

**LAPORAN PRAKTIKUM ALGORITMA DAN
PEMROGRAMAN**

PEKAN 4 STATEMENT CONDISIONAL DI JAVA



Disusun Oleh :

DIKA GIOWANDA

NIM 2311533025

Dosen Pengampu :

DR. WAHYUDI, S.T, M.T

Asisten Praktikum :

JOVANTRI IMMANUEL GULO

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, SEPTEMBER 2025

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunianya, sehingga penulis dapat menyelesaikan laporan praktikum ini dengan baik. Laporan ini disusun sebagai bentuk dokumentasi dan refleksi dari kegiatan praktikum algoritma dan pemrograman yang berfokus pada statement kondisional dalam bahasa pemrograman Java.

Dalam dunia pemrograman, kemampuan sebuah program untuk mengambil keputusan berdasarkan kondisi tertentu merupakan fondasi utama dalam membangun logika yang dinamis dan responsif. Statement kondisional seperti *if*, *else*, *else if*, dan *switch* memungkinkan program untuk mengevaluasi ekspresi logika dan mengeksekusi blok kode yang sesuai-mirip dengan cara manusia membuat pilihan dalam kehidupan sehari-hari.

Melalui praktikum ini, penulis tidak hanya mempelajari sintaks dan struktur statement kondisional, tetapi juga melatih kemampuan berpikir logis, menganalisis kasus, serta menerapkan konsep dalam berbagai skenario nyata seperti penilaian nilai, klasifikasi usia, hingga sistem validasi sederhana.

Pemahaman terhadap materi ini menjadi langkah penting sebelum melangkah ke konsep pemrograman yang lebih kompleks, seperti perulangan, fungsi, maupun pemrograman berorientasi objek.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini bermanfaat bagi pembaca dan dapat menjadi referensi dalam memahami dasar-dasar logika percabangan dalam Java.

Padang, 2025

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN.....	1
1.1 Pengertian Praktikum.....	1
1.2 Tujuan Praktikum.....	1
1.3 Persyaratan Praktikum.....	1
1.4 Tempat dan Waktu Praktikum.....	2
1.5 Manfaat Praktikum.....	2
BAB 2 PEMBAHASAN PRAKTIKUM.....	3
2.1 Pengertian Statement Condisional.....	3
2.2 Pengertian If.....	3
2.3 Pengertian Else.....	4
2.4 Pengertian If Else.....	5
2.5 Pengertian Switch.....	7
2.6 Langkah Langkah Pengerjaan Praktikum Pekan 3.....	10
BAB 3 PENUTUP.....	25
3.1 Kesimpulan.....	25
3.2 Saran.....	25
DAFTAR PUSTAKA.....	27

BAB 1

PENDAHULUAN

1.1 Pengertian Praktikum

Praktikum Java adalah kegiatan pembelajaran yang dilakukan di laboratorium komputer untuk mengasah keterampilan mahasiswa dalam memahami serta menerapkan konsep pemrograman Java. Kegiatan ini tidak hanya menekankan pada penguasaan teori, tetapi juga pada latihan penyusunan kode program, pengujian, hingga analisis hasil eksekusi. Praktikum dipandang sebagai wahana latihan yang menjembatani pemahaman konseptual dengan kemampuan teknis pemrograman.

1.2 Tujuan Praktikum

Tujuan dari pelaksanaan praktikum ini antara lain sebagai berikut :

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik.
4. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
5. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.
6. Mengetahui dan mengaplikasikan if, else, if else, dan switch.

1.3 Persyaratan Praktikum

Agar praktikum berjalan lancar, mahasiswa perlu memenuhi beberapa persyaratan berikut:

1. Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.

2. Membawa perlengkapan yang diperlukan, antara lain laptop atau komputer yang sudah terpasang *Java Development Kit* (JDK) dan *Integrated Development Environment* (IDE) yang direkomendasikan.
3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

1.4 Waktu dan Tempat Praktikum

Pelaksanaan praktikum Java mengikuti kalender akademik yang berlaku pada program studi. Setiap sesi praktikum dilaksanakan sesuai jadwal yang ditentukan oleh dosen pengampu. Tempat kegiatan umumnya berlangsung di laboratorium komputer, namun pada kondisi tertentu dapat dilaksanakan secara mandiri dengan perangkat masing-masing, selama memenuhi syarat teknis yang ditetapkan.

1.5 Manfaat Praktikum

Manfaat praktikum bahasa Java meliputi kemampuan memahami dan menerapkan konsep dasar pemrograman seperti tipe data dan kontrol alur, melatih penggunaan kode yang modular dan dapat digunakan kembali, serta membangun pemahaman tentang sintaksis dan struktur *object oriented programming* (OOP) yang lebih mendalam. Selain itu, praktikum ini juga membantu membangun fondasi untuk mengembangkan aplikasi lintas platform yang lebih kompleks di berbagai bidang teknologi.

BAB 2

PEMBAHASAN PRAKTIKUM

2.1 Pengertian Statement Condisional

Statement condisional di Java adalah struktur kontrol yang memungkinkan program membuat keputusan dan menjalankan blok kode yang berbeda berdasarkan kondisi tertentu yang bernilai benar (*true*) atau salah (*false*). Pernyataan ini berfungsi untuk mengontrol alur eksekusi program, memungkinkan program untuk merespons input dan skenario yang berbeda secara dinamis. Jenis pernyataan kondisional umum di Java meliputi *if*, *else*, *else if*, dan *switch*.

2.2 Pengertian If

Pernyataan *if* pada statement kondisional Java digunakan untuk mengevaluasi suatu kondisi (ekspresi *boolean* yang bernilai benar atau salah) dan menjalankan blok kode tertentu hanya jika kondisi tersebut bernilai benar. Pernyataan ini merupakan kontrol alur fundamental yang memungkinkan program membuat keputusan dan menjalankan instruksi secara fleksibel berdasarkan kondisi yang ada.

Cara Kerja Pernyataan *if* :

Kondisi *Boolean*: Pernyataan *if* memerlukan sebuah ekspresi yang menghasilkan nilai boolean (*true* atau *false*).

Eksekusi Bersyarat:

- Jika kondisi bernilai *true*, maka blok kode di dalam kurung kurawal { } setelah *if* akan dieksekusi.
- Jika kondisi bernilai *false*, maka blok kode tersebut akan dilewati dan program akan melanjutkan eksekusi ke pernyataan berikutnya setelah blok *if*.

Sintaks Dasar:

```
if (kondisi) {  
    // Blok kode yang akan dieksekusi  
    // jika kondisi bernilai true
```

```
}
```

- *if*: Kata kunci yang memulai pernyataan kondisional.
- kondisi: Ekspresi *boolean* yang dievaluasi.
- {}: Kurung kurawal yang berisi blok kode yang akan dijalankan.

Contoh Penggunaan :

```
public class Contoh If {  
    public static void main(String[] args) {  
        int nilai = 20;  
  
        if (nilai > 15) { // Kondisi: nilai lebih besar dari 15  
            System.out.println("Nilai lebih besar dari 15!"); // Kode ini akan  
dieksekusi  
        }  
    }  
}
```

Dalam contoh ini, karena nilai (20) lebih besar dari 15, maka kondisi `nilai > 15` bernilai *true*, sehingga pesan "Nilai lebih besar dari 15!" akan dicetak ke konsol.

2.3 Pengertian Else

Pernyataan *else* dalam kondisional Java digunakan sebagai blok kode "cadangan" atau "alternatif" yang akan dijalankan jika kondisi *if* sebelumnya tidak terpenuhi atau bernilai *false*. Statement *else* selalu mengikuti statement *if* dan tidak memerlukan kondisi sendiri, karena ia bertindak sebagai *catch-all* untuk semua kasus di mana kondisi *if* tidak dipenuhi, memberikan jalur eksekusi sekunder dalam program.

Cara Kerja:

- Kondisi *if* dievaluasi: Program akan memeriksa kondisi yang ditentukan dalam pernyataan *if*.
- Jika kondisi *true* : Blok kode di dalam pernyataan *if* akan dieksekusi.

- Jika kondisi *false* : Program akan melewati blok *if* dan langsung mengeksekusi blok kode di dalam pernyataan *else*.

Sintaks Dasar:

```
if (kondisi_true) {
    // Kode yang dijalankan jika kondisi_true adalah benar
} else {
    // Kode yang dijalankan jika kondisi_true adalah salah (false)
}
```

Contoh penggunaan:

Misalkan Anda ingin memeriksa apakah suatu angka positif atau negatif.

```
int angka = -5;
```

```
if (angka > 0) {
    System.out.println("Angka positif");
} else {
    System.out.println("Angka tidak positif (bisa nol atau negatif)");
}
```

Dalam contoh ini, karena angka bernilai -5 (yang tidak lebih besar dari 0), kondisi *if* (*angka > 0*) bernilai *false*. Oleh karena itu, blok kode di dalam pernyataan *else* akan dieksekusi, dan program akan mencetak "Angka tidak positif (bisa nol atau negatif)".

2.4 Pengertian If Else

Pernyataan *if else* kondisional di Java digunakan untuk mengevaluasi beberapa kondisi secara berurutan. Jika kondisi pertama dalam *if* bernilai salah, program akan memeriksa kondisi *else if* berikutnya. Pernyataan *else if* dieksekusi jika kondisi tersebut benar, dan program kemudian akan melewati *else if* dan *else* berikutnya. Jika semua kondisi *if* dan *else if* bernilai salah, maka blok *else* (jika ada) yang akan dieksekusi.

Sintaks Dasar:


```

if (kondisi1) {
    // Blok kode jika kondisi1 bernilai true
} else if (kondisi2) {
    // Blok kode jika kondisi1 bernilai false dan kondisi2 bernilai true
} else if (kondisi3) {
    // Blok kode jika kondisi1 dan kondisi2 bernilai false, dan kondisi3
    bernilai true
} else {
    // Blok kode jika semua kondisi di atas bernilai false
}

```

Cara Kerja:

- Program dimulai dengan mengevaluasi kondisi pertama di dalam *if*.
- Jika kondisi1 adalah *true*, blok kode di dalam *if* akan dieksekusi, dan sisanya (termasuk *else if* dan *else*) akan dilewati.
- Jika kondisi1 adalah *false*, program beralih ke pernyataan *else if* pertama.
- Jika kondisi2 di dalam *else if* adalah *true*, blok kode di dalamnya akan dieksekusi, dan program akan melewati semua *else if* berikutnya serta blok *else*.
- Proses ini berlanjut untuk setiap *else if* hingga salah satu kondisi bernilai *true*.
- Jika semua kondisi *if* dan *else if* bernilai *false*, maka blok *else* akan dieksekusi (jika pernyataan *else* ada).

Contoh penggunaan:

```

public class ContohElseIf {
    public static void main(String[] args) {
        int nilai = 70;

        if (nilai >= 90) {
            System.out.println("Nilai Anda A");
        } else if (nilai >= 80) {

```

```

        System.out.println("Nilai Anda B");
    } else if (nilai >= 70) {
        System.out.println("Nilai Anda C"); // Kondisi ini terpenuhi
    } else if (nilai >= 60) {
        System.out.println("Nilai Anda D");
    } else {
        System.out.println("Nilai Anda E");
    }
}
}

```

Dalam contoh ini, karena nilai adalah 70, maka program akan mengeksekusi blok kode di dalam *else if* (nilai >= 70) dan mencetak "Nilai Anda C", lalu mengabaikan sisa blok *else if* dan *else*.

2.5 Pengertian Switch

Pernyataan *switch* pada statement kondisional Java adalah pernyataan alur kontrol yang memungkinkan eksekusi salah satu blok kode dari beberapa alternatif, berdasarkan nilai dari sebuah ekspresi. Pernyataan ini digunakan sebagai alternatif yang lebih ringkas dan efisien daripada serangkaian *if-else* saat membandingkan satu ekspresi dengan banyak nilai. *Switch* mendukung nilai *byte*, *short*, *char*, *int*, *long*, *enum*, *String*, atau kelas pembungkusnya.

Cara Kerja *switch*:

- Ekspresi yang Dievaluasi: Pernyataan *switch* dimulai dengan *switch* (ekspresi), di mana ekspresi ini adalah variabel atau nilai yang akan dibandingkan dengan berbagai kondisi (kasus).
- Kasus (*Case*): Setiap *case* memiliki nilai literal atau konstanta yang akan dibandingkan dengan ekspresi. Jika ekspresi cocok dengan nilai *case*, blok kode di bawah *case* tersebut akan dieksekusi.
- Pernyataan *break*: Kata kunci *break* sangat penting untuk menghentikan eksekusi blok kode *case* setelah kecocokan

ditemukan. Tanpa *break*, program akan "turun" dan mengeksekusi blok kode di *case* berikutnya meskipun tidak ada kecocokan lagi, yang sering kali bukan perilaku yang diinginkan.

- Kasus default: Kata kunci default bersifat opsional. Jika tidak ada satu pun *case* yang cocok dengan nilai ekspresi, blok kode di bawah default akan dieksekusi.

Contoh Sintaks:

```
switch (ekspresi) {  
    case nilai1:  
        // Kode yang akan dijalankan jika ekspresi == nilai1  
        break; // Menghentikan eksekusi setelah case ini  
    case nilai2:  
        // Kode yang akan dijalankan jika ekspresi == nilai2  
        break;  
    default:  
        // Kode yang akan dijalankan jika tidak ada case yang cocok  
}
```

Kapan Menggunakan *switch*?

- Banyak Alternatif: Ketika Anda memiliki banyak kemungkinan nilai untuk diperiksa dari satu variabel.
- Keterbacaan Kode: Membuat kode lebih mudah dibaca dan disederhanakan dibandingkan menggunakan serangkaian panjang pernyataan *if-else if*.
- Efisiensi: Dalam beberapa kasus, *switch* bisa lebih cepat daripada *if-else* karena *switch* sering kali menggunakan tabel untuk pencarian, bukan perbandingan berurutan.

Contoh Penggunaan:

Berikut adalah contoh kode Java yang menunjukkan penggunaan pernyataan *switch* untuk menentukan nama hari berdasarkan nomornya:

```
public class SwitchDemo {  
    public static void main(String[] args) {
```

```
int hari = 3; // Misal: Hari ke-3 adalah Rabu
```

```
switch (hari) {  
    case 1:  
        System.out.println("Senin");  
        break;  
    case 2:  
        System.out.println("Selasa");  
        break;  
    case 3:  
        System.out.println("Rabu"); // Ini yang akan dicetak  
        break;  
    case 4:  
        System.out.println("Kamis");  
        break;  
    case 5:  
        System.out.println("Jumat");  
        break;  
    case 6:  
        System.out.println("Sabtu");  
        break;  
    case 7:  
        System.out.println("Minggu");  
        break;  
    default:  
        System.out.println("Nomor hari tidak valid.");  
}  
}
```

Cara Kerja Contoh di Atas

Variabel hari diberi nilai 3.

Pernyataan switch mengevaluasi hari.

Sistem memeriksa case mana yang cocok dengan nilai 3.

case 3: cocok dengan nilai hari, sehingga blok kode di dalamnya `System.out.println("Rabu");` dieksekusi.

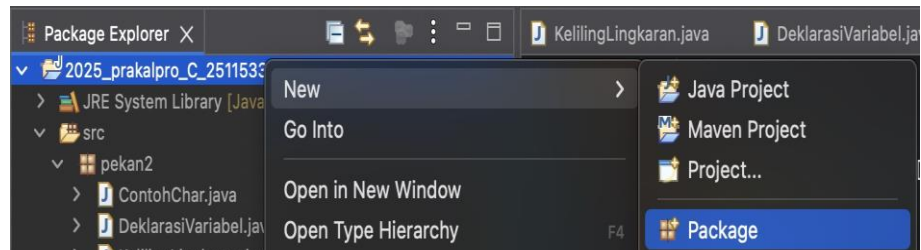
break; kemudian menghentikan eksekusi pernyataan *switch*.

Hasil yang akan ditampilkan adalah: Rabu

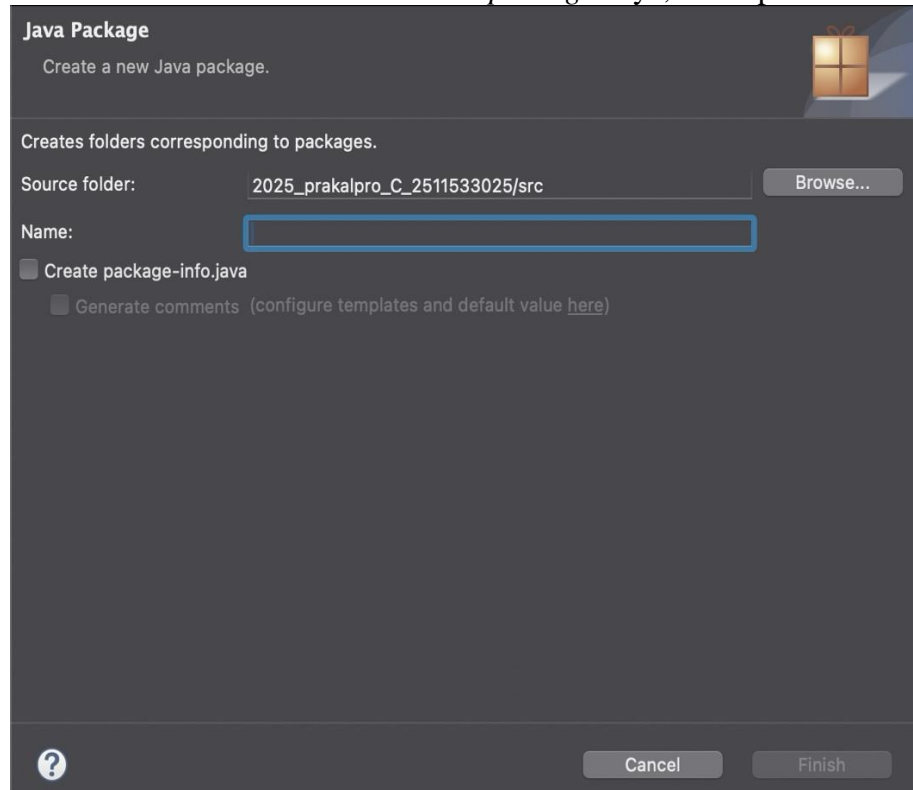
2.6 Langkah Langkah Pengerjaan Praktikum Pekan 4

Sebelum ke langkah Langkah pengerjaan contoh proyek, saya akan menjelaskan terlebih dahulu gimana caranya menambahkan proyek nya terlebih dahulu :

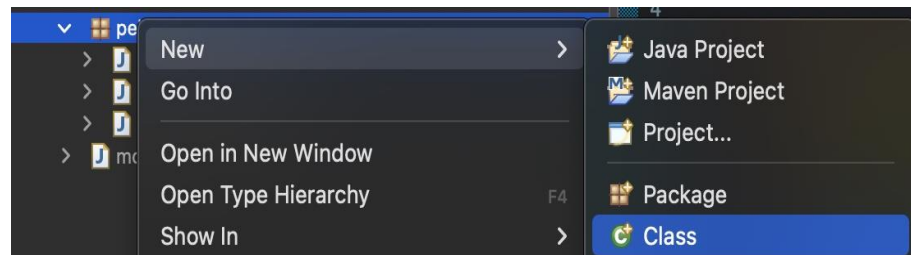
1. Setelah kalian masuk ke eclipse java dan membuat folder file proyek java, click kanan pada mouse kalian , kemudian pilih *new* terus pilih *package*.



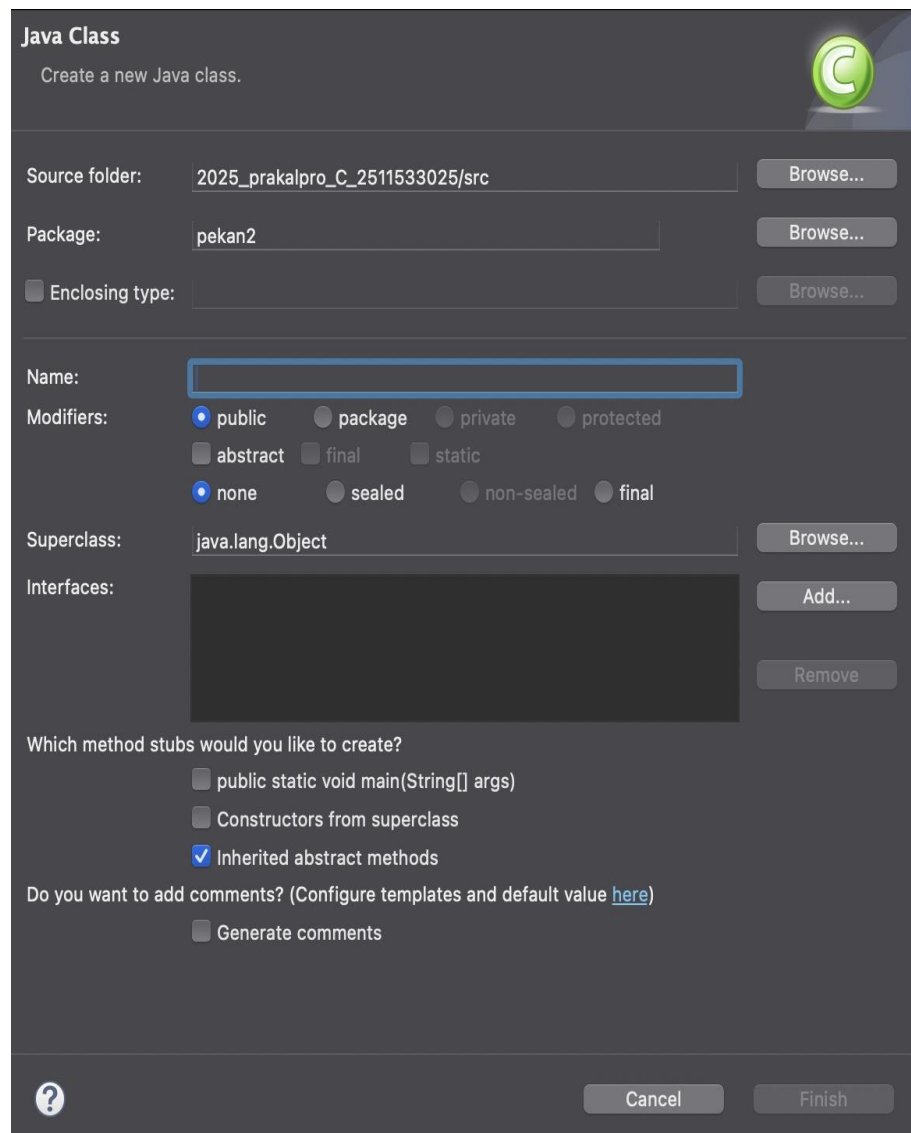
Setelah di click akan muncul halaman java *package*, dimana kita disuruh bikin nama untuk folder *package* nya, lalu pilih finish.



2. Jika folder *packagenya* sudah jadi, click kanan mouse pada folder *package* nya, lalu akan muncul pilihan, kalian pilih *new* kemudian pilih *class*.



Setelah di click akan muncul halaman java *class*, dimana kita disuruh untuk membuat nama *classnya* dan jangan lupa untuk centang pada bagian *public static void main*, lalu click finis.



Setelah selesai pembuatan folder *class* nya maka baru kita bisa membuat projek yang kita inginkan.

Catatan :

Jika kita ingin membuat proyek baru lagi, maka cukup buat folder *class* yang baru.

A. Baiklah karena proses menambahkan proyek sudah dijelaskan, selanjutnya langkah langkah pengerjaan proyek :

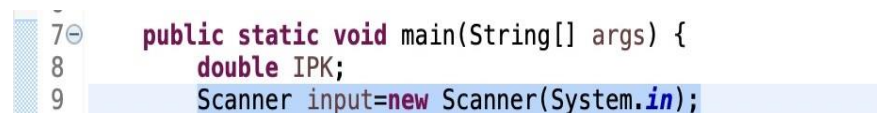
1. Membuat Proyek Latifl

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setingnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai public classnya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas, Seperti gambar dibawah ini.



```
1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class latifl {
```

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (Java Virtual Machine) untuk menjalankan aplikasi mandiri. Lalu masukan kode *Scanner* untuk membaca input dari berbagai sumber, terutama dari keyboard, dan mengurainya menjadi berbagai tipe data seperti *string*, bilangan bulat (*int*), bilangan desimal (*double*), dan tipe data lainnya. Seperti gambar dibawah ini:



```
7 public static void main(String[] args) {
8     double IPK;
9     Scanner input=new Scanner(System.in);
```

3. Masukan kode *system.out.print* dan *system.out.println* untuk melanjutkan proses programnya. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Seperti gambar dibawah ini.

```

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  }
20

```


4. Berikut adalah gambar apabila memasukan kode sudah selesai.

```

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 }

```

5. Hasil proyek Latif1.



```

Problems @ Javadoc Declaration Console X Git Staging
<terminated> latif1 [Java Application] /Volumes/Eclipse/Eclipse.app/Contents/E
Input IPK Anda = 3,97
Anda Lulus Sangat Memuaskan dengan IPK 3.97

```

2. Membuat Proyek Ifelse1

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa pembuka sebagai pembuka perintah. Seperti gambar dibawah ini.

```

1  package pekan4;
2
3  import java.util.Scanner;
4
5  public class Ifelse1 {

```


2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (Java Virtual Machine) untuk menjalankan aplikasi mandiri. Lalu masukan kode *double* untuk merepresentasikan bilangan desimal (bilangan pecahan) dengan presisi ganda, menggunakan 64-bit untuk menyimpan nilainya. Setelah itu, masukan kode *Scanner* untuk membaca input dari berbagai sumber, terutama dari keyboard, dan mengurainya menjadi berbagai tipe data seperti string, bilangan bulat (*int*), bilangan desimal (*double*), dan tipe data lainnya. Seperti gambar dibawah ini:

```
7 public static void main(String[] args) {  
8     double IPK;  
9     Scanner input=new Scanner(System.in);
```

3. Masukan kode *system.out.print* dan *system.out.println* untuk melanjutkan proses programnya. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Seperti gambar dibawah ini.

```
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     } else {  
16         System.out.println("Anda Tidak Lulus");  
17     }  
18  
19 }  
20  
21 }
```

4. Berikut adalah gambar apabila memasukan kode sudah selesai.

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class Ifelse1 {
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        } else {
16            System.out.println("Anda Tidak Lulus");
17        }
18    }
19 }
20
21 }

```

5. Hasil projek Ifelse1.



3. Membuat Projek Nilai

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public classnya* berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah. Seperti gambar dibawah ini.



2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat class baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (Java Virtual Machine) untuk menjalankan aplikasi mandiri. Setelah itu

masukan kode *int* untuk menyimpan data bilangan bulat, lalu masukan kode Scanner untuk membaca input dari berbagai sumber, terutama dari keyboard, dan mengurainya menjadi berbagai tipe data seperti *string*, bilangan bulat (*int*), bilangan desimal (*double*), dan tipe data lainnya. Seperti gambar dibawah ini:

```
6  
7 public static void main(String[] args) {  
8     int nilai;  
9     Scanner input=new Scanner(System.in);
```

3. Masukan kode *system.out.print* dan *system.out.println* untuk melanjutkan proses programnya. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Seperti gambar dibawah ini.

```
10     System.out.print("Inputkan 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("E");  
24     }  
25 }  
26  
27 }
```

4. Berikut adalah gambar apabila memasukan kode sudah selesai.

```

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("Inputkan 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("E");
24        }
25    }
26
27 }

```

5. Hasil proyek Nilai.



d. Membuat Proyek multiIf

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setingnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai public classnya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah. Seperti gambar dibawah ini.

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class multIf {

```

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (Java Virtual Machine) untuk menjalankan aplikasi mandiri. Setelah itu masukan kode *int* untuk menyimpan data bilangan bulat, masukan kode *char* karena digunakan dalam berbagai operasi manipulasi teks, seperti memproses input, mengurai data, serta operasi yang membutuhkan kecepatan dan konsistensi. lalu masukan kode *Scanner* untuk membaca input dari berbagai sumber, terutama dari keyboard, dan mengurainya menjadi berbagai tipe data seperti *string*, bilangan bulat (*int*), bilangan desimal (*double*), dan tipe data lainnya. Seperti gambar dibawah ini:

```

7 public static void main(String[] args) {
8     int umur;
9     char sim;
10    Scanner a= new Scanner(System.in);

```

3. Masukan kode *system.out.print* dan *system.out.println* untuk melanjutkan proses programnya. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Seperti gambar dibawah ini.

```

11    System.out.print("Input umur anda: ");
12    umur= a.nextInt();
13    System.out.print("Apakah Anda Sudah Punya Sim C: ");
14    sim=a.next().charAt(0);
15    a.close();
16    if((umur >= 17)&&(sim=='y')) {
17        System.out.println("Anda sudah dewasa dan boleh bawa motor");
18    }
19    if((umur >= 17)&&(sim!='y')) {
20        System.out.println("Anda sudah dewasa tetapi tidak boleh bawa motor");
21    }
22    if((umur < 17)&&(sim!='y')) {
23        System.out.println("Anda belum Cukup Umur bawa motor");
24    }
25    if((umur < 17)&&(sim=='y')) {
26        System.out.println("Anda belum Cukup Umur punya SIM");
27    }
28
29 }
30
31 }

```

4. Berikut adalah gambar apabila memasukan kode sudah selesai.

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class multIf {
6
7     public static void main(String[] args) {
8         int umur;
9         char sim;
10        Scanner a= new Scanner(System.in);
11        System.out.print("Input umur anda: ");
12        umur= a.nextInt();
13        System.out.print("Apakah Anda Sudah Punya Sim C: ");
14        sim=a.next().charAt(0);
15        a.close();
16        if((umur >= 17)&&(sim=='y')) {
17            System.out.println("Anda sudah dewasa dan boleh bawa motor");
18        }
19        if((umur >= 17)&&(sim!='y')) {
20            System.out.println("Anda sudah dewasa tetapi tidak boleh bawa motor");
21        }
22        if((umur < 17)&&(sim!='y')) {
23            System.out.println("Anda belum Cukup Umur bawa motor");
24        }
25        if((umur < 17)&&(sim=='y')) {
26            System.out.println("Anda belum Cukup Umur punya SIM");
27        }
28    }
29 }
30
31 }

```

5. Hasil proyek multIf.

```

<terminated> multIf [Java Application] /Volumes/Eclipse/Eclipse
Input umur anda: 19
Apakah Anda Sudah Punya Sim C: y
Anda sudah dewasa dan boleh bawa motor

```

e. Membuat proyek NamaBulan

1. Buat *class*, namakan sesuai dengan yang diperintahkan atau yang diinginkan, untuk format setinggnya sesuai dengan format yang sudah disediakan atau sesuai intruksi yang diberikan. Jangan sampai *public class*nya berbeda dengan nama folder *class* yang kita buat, karena merupakan kunci untuk mendeklarasikan sebuah kelas dan jangan lupa kasih kurung kurawa sebagai pembuka perintah. Seperti gambar dibawah ini.

```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class NamaBulan {
6

```

2. Biasanya kode *public static void main(String[] args)* akan muncul otomatis ketika membuat *class* baru, dimana berfungsi sebagai titik masuk utama untuk program Java yang memungkinkan JVM (Java

Virtual Machine) untuk menjalankan aplikasi mandiri. Lalu masukan kode *Scanner* untuk membaca input dari berbagai sumber, terutama dari keyboard, dan mengurainya menjadi berbagai tipe data seperti *string*, bilangan bulat (*int*), bilangan desimal (*double*), dan tipe data lainnya. Masukan kode *int* untuk menyimpan data bilangan bulat. Setelah itu, masukan kode *switch* sebagai pernyataan kontrol alur yang memungkinkan eksekusi satu blok kode dari banyak alternatif berdasarkan nilai sebuah ekspresi. Seperti gambar dibawah ini.

```
7 public static void main(String[] args) {  
8     Scanner scanner = new Scanner(System.in);  
9     System.out.print("Masukan angka bulan (1 - 12): ");  
10    int bulan = scanner.nextInt();  
11    switch (bulan) {
```

3. Masukan kode *case* karena berfungsi sebagai bagian dari pernyataan *switch* yang menentukan blok kode mana yang akan dieksekusi jika nilai ekspresi *switch* cocok dengan nilai spesifik yang diberikan pada case tersebut, lalu masukan kode *system.out.println* untuk melanjutkan proses programnya. Usahakan semua sesuai dengan arahan yang diberikan, jika ada tanda silang maka ada kode yang tidak sesuai, untuk menghilangkan tanda silangnya kalian harus mencari letak kesalahannya , jika sudah ketemu segera diperbaiki, supaya hasilnya tidak eror dan jangan lupa jika sudah selesai memasukan kode program diakhiri dengan kurung kurawa tertutup. Seperti gambar dibawah ini.

```

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("Desemeber");
47     break;
48 default:
49     System.out.println("Angka tidak valid");
50 }
51 scanner.close();
52 }
53

```

4. Berikut adalah gambar apabila memasukan kode sudah selesai.


```

1 package pekan4;
2
3 import java.util.Scanner;
4
5 public class NamaBulan {
6
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         System.out.print("Masukan 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            default:
49                System.out.println("Angka tidak valid");
50        }
51        scanner.close();
52    }
53
54 }

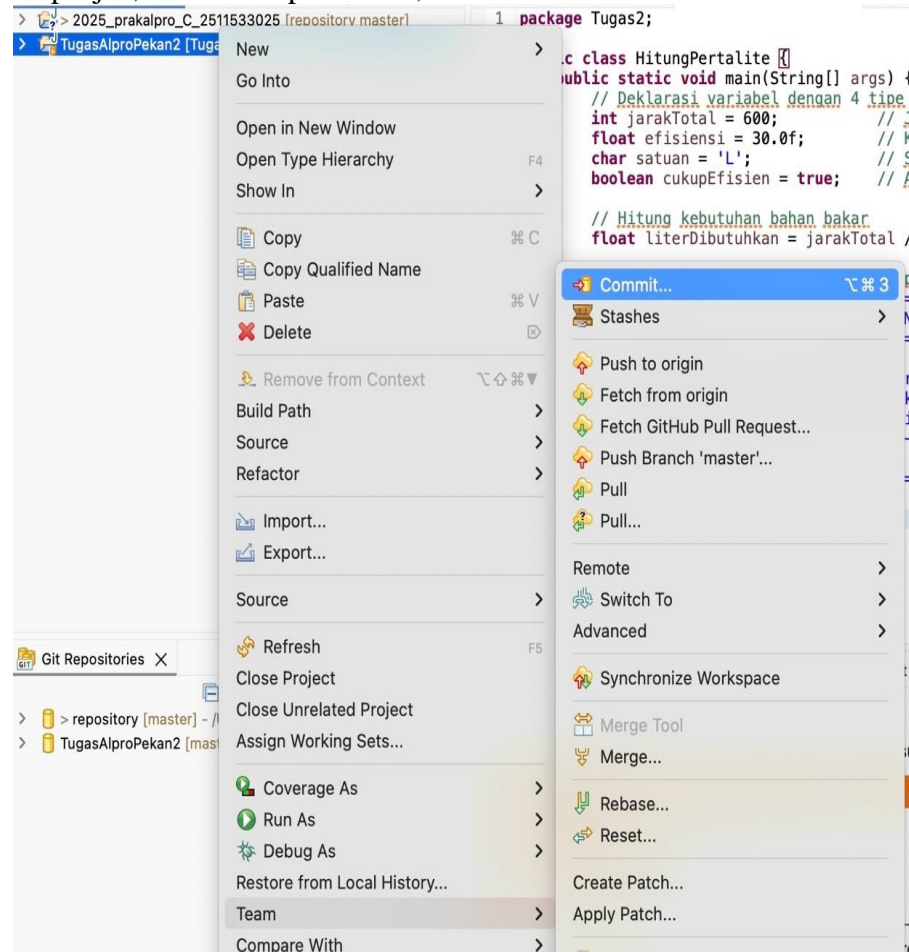
```

5. Hasil proyek NamaBulan.



B. Langkah Penyimpanan

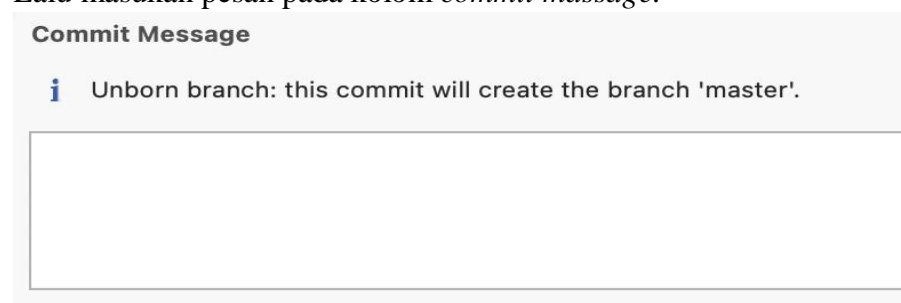
1. Sebelum kita masuk ke langkah, kita harus buat akun github dulu. Jika sudah buat akun github, baru bisa kita simpan di githubnya. Selanjutnya ikutin langkah ini. Tekan kanan mouse pada folder proyek, setelah itu pilih team, terus ke *commit*.



2. Lalu tekan tombol plus 2/double plus warna hijau.



3. Lalu masukan pesan pada kolom *commit message*.



4. Setelah itu tekan *commit* and pus, lalu ikutin arahan untuk memasukan nama dan pw akun git hub kalian, maka projek kalian sudah tersimpan di akun github kalian.



5. Projek yang sudah disimpan di akun projek.



DikaSiNpc/2025_prakalpro_C_251153
3025

Sedikit tambahan jika ingin cek hasil program nya tekan tombol play warna hijau.



Tapi pada program yang sekarang berbeda dari sebelumnya karena pada program ini input kita masukan angka atau huruf sesuai keinginan kita pada kolom *console*, tapi ingat harus sesuai perintah kode yang kita masukan.

BAB 3

PENUTUP

3.1 Kesimpulan

Dari praktikum yang telah dilakukan, maka dapat diambil kesimpulan bahwa Statement Kondisional merupakan salah satu konsep fundamental dalam pemrograman Java yang memungkinkan program mengambil keputusan berdasarkan kondisi tertentu. Melalui praktikum ini, dapat disimpulkan bahwa struktur percabangan seperti *if*, *else*, *if else*, dan *switch* sangat penting untuk mengontrol alur eksekusi program sesuai dengan logika yang diinginkan. Setiap bentuk statement kondisional memiliki karakteristik dan penggunaan yang berbeda, tergantung pada kompleksitas kondisi dan jumlah kemungkinan hasil. Pemahaman yang baik terhadap sintaks, logika evaluasi kondisi, serta penggunaan operator relasional dan logika sangat menentukan keberhasilan dalam mengimplementasikan percabangan secara efektif. Tapi jika salah satu aja kode yang dibuat maka akan terjadi error. Oleh karena itu kita harus berhati hati ketika memasukan kode dan ikutin proses sesuai arahan yang diberikan. Dan saya sangat senang melaksanakan praktikum ini karena bisa menambah ilmu pengetahuan dan wawasan akan kemajuan teknologi.

3.2 Saran

1. Perbanyak Latihan Kasus Nyata

Agar lebih terbiasa, disarankan untuk mencoba berbagai skenario penggunaan statement kondisional dalam konteks kehidupan sehari-hari-seperti sistem penilaian, kalkulator diskon, validasi input, atau menu interaktif.

2. Perhatikan Tata Penulisan Kode (Indentasi dan Format)

Penulisan kode yang rapi dan terstruktur memudahkan pembacaan, debugging, serta pengembangan program di masa depan-terutama saat menggunakan percabangan bersarang (*nested if*).

3. Pahami Perbedaan if-else dan switch

Gunakan *switch* hanya untuk kondisi yang melibatkan perbandingan nilai tetap (konstan) dari satu variabel, sedangkan *if-else* lebih fleksibel untuk kondisi kompleks yang melibatkan operator logika atau relasional.

4. Uji Berbagai Kondisi Input

Lakukan pengujian dengan berbagai jenis input (termasuk input ekstrem atau tidak valid) untuk memastikan program bersifat *robust* dan tidak mudah error.

DAFTAR PUSTAKA

- Sumber daring (website):
 - [1] ScholarHat, "Java Conditional Statements: *If, If-Else and Switch*," ScholarHat. [Online]. Available: <https://www.scholarhat.com/tutorial/java/java-conditional-statements-if-else-switch>. [Accessed: Apr. 5, 2025].
 - [2] W3Schools, "Java Conditions," W3Schools Online Web Tutorials. [Online]. Available: https://www.w3schools.com/java/java_condition_s.asp. [Accessed: Apr. 5, 2025].
 - [3] DotNet Tricks, "Java Conditional Statements: *If, Else and Switch*," LinkedIn, Jan. 14, 2023. [Online]. Available: <https://www.linkedin.com/pulse/java-conditional-statements-else-switch-dotnettricks>. [Accessed: Apr. 5, 2025].
 - [4] Unstop, "*If-Else Statement In Java | Types, Syntax & More (+Code Examples)*," Unstop Blog, Feb. 6, 2025. [Online]. Available: <https://unstop.com/blog/if-elsestatement-java>. [Accessed: Apr. 5, 2025].
 - [5] Oracle, "*The switch Statement*," Java Tutorials, Oracle Corp. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>. [Accessed: Apr. 5, 2025].