan belum cukup umur punya SIM.
6. Menutup objek *Scanner*.
7. Menjaankan program dan menguji dengan berbagai kombinasi *input* umur dan SIM.

Dengan *multi-if*, program bisa memeriksa beberapa kondisi *independent* sekaligus. Namun, kelemahannya adalah ada kemungkinan lebih dari satu *if* dijalankan jika kondisi tidak saling eksklusif.

2.4. Percabangan If-Else If

Struktur *if-else if* digunakan untuk kondisi yang lebih dari dua pilihan. Berbeda dengan *multi-if*, hanya satu kondisi yang dijalankan karena dievaluasi secara berurut dari atas ke bawah.

```

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
24 }

```

Gambar 2.4. Program Java dengan percabangan *if-else if*.

Program meminta input berupa nilai angka, kemudian menentukan huruf grade sesuai rentang nilai.

Langkah kerja:

1. Membuat *class* baru dengan nama Nilai.
2. Mendeklarasikan variabel nilai bertipe *int*.
3. Menggunakan objek *Scanner* untuk menerima *input* nilai angka dari pengguna.
4. Memasukkan kondisi percabangan *if-else if* untuk menentukan nilai huruf berdasarkan rentang:
 - a. Nilai $\geq 81 \rightarrow A$
 - b. Nilai $\geq 70 \rightarrow B$
 - c. Nilai $\geq 60 \rightarrow C$
 - d. Nilai $\geq 50 \rightarrow D$

- e. Selain itu → E
- 5. Menutup objek *Scanner*.
- 6. Menjalankan program dan menguji dengan beberapa nilai angka berbeda.

if-else if sangat cocok untuk klarifikasi, seperti penilaian. Program hanya akan menampilkan satu hasil akhir meskipun ada banyak kondisi.

2.5. Percabangan Switch-Case.

Switch-case digunakan untuk memilih satu dari banyak kemungkinan berdasarkan nilai variabel tunggal. Struktur ini lebih rapi dibandingkan menggunakan banyak *if-else if*.

```
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 }
51
52 }
53

```

Gambar 2.5. Program Java dengan percabangan *Switch-Case*.

Program meminta *input* angka bulan (1-12) dan mencetak nama bulan yang sesuai.

Langkah kerja:

1. Membuat *class* baru dengan nama NamaBulan.
2. Mendeklarasikan variabel bulan bertipe *int*.
3. Membuat objek *Scanner* untuk meminta *input* angka dari pengguna.
4. Menggunakan struktur *switch-case* untuk memetakan angka ke nama bulan.
5. Menambahkan *break* di setiap case untuk mencegah eksekusi berlanjut ke case berikutnya.
6. Menyediakan *default* untuk menangani *input* yang tidak valid.
7. Menutup objek *Scanner*.

8. Menjalankan program dan menguji dengan berbagai *input* angka 1-12 sertap input di luar *range*.

Switch-case lebih ringkas daripada *if-else if* Ketika kondisi hanya membandingkan satu variabel. Pada program ini, input angka 1-12 dapat langsung dipetakan ke nama bulan.

BAB III

KESIMPULAN

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa struktur percabangan merupakan salah satu konsep dasar yang sangat penting dalam pemrograman Java. Melalui percabangan *if*, *if-else*, *if-else if*, dan *switch-case*, program dapat mengambil keputusan secara otomatis sesuai dengan kondisi yang diberikan. Hal ini membuat program menjadi lebih dinamis dan mampu menyesuaikan hasil *output* berdasarkan *input* dari pengguna.

Selain itu, masing-masing bentuk percabangan memiliki fungsi yang berbeda sesuai kebutuhan. Struktur *if* digunakan untuk kondisi sederhana, *if-else* untuk dua kemungkinan, *if-else if* untuk banyak kondisi berurutan, *multi-if* untuk kondisi independent, dan *switch-case* lebih efisien untuk pilihan yang jumlahnya banyak tetapi berasal dari satu variabel. Dengan memahami dan mengimplementasikan percabangan, mahasiswa dapat melatih logika pemrograman serta membangun program yang lebih fleksibel dan bermanfaat dalam menyelesaikan berbagai permasalahan nyata.

DAFTAR PUSTAKA

- [1] Oracle, "The Java Tutorials," Oracle Java Documentation, 25 october 2024.
[Online]. Available:
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>.
- [2] A. Peev, "Java Basics Tutorial – Part 6 – Advanced Conditional Statements," SoftUni Global, 22 october 2021. [Online]. Available: https://softuni.org/code-lessons/java-basics-tutorial-part-6-advanced-conditional-statements/?utm_source=chatgpt.com.
- [3] M. Update, "Tutorial Java: Percabangan Switch-case," Mahasiswa update, 07 September 2023. [Online]. Available: https://mahasiswaupdate.com/materi-kuliah/percabangan-switch-case-java/?utm_source=chatgpt.com.