

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

STATEMENT CONDITIONAL

PEMROGRAMAN JAVA

disusun Oleh:

Nama: Amiratul Fadhilah

NIM: 2511532023

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

Asisten Praktikum: Jovantri Immanuel Gulo



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Puji syukur ke hadirat Allah Yang Maha Esa atas rahmat dan karunia-Nya, sehingga laporan praktikum dengan judul “*Statement Conditional*” ini dapat diselesaikan dengan baik dan tepat waktu.

Laporan ini disusun sebagai bagian dari kegiatan praktikum Algoritma dan Pemrograman yang bertujuan untuk memperdalam pemahaman mahasiswa mengenai konsep dasar kontrol alur program di Java. Secara khusus, laporan ini akan membahas implementasi *statement conditional* seperti *if*, *if-else*, *if-else-if*, dan *switch-case* yang menjadi fondasi dalam logika pengambilan keputusan pada sebuah program.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat serta pemahaman yang lebih mendalam bagi pembaca.

Padang, 01 Oktober 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	i
---------------------	---

DAFTAR ISI.....	ii
-----------------	----

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	1

BAB II PEMBAHASAN

2.1 Dasar Teori.....	2
2.2 Jenis-Jenis <i>Statement Conditional</i>	2
2.3 Kode Program dan Pembahasan	3
2.4 Langkah-Langkah Praktikum.....	8

BAB III KESIMPULAN

3.1 Kesimpulan	9
3.2 Saran.....	9
3.3 DAFTAR PUSTAKA.....	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Bahasa pemrograman Java merupakan salah satu bahasa populer yang digunakan untuk mengembangkan aplikasi berbasis *desktop*, *web*, maupun *mobile*. Dalam proses pemrograman, pengambilan keputusan menjadi bagian penting agar program dapat berjalan sesuai kondisi tertentu. Hal ini diwujudkan melalui penggunaan *statement conditional* seperti *if*, *if-else*, *if-else-if*, maupun *switch-case*.

Dengan adanya *statement conditional*, alur logika program dapat diatur sehingga program mampu merespons berbagai input atau keadaan. Oleh karena itu, pemahaman konsep ini menjadi dasar penting bagi mahasiswa dalam mempelajari algoritma dan pemrograman.

1.2 Tujuan Praktikum

Berdasarkan latar belakang di atas, tujuan dari pelaksanaan praktikum ini adalah sebagai berikut:

1. Memahami konsep dasar dan fungsi dari *statement conditional* pada bahasa pemrogramanp Java.
2. Melatih kemampuan dalam menulis kode program yang melibatkan pengambilan keputusan.
3. Meningkatkan keterampilan dalam menganalisis hasil eksekusi program berdasarkan kondisi yang diberikan.
4. Membiasakan diri dalam menyusun dokumentasi dan laporan hasil praktikum secara sistematis dan terstruktur.

1.3 Manfaat Praktikum

1. Meningkatkan pemahaman mahasiswa terhadap logika pemrograman.
2. Membiasakan mahasiswa berpikir secara sistematis dalam menyusun algoritma.
3. Memberikan pengalaman langsung dalam merancang alur program yang efisien dan sesuai dengan kebutuhan.

BAB II

PEMBAHASAN

2.1 Dasar Teori

Secara umum, *statement conditional* adalah perintah yang memungkinkan program memilih satu atau beberapa jalur eksekusi berdasarkan kondisi logis yang diberikan. Dengan adanya *statement conditional*, program tidak hanya sekadar mengeksekusi instruksi, tetapi juga mampu “memutuskan” tindakan apa yang harus dilakukan sesuai dengan nilai *input*, variabel, atau ekspresi tertentu.

Dalam bahasa Java, *statement conditional* digunakan untuk mengontrol aliran (*flow control*) sehingga lebih responsif, fleksibel, dan adaptif terhadap berbagai kemungkinan yang muncul. Konsep ini sangat penting karena hampir semua aplikasi membutuhkan pengambilan keputusan, mulai dari validasi data pengguna, penentuan kategori nilai, hingga pemilihan menu atau opsi tertentu.

2.2 Jenis-Jenis *Statement Conditional*

1. **Statement if:**
Merupakan bentuk paling dasar dari percabangan. Blok kode di dalam *if* hanya akan dieksekusi jika kondisi yang diberikan bernilai benar (*true*).
2. **Statement if-else:**
Digunakan untuk menangani dua kemungkinan alur eksekusi. Jika kondisi pada *if* bernilai benar (*true*), blok *if* dieksekusi. Jika salah (*false*), maka blok *else* yang akan dieksekusi.
3. **Statement if-else-if:**
Merupakan struktur percabangan untuk menangani lebih dari dua kondisi yang tidak bisa terjadi bersamaan (eksklusif). Program akan memeriksa kondisi dari atas ke bawah dan hanya akan mengeksekusi blok kode dari kondisi pertama yang bernilai *true*. Jika tidak ada kondisi yang terpenuhi, blok *else* terakhir akan dieksekusi.
4. **Pola multiple if:**
Pola ini menggunakan serangkaian *statement if* tunggal. Berbeda dengan *if-else-if*, setiap kondisi *if* dalam pola ini akan selalu dievaluasi tanpa

memedulkan apakah kondisi *if* sebelumnya bernilai benar atau salah. Pola ini digunakan ketika beberapa kondisi yang tidak saling eksklusif mungkin perlu ditangani secara bersamaan. Namun, jika digunakan untuk kondisi yang sebenarnya eksklusif, pola ini menjadi kurang efisien.

5. Statement switch-case:

Merupakan alternatif yang lebih sederhana dari *if-else-if* untuk kondisi bercabang banyak karena lebih efisien dan mudah dibaca. Biasanya digunakan untuk memeriksa kesamaan nilai dari sebuah variabel dengan beberapa nilai konstan (*case*) yang telah ditentukan.

2.3 Kode Program dan Pembahasan

a. latIf1.java (if)

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

Kode Program 2.1

Contoh *Output*:

Input IPK Anda = 3.5

Anda Lulus Sangat Memuaskan dengan IPK 3.5

Input IPK Anda = 2.5

(Tidak ada *output*)

Pada program ini digunakan struktur dasar *if*. Program meminta masukan berupa IPK, kemudian dilakukan pengecekan apakah nilai tersebut lebih besar dari 2.75. Jika kondisi terpenuhi (*true*), maka

pesan kelulusan akan dicetak. Namun, jika kondisi tidak terpenuhi (false), program akan langsung berakhir tanpa mencetak pesan apapun karena tidak ada blok *else* yang menangani kondisi sebaliknya.

b. Ifelse1.java (if-else)

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

Kode Program 2.2

Contoh *Output*:

Input IPK Anda = 3.1

Anda Lulus Sangat Memuaskan dengan IPK 3.1

Input IPK Anda = 2.70

Anda Tidak Lulus

Program ini adalah pengembangan dari program sebelumnya dengan menambahkan blok *else*. Struktur *if-else* memastikan bahwa selalu ada satu blok kode yang dieksekusi. Jika kondisi $IPK > 2.75$ bernilai *true*, blok *if* dijalankan. Jika *false*, maka alur eksekusi akan langsung menuju blok *else* untuk mencetak pesan “Anda Tidak Lulus”.

c. Nilai.java (if-else-if)

```

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

```

Kode Program 2.3

Contoh Output:

Inputkan nilai angka= 75

B

Program ini menggunakan struktur *if-else-if* untuk mengonversi nilai angka menjadi nilai huruf. Pemeriksaan kondisi dilakukan secara berurutan. Misalnya, untuk *input* 75, kondisi pertama ($\text{nilai} \geq 81$) bernilai *false*. Program lanjut ke kondisi kedua ($\text{nilai} \geq 70$) yang bernilai *true*, sehingga mencetak “B” dan sisa struktur percabangan diabaikan. Blok *else* terakhir berfungsi sebagai penampung untuk semua kondisi yang tidak terpenuhi di atasnya.

- d. multiIf.java (multiple if)


```

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 bawa 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 buat sim");
26         }
27     }
28 }

```

Kode Program 2.4

Contoh *Output*:

Input umur anda: 20

Apakah Anda Sudah Punya Sim C (y/n): y

Anda sudah dewasa dan boleh bawa motor

Program ini menggunakan beberapa *if* terpisah untuk mengecek kombinasi kondisi umur dan kepemilikan SIM. Meskipun fungsional, pendekatan ini kurang efisien karena setiap kondisi *if* akan selalu diperiksa, bahkan setelah salah satu kondisi sudah terpenuhi. Untuk hal seperti ini, penggunaan struktur *if-else-if* akan lebih optimal karena hanya satu blok kode yang relevan yang akan dieksekusi.

- e. NamaBulan.java (switch-case)

```

1  package pekan4;
2
3  import java.util.Scanner;
4
5  public class MamaBulan {
6
7      public static void main (String[] args) {
8          Scanner scanner = new Scanner(System.in);
9          System.out.print("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;
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         }
49         scanner.close();
50     }
51 }
52

```

Kode Program 2.5

Contoh Output:

Masukkan Angka Bulan (1 - 12): 9

September

Program ini menunjukkan implementasi *switch-case* yang efektif. Program membandingkan nilai variabel bulan dengan setiap *case*. Ketika nilai yang cocok ditemukan (misalnya 9), kode di dalam *case* tersebut dieksekusi. Pernyataan *break* sangat krusial untuk menghentikan eksekusi dan keluar dari blok *switch*. Tanpa *break*, program akan melanjutkan eksekusi ke *case* berikutnya, yang dalam

konteks ini akan menghasilkan *output* yang salah. Struktur ini lebih rapi dan seringkali lebih cepat dibandingkan menggunakan *if-else-if*.

2.4 Langkah-Langkah Praktikum

Proses pelaksanaan praktikum ini mengikuti langkah-langkah sistematis sebagai berikut:

1. Menyiapkan perangkat dengan *Java Development Kit* (JDK) dan *Integrated Development Environment* (IDE) seperti Eclipse telah terpasang.
2. Membuat *package* baru dengan nama *pekan4*.
3. Membuat beberapa *class* Java baru sesuai dengan nama file program yang akan diuji (*latIf1.java*, *Ifelse1.java*, *Nilai.java*, *multiIf.java*, *NamaBulan.java*).
4. Menuliskan kode program dengan format dan sintaksis yang benar.
5. Melakukan kompilasi (*compile*) dan eksekusi (*run*) pada setiap program secara bergantian.
6. Menganalisis *output* program berdasarkan *input* atau kondisi yang diberikan.
7. Menganalisis alur eksekusi setiap program untuk memahami bagaimana *statement conditional* yang digunakan mempengaruhi *output* yang dihasilkan.
8. Menyusun laporan praktikum.

BAB III

PENUTUP

3.1 Kesimpulan

Dari praktikum yang telah dilaksanakan, dapat disimpulkan bahwa:

1. *Statement conditional* merupakan intruksi penting dalam pemrograman untuk mengatur alur eksekusi berdasarkan kondisi tertentu.
2. Java menyediakan beberapa bentuk *conditional statement* yang memiliki fungsi berbeda sesuai kebutuhan logika program.
 - Struktur *if* digunakan untuk eksekusi tunggal bersyarat, *if-else* untuk dua alur alternatif, dan *if-else-if* untuk beberapa kondisi eksklusif yang berurutan.
 - Struktur *switch-case* merupakan alternatif yang efisien dan mudah dibaca untuk percabangan *if-else-if* yang panjang, karena keduanya digunakan untuk kondisi yang eksklusif.
3. Pemilihan struktur percabangan yang tepat bergantung pada kompleksitas masalah dan jumlah kondisi yang harus ditangani.
4. Implementasi langsung melalui kode program membantu mahasiswa memahami perbedaan serta keunggulan masing-masing bentuk *statement conditional*.

3.2 Saran

Disarankan mahasiswa untuk:

- Melatih penggunaan *statement conditional* dengan kasus yang lebih kompleks, misalnya dengan mengombinasikan logika AND, OR, maupun nested *if*
- Mengeksplorasi penggunaan operator *ternary* (*? :*) sebagai bentuk ringkas dari *if-else*
- Menggabungkan penggunaan *statement conditional* dengan struktur perulangan (*looping*) untuk kasus seperti validasi *input* pengguna.

DAFTAR PUSTAKA

- [1] S. Sufyan, "Conditional Statement Else If If di Java," *Sufyan97.com*, 2025. [Daring]. Tersedia pada: <https://www.sufyan97.com/2025/03/conditional-statement-el-if-if-di-java.html>. [Diakses: 30-Sep-2025].
- [2] A. R. Huda, "Laporan Hasil Alpro II Modul 1," *Studocu*, 2023. [Daring]. Tersedia pada: <https://www.studocu.id/id/document/universitas-palangka-raya/algoritma-dan-pemrograman-i/laporan-hasil-alpro-ii-modul-1/73538878>. [Diakses: 30-Sep-2025].
- [3] Oracle, "The Java™ Tutorials: The switch Statement," *Oracle Docs*, 2024. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/controlflow/switch.html>. [Diakses: 30-Sep-2025].
- [4] R. Hidayat, "Laporan Praktikum Algoritma dan Pemrograman," *Scribd*, 2025. [Daring]. Tersedia pada: <https://www.scribd.com/document/725244343/PEKAN-2-LAPORAN-PRAKTIKUM-ALGORITMA-PEMROGRAMAN-RAHMATHIDAYAT>. [Diakses: 01-Okt-2025].
- [5] Tim Penyusun, *Pedoman Penulisan Laporan Praktikum Java*. Padang: Departemen Informatika, Universitas Andalas, 2025.