

**LAPORAN PRAKTIKUM ALGORITMA DAN PEMOGRAMAN**

**“CONDITIONAL STATEMENT”**

**DISUSUN OLEH:**

**MUHAMMAD FATHAN EDLIN**

**2511537001**

**DOSEN PENGAMPU:**

**Dr. WAHYUDI, S.T,M.T**

**ASISTEN PRAKTIKUM:**

**JOVANTRI IMMANUEL GELO**



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena atas rahmat, karunia, serta hidayah-Nya, penulis dapat menyelesaikan laporan praktikum Algoritma dan Pemrograman dengan topik “Conditional Statement” tepat pada waktunya. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dan dokumentasi kegiatan praktikum sekaligus sarana untuk memperdalam pemahaman mengenai konsep percabangan dalam pemrograman.

Materi Conditional Statement merupakan salah satu bagian penting dalam pembelajaran algoritma dan pemrograman, karena memungkinkan program untuk mengambil keputusan berdasarkan kondisi tertentu. Dengan adanya struktur percabangan seperti *if*, *if-else*, *if-else-if*, maupun *switch-case*, sebuah program dapat dijalankan secara lebih fleksibel, dinamis, serta sesuai dengan kebutuhan pengguna. Melalui praktikum ini, mahasiswa diharapkan tidak hanya memahami teori, tetapi juga mampu menerapkannya secara langsung ke dalam kode program yang efisien dan tepat.

Dalam penyusunan laporan ini, penulis menyadari bahwa masih terdapat keterbatasan baik dari segi isi maupun penjelasan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun agar laporan ini dapat lebih baik di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah Algoritma dan Pemrograman, asisten praktikum, serta semua pihak yang telah memberikan bimbingan, arahan, dan dukungan hingga terselesaikannya laporan ini. Semoga laporan ini dapat memberikan manfaat, khususnya bagi penulis sendiri dan umumnya bagi para pembaca yang ingin memahami lebih dalam mengenai Conditional Statement dalam pemrograman.

Padang, 30 September 2025

Muhammad Fathan Edlin

## DAFTAR ISI

KATA PENGANTAR.....	
DAFTAR ISI.....	
BAB I.....	
PENDAHULUAN.....	
1.1 Latar Belakang.....	
1.2 Tujuan.....	
1.3 Manfaat.....	
BAB II.....	
PEMBAHASAN .....	
2.1 Dasar Teori.....	
2.2 Jenis – Jenis Conditional Statement.....	
2.3 Penerapan Conditional Statement.....	
2.4 Analisis Hasil Praktikum.....	
BAB III.....	
KESIMPULAN DAN SARAN.....	
3.1 Kesimpulan.....	
3.2 Saran.....	
DAFTAR PUSTAKA.....	

## **BAB I**

### **1.1 Latar Belakang**

Dalam dunia pemrograman, salah satu konsep fundamental yang harus dipahami oleh setiap programmer pemula adalah conditional statement atau pernyataan kondisi. Conditional statement digunakan untuk mengatur alur logika program berdasarkan kondisi tertentu, sehingga program dapat merespons dengan cara yang berbeda sesuai dengan input maupun keadaan yang terjadi.

Tanpa adanya conditional statement, program hanya akan berjalan secara linear dari atas ke bawah tanpa adanya percabangan keputusan. Hal ini tentu membatasi fleksibilitas dan fungsionalitas program. Oleh karena itu, conditional statement sangat penting agar program dapat “berpikir” layaknya manusia yang harus memilih suatu tindakan berdasarkan situasi.

Dalam mata kuliah Algoritma dan Pemrograman, pemahaman mengenai conditional statement menjadi dasar yang harus dikuasai sebelum melangkah ke materi yang lebih kompleks, seperti perulangan, array, maupun struktur data. Dengan mempelajari conditional statement, mahasiswa dapat mengembangkan kemampuan logika, memahami alur percabangan, serta membangun pondasi kuat dalam pembuatan algoritma dan implementasi kode program.

Melalui praktikum ini, mahasiswa diharapkan mampu mengaplikasikan teori conditional statement ke dalam bentuk program nyata, menganalisis alur logika percabangan, serta menyelesaikan permasalahan sederhana menggunakan konsep *if*, *if-else*, *nested if*, maupun *switch-case*. Hal ini diharapkan menjadi bekal penting dalam penguasaan keterampilan pemrograman yang lebih lanjut.

## **1.2 Tujuan**

1. Memahami konsep dasar conditional statement dalam pemrograman.
2. Mampu mengimplementasikan conditional statement pada kode program menggunakan berbagai bentuk percabangan (*if, if-else, nested if, switch-case*).
3. Melatih kemampuan logika dan analisis dalam menentukan alur eksekusi program berdasarkan kondisi tertentu.
4. Menyelesaikan permasalahan sederhana menggunakan algoritma dengan conditional statement.

## **1.3 Manfaat**

1. Memberikan pemahaman yang lebih baik tentang bagaimana program bekerja dengan logika percabangan.
2. Meningkatkan keterampilan mahasiswa dalam menulis kode program yang efisien, dinamis, dan sesuai kebutuhan.
3. Menjadi dasar untuk memahami materi lanjutan dalam pemrograman, seperti perulangan, array, maupun pemrograman berbasis objek.
4. Membekali mahasiswa dengan kemampuan praktis dalam menyelesaikan masalah nyata melalui pemrograman.

## BAB II

### PEMBAHASAN

#### 2.1 Dasar Teori

Dalam pemrograman, conditional statement atau pernyataan kondisi merupakan instruksi yang digunakan untuk mengambil keputusan berdasarkan suatu kondisi logika. Pernyataan kondisi ini memungkinkan sebuah program untuk mengeksekusi bagian kode tertentu hanya jika syarat yang ditentukan bernilai benar (*true*).

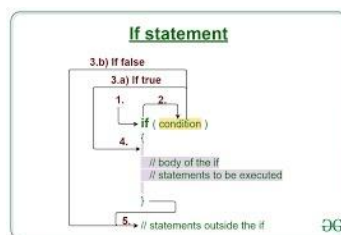
Conditional statement menjadi elemen penting dalam algoritma karena hampir semua permasalahan membutuhkan pengambilan keputusan. Misalnya, menentukan apakah seorang mahasiswa lulus berdasarkan nilai, atau menampilkan diskon jika pembelian melebihi jumlah tertentu.

Secara umum, conditional statement menggunakan operator relasional (>, <, >=, <=, ==, !=) dan operator logika (&&, ||, !) untuk membandingkan nilai dan menentukan kondisi yang akan dijalankan.

#### 2.2 Jenis – Jenis Conditional Statement

##### 1. If Statement

Bentuk paling sederhana dari conditional statement. Instruksi dalam blok *if* hanya dijalankan jika kondisi bernilai *true*.



##### 2. If-Else Statement

Menyediakan dua jalur eksekusi: jika kondisi bernilai *true*, jalur pertama dijalankan; jika *false*, jalur kedua dijalankan.

```
nilai = 80
if nilai > 65:
    print("Lulus")
else:
    print("Tidak Lulus")
```

Lulus

### 3. If-Else If-Else Statement

Digunakan untuk mengecek beberapa kondisi secara berurutan. Jika salah satu kondisi terpenuhi, blok kode yang sesuai akan dieksekusi.

```
1 #include <iostream>
2 using namespace std ;
3
4 int main() {
5     int nomer ;
6
7     cout<<"Masukkan [1/2/3] : ";cin>>nomer;
8     if (nomer<=0){
9         cout<<"kurang dari 1"<<endl;
10    }else if(nomer==1){
11        cout<<"1"<<endl;
12    }else if(nomer==2){
13        cout<<"2"<<endl;
14    }else if(nomer==3){
15        cout<<"3"<<endl;
16    }else{
17        cout<<"lebih dari 3"<<endl;
18    }
19    return 0 ;
20 }
```

### 4. Nested If

Yaitu kondisi *if* yang berada di dalam kondisi *if* lainnya. Digunakan ketika keputusan yang diambil tergantung pada kondisi lebih spesifik.

```
Bentuk umum pendisian Nested If:
if(kondisi1)
{ statement yang dikerjakan jika kondisi1 benar }
    if (kondisi1.2)
    { statement yang dikerjakan jika kondisi1.1 benar }
else if(kondisi2)
{ statement yang dikerjakan jika kondisi2 benar }
    if(kondisi2.2)
    { statement yang dikerjakan jika kondisi2.2 benar }
else
    if(kondisi3)
    { statement yang dikerjakan jika kondisi3 salah }
```

### 5. Switch-Case Statement

Alternatif dari *if-else if* ketika terdapat banyak kondisi yang membandingkan sebuah variabel dengan nilai tetap. Struktur ini lebih ringkas dan mudah dibaca.

```
int i=2; // you take an input

switch (i) {
    case 1:
        System.out.println("One");
        break;
    case 2:
        System.out.println("Two");
        break;
    case 3:
        System.out.println("Three");
        break;
    default:
        System.out.println("no match found");
}

// the output will be Two
```

## 2.3 Penerapan Conditional Statement

Dalam praktikum ini, mahasiswa diminta untuk membuat program sederhana yang menggunakan conditional statement. Contoh kasus yang diujikan adalah:

- Menentukan apakah sebuah bilangan ganjil atau genap.
- Menentukan nilai huruf (A, B, C, D) dari input nilai angka.
- Menampilkan nama hari berdasarkan input angka 1–7.

Dengan implementasi ini, mahasiswa dapat memahami bagaimana percabangan bekerja untuk menyelesaikan masalah nyata dalam pemrograman.

## 2.4 Analisis Hasil Praktikum

Berdasarkan hasil praktikum, penggunaan conditional statement terbukti memudahkan dalam pengambilan keputusan di dalam program. Dengan penerapan *if-else* maupun *switch-case*, program dapat berjalan lebih fleksibel dan sesuai dengan input yang diberikan.

Beberapa hal yang dapat disimpulkan dari hasil praktikum:

1. If-Else cocok digunakan untuk kondisi sederhana dengan dua kemungkinan.
2. If-Else If-Else lebih efektif untuk kondisi yang memiliki banyak kemungkinan dengan rentang nilai.
3. Switch-Case lebih efisien digunakan jika kondisi berbentuk pilihan tetap (*fixed values*) seperti angka hari, menu, atau pilihan tertentu.
4. Logika program harus disusun secara sistematis agar percabangan tidak tumpang tindih dan hasil sesuai dengan yang diharapkan.

Dengan memahami berbagai bentuk conditional statement, mahasiswa diharapkan mampu membangun program yang lebih kompleks dan bermanfaat dalam menyelesaikan berbagai permasalahan algoritmik.



## BAB III

### KESIMPULAN DAN SARAN

#### 3.1 Kesimpulan

Berdasarkan hasil praktikum Algoritma dan Pemrograman dengan topik Conditional Statement, dapat disimpulkan beberapa hal sebagai berikut:

1. Conditional Statement merupakan struktur dasar dalam pemrograman yang berfungsi untuk mengambil keputusan berdasarkan kondisi tertentu.
2. Bentuk conditional statement yang umum digunakan meliputi *if*, *if-else*, *if-else if-else*, *nested if*, dan *switch-case*.
3. Penggunaan conditional statement membuat program menjadi lebih fleksibel, dinamis, dan mampu menyesuaikan hasil eksekusi dengan input yang diberikan.
4. Setiap jenis conditional statement memiliki keunggulan masing-masing:
  - *If-else* efektif untuk kondisi sederhana,
  - *If-else if-else* sesuai untuk kondisi bertingkat,
  - *Switch-case* lebih efisien jika pilihan kondisi berupa nilai konstan.
5. Melalui praktikum ini, mahasiswa dapat meningkatkan pemahaman logika percabangan, keterampilan menulis kode, serta kemampuan dalam menyusun algoritma yang sistematis.

### 3.2 Saran

Agar pemahaman mengenai conditional statement semakin optimal, penulis memberikan beberapa saran sebagai berikut:

1. Mahasiswa perlu lebih banyak berlatih dengan variasi soal pemrograman yang menggunakan percabangan.
2. Dalam menulis kode, disarankan memperhatikan kerapian, indentasi, serta komentar agar program lebih mudah dibaca dan dipahami.
3. Penggunaan conditional statement sebaiknya disesuaikan dengan kasus yang dihadapi agar program tetap efisien dan tidak boros kode.
4. Praktikum selanjutnya sebaiknya dikembangkan ke kasus yang lebih kompleks dengan menggabungkan conditional statement dengan konsep lain, seperti perulangan (*looping*) dan array.
5. Asisten praktikum diharapkan terus memberikan contoh kasus nyata yang relevan dengan kebutuhan sehari-hari sehingga mahasiswa lebih mudah memahami manfaat penerapan conditional statement.

## DAFTAR PUSTAKA

- [1] Oracle, “The Java™ Tutorials – Conditional Statements,” [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/if.html>
- [2] W3Schools, “Java If...Else Statement,” [Daring]. Tersedia pada: [https://www.w3schools.com/java/java\\_conditions.asp](https://www.w3schools.com/java/java_conditions.asp)
- [3] GeeksforGeeks, “Decision Making in Java (if, if-else, switch, break, continue, jump),” [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/decision-making-javaif-else-switch-break-continue-jump/>
- [4] Tutorialspoint, “Java – Decision Making,” [Daring]. Tersedia pada: [https://www.tutorialspoint.com/java/java\\_decision\\_making.htm](https://www.tutorialspoint.com/java/java_decision_making.htm)