

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
“STRUKTUR KONTROL PERCABANGAN”

DISUSUN OLEH:

ABDUL JABBAR

2511533021

DOSEN PENGAMPU:

DR. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

JOVANTRI IMMANUEL GULO



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Allah SWT karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan penulisan materi mengenai struktur kontrol percabangan dalam bahasa pemrograman Java. Penulisan ini bertujuan untuk memberikan pemahaman kepada pembaca tentang konsep dasar, fungsi, serta penerapan struktur kontrol percabangan atau pernyataan kondisi yang menjadi salah satu elemen penting dalam logika pemrograman.

struktur kontrol percabangan digunakan untuk mengatur alur eksekusi program berdasarkan kondisi tertentu, seperti penggunaan if, if-else, if-else-if, dan switch-case. Dengan adanya pernyataan kondisi ini, sebuah program mampu mengambil keputusan dan mengeksekusi blok kode tertentu sesuai syarat yang diberikan. Hal ini menjadikan program lebih dinamis, interaktif, dan sesuai dengan kebutuhan logika yang diinginkan.

Penulis menyadari bahwa penjelasan yang disajikan masih jauh dari sempurna. Oleh karena itu, kritik dan saran dari pembaca sangat diharapkan demi perbaikan dan penyempurnaan di masa mendatang. Semoga tulisan ini dapat memberikan manfaat serta menjadi bahan referensi bagi para pembaca yang sedang mempelajari dasar-dasar pemrograman Java.

Padang, 30 september 2025

Abdul jabbar

DAFTAR ISI

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN	1
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 TUJUAN	1
1.3 Manfaat.....	2
BAB II PEMBAHASAN.....	3
2.1 Struktur Kontrol Percabangan.....	3
2.2 Jenis Struktur Kontrol Percabangan	3
2.2.1 Statement If	3
2.2.2 Statement if else	4
2.2.3 Statement if else if	4
2.2.4 Statement Switch Case.....	4
2.3 Kode Program Percabangan IF	4
2.3.1 Kode Program Percabangan IF.....	5
2.3.2 Percabangan If-Else	7
2.3.3 Percabangan if Else-If.....	7
2.3.4 Percabangan Switch-Case.....	8
BAB III PENUTUP	10
3.1 Kesimpulan	10
3.2 Saran	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, logika pengambilan keputusan merupakan salah satu konsep fundamental yang harus dikuasai oleh setiap programmer. Bahasa pemrograman, termasuk Java, menyediakan cara untuk membuat program dapat memilih jalannya sendiri sesuai kondisi yang terjadi. Salah satu cara yang digunakan adalah melalui statement conditional atau pernyataan kondisi.

Statement conditional berperan penting dalam menentukan alur eksekusi sebuah program. Tanpa adanya pernyataan kondisi, program hanya akan berjalan secara linear dari atas ke bawah tanpa mempertimbangkan keadaan tertentu. Dengan adanya conditional statement, program dapat menguji sebuah syarat, kemudian mengeksekusi perintah yang sesuai dengan hasil evaluasi kondisi tersebut.

Di Java, statement conditional terdiri dari beberapa bentuk utama seperti if, if-else, if-else-if, dan switch-case. Setiap bentuk memiliki karakteristik dan kegunaannya masing-masing, mulai dari pengambilan keputusan sederhana hingga penanganan kondisi yang lebih kompleks. Penguasaan terhadap konsep ini sangat penting karena hampir seluruh program modern memanfaatkan logika kondisi, baik dalam aplikasi sederhana maupun sistem berskala besar.

Berdasarkan hal tersebut, pembahasan mengenai statement conditional di Java perlu diberikan secara komprehensif agar pembelajar dapat memahami fungsinya, perbedaannya, serta penerapannya dalam membangun sebuah program yang lebih dinamis dan fleksibel.

1.2 TUJUAN

1. Untuk menjelaskan pengertian dan konsep dasar statement conditional dalam bahasa pemrograman Java.
2. Untuk memahami jenis-jenis statement conditional seperti if, if-else, if-else-if, dan switch-case beserta cara penggunaannya.
3. Untuk memberikan contoh penerapan statement conditional dalam program sederhana agar mempermudah pemahaman.
4. Untuk melatih kemampuan berpikir logis dalam menyusun alur keputusan di dalam program.
5. Untuk mempersiapkan pembaca dalam mengembangkan program Java yang lebih kompleks, dinamis, dan sesuai kebutuhan.

1.3 Manfaat

1. Membantu pembaca memahami pentingnya logika pengambilan keputusan dalam pemrograman.
2. Memberikan dasar pengetahuan yang kuat untuk mempelajari topik pemrograman yang lebih lanjut, seperti perulangan, array, maupun struktur data.
3. Melatih kemampuan analisis dan pemecahan masalah melalui penerapan kondisi dalam program.
4. Membantu mahasiswa atau pembelajar pemula untuk menyusun kode program yang lebih efisien, terstruktur, dan mudah dipahami.
5. Menjadi bekal praktis dalam mengembangkan aplikasi nyata yang membutuhkan pengambilan keputusan berdasarkan kondisi tertentu

BAB II PEMBAHASAN

2.1 Struktur Kontrol Percabangan

Kondisi dan pernyataan if memungkinkan untuk mengendalikan aliran program - memutuskan kode mana yang dijalankan, dan kode mana yang dilewati. Bayangkan seperti di dunia nyata: Kalau hujan, bawa payung. Kalau tidak, jangan lakukan apa pun. Setiap if pernyataan membutuhkan kondisi yang menghasilkan true atau false.

2.2 Jenis Struktur Kontrol Percabangan

2.2.1 Statement If

Pernyataan if akan menentukan sebuah pernyataan (atau blok kode) yang akan eksekusi jika dan hanya jika persyaratan bernilai benar (true)

Bentuk dari pernyataan if:

```
If ( boolean_expression )  
statement;  
  
atau  
  
if ( boolean_expression ) {  
statement 1;  
statement2;  
...  
}
```

2.2.2 Statement if else

if-else adalah salah satu bentuk conditional statement di Java yang digunakan untuk memilih dua jalur eksekusi. Jika kondisi bernilai true, maka program akan menjalankan blok kode pada bagian if. Jika kondisi bernilai false, maka program akan menjalankan blok kode pada bagian else.

2.2.3 Statement if else if

if-else-if adalah bentuk conditional statement yang digunakan ketika ada lebih dari dua kondisi yang ingin diuji. Struktur ini memungkinkan program memilih satu di antara beberapa jalur eksekusi, tergantung pada kondisi yang bernilai true terlebih dahulu.

2.2.4 Statement Switch Case

switch-case adalah salah satu bentuk conditional statement di Java yang digunakan untuk memilih satu jalur eksekusi dari banyak kemungkinan berdasarkan nilai variabel atau ekspresi.

Jika nilai variabel sesuai dengan salah satu case, maka blok kode di dalam case tersebut akan dijalankan. Jika tidak ada case yang cocok, maka blok default akan dieksekusi (jika ada).

2.3 Kode Program Percabangan IF

Kode program pertama Adalah statement If berikut contoh kode program untuk syarat kelulusan dengan minimal 2.75

2.3.1 Kode Program Percabangan IF

```
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.print("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 }
```

Kode program 2.1

Langkah pengerjaan kode program sebagai berikut:

1. Membuat package

package pekan4;

Program diletakkan dalam package bernama pekan4 agar lebih terorganisir.

2. Mengimpor library Scanner

import java.util.Scanner;

Library Scanner digunakan untuk membaca input dari pengguna melalui keyboard.

3. Membuat kelas utama

public class latIf1 {Nama kelas adalah latIf1, sesuai dengan nama file latIf1.java.

4. Membuat method utama main

public static void main(String[] args) {

Method main adalah titik awal eksekusi program di Java.

5. Mendeklarasikan variabel IPK

double IPK;

Variabel IPK bertipe double untuk menyimpan nilai Indeks Prestasi Kumulatif.

6. Membuat objek Scanner

Scanner input = new Scanner(System.in);

Objek input digunakan untuk membaca data yang dimasukkan oleh pengguna.

7. Menampilkan perintah input ke layar

```
System.out.print("input IPK Anda = ");
```

Program meminta pengguna memasukkan nilai IPK.

8. Membaca input pengguna

```
IPK = input.nextDouble();
```

Nilai IPK yang diketik pengguna dibaca dan disimpan ke variabel IPK.

9. Menutup Scanner

```
input.close();
```

Menutup objek Scanner setelah selesai digunakan untuk menghemat resource.

10. Membuat kondisi dengan if

```
if (IPK > 2.75) { System.out.println("Anda lulus sangat Memuaskan dengan  
IPK " + IPK);}
```

Program mengecek apakah nilai IPK lebih dari 2.75.

Jika benar → tampil pesan “Anda lulus sangat Memuaskan dengan IPK ...”.

Jika salah → program tidak menampilkan apa-apa (karena tidak ada else).

```
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 Bawa Motor ");
22        }
23        if ((umur < 17) && (sim!='Y')) {
24            System.out.println("Anda belum cukup umur punya SIM ");
25        }
26    }
27 }
```

Kode program 2.2

Program ini meminta input umur dan status SIM C, lalu menampilkan pesan sesuai kondisi. Jika umur ≥ 17 dan punya SIM, boleh bawa motor. Jika umur ≥ 17 tapi tidak punya SIM, tidak boleh bawa motor. Jika umur < 17 dan punya SIM, dianggap belum cukup umur punya SIM. Jika umur < 17 dan tidak punya SIM,

belum boleh bawa motor. Untuk langkah pengerjaan kode program multi if sama dengan pengerjaan kode program if.

2.3.2 Percabangan If-Else

bentuk conditional statement di Java yang digunakan untuk memilih dua jalur eksekusi. Jika kondisi bernilai true, maka program akan menjalankan blok kode pada bagian if. Jika kondisi bernilai false, maka program akan menjalankan blok kode pada bagian 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 }
```

Kode Program 2.3

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat scanner input supaya bisa input data dari keyboard serta tidak lupa untuk menutup scanner supaya program tidak error
3. Membuat percabangan if-else dan hasil dari if jika pernyataan bernilai benar begitu pula pernyataan ketika bernilai salah
4. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

2.3.3 Percabangan if Else-If

bentuk conditional statement yang digunakan ketika ada lebih dari dua kondisi yang ingin diuji

```
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("input 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("D");
24        }
25    }
26 }
```

Kode program 2.4

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat scanner input supaya bisa input data dari keyboard serta tidak lupa untuk menutup scanner supaya program tidak error
3. Membuat percabangan else-if dan hasil dari if jika pernyataan bernilai benar begitu pula pernyataan ketika bernilai salah
4. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat

2.3.4 Percabangan Switch-Case

yang digunakan untuk memilih satu jalur eksekusi dari banyak kemungkinan berdasarkan nilai variabel atau ekspresi.

```

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

```

Kode Program 5

Langkah pengerjaan kode program sebagai berikut:

1. Buatlah sebuah *package* dan *class* baru untuk memulai membuat program
2. Buat scanner input supaya bisa input data dari keyboard serta tidak lupa untuk menutup scanner supaya program tidak error
3. Membuat percabangan switch-case
4. Coba jalankan program dan cek Kembali jika ada kesalahan pada kode program yang dibuat
5. Setiap case diakhiri dengan break agar program keluar dari switch setelah menemukan bulan yang sesuai.
6. Scanner ditutup setelah selesai membaca input.

BAB III PENUTUP

3.1 Kesimpulan

Struktur kontrol percabangan merupakan salah satu komponen penting dalam bahasa pemrograman Java yang berfungsi untuk mengatur alur jalannya program berdasarkan kondisi tertentu. Dengan adanya percabangan, program dapat mengambil keputusan dan memilih jalur eksekusi yang sesuai, sehingga tidak hanya berjalan secara linear dari atas ke bawah. Dalam Java, percabangan dapat dibuat menggunakan beberapa bentuk, yaitu if, if-else, if-else-if, dan switch-case. Pernyataan if digunakan untuk mengeksekusi kode jika kondisi bernilai benar, if-else memberikan dua pilihan eksekusi antara kondisi benar atau salah, sedangkan if-else-if memungkinkan pengecekan beberapa kondisi secara berurutan. Sementara itu, switch-case lebih cocok digunakan untuk percabangan dengan nilai yang pasti seperti angka, karakter, atau string. Dengan memahami dan menguasai struktur kontrol percabangan, seorang programmer dapat membuat program yang lebih dinamis, fleksibel, serta mampu menyelesaikan berbagai permasalahan logika dalam pemrograman.

3.2 Saran

Dalam mempelajari struktur kontrol percabangan di Java, sebaiknya mahasiswa atau pembelajar pemula tidak hanya memahami teori, tetapi juga langsung berlatih dengan membuat program sederhana agar logika percabangan lebih mudah dipahami. Penggunaan variasi percabangan seperti if, if-else, if-else-if, maupun switch-case perlu dikuasai dengan baik, karena setiap bentuk memiliki fungsi dan kelebihan masing-masing. Selain itu, pembelajar juga disarankan untuk mencoba mengembangkan contoh-contoh program nyata, misalnya sistem penilaian, kategori umur, atau menu pilihan, sehingga konsep percabangan lebih aplikatif dan relevan dengan kehidupan sehari-hari. Dengan berlatih secara konsisten, pemahaman mengenai percabangan akan semakin kuat dan menjadi bekal penting dalam mempelajari materi pemrograman yang lebih lanjut.

DAFTAR PUSTAKA

- [1] *Scribd Struktur Kontrol Percabangan* , Tersedia pada:
<https://id.scribd.com/document/406896698/3-Struktur-Kontrol-Percabangan-docx>
di Akses pada [1-Oktober-2025]
- [2] *Petani Kode Memahami Bentuk Percabangan dalam Java* , Tersedia pada:
<https://www.petanikode.com/java-percabangan/>
di Akses pada [1-Oktober-2025]
- [3] *W3Schools Java Conditions and If Statements* Tersedia pada:
https://www.w3schools.com/java/java_conditions.asp
di Akses pada [1-Oktober-2025]