

Nama & NPM	Topik:	Tanggal:
Zaira Ayu Wandira (G1F024055)	IF dan SWITCH Java	19 september 2024
[No.1] Identifikasi Masalah:		
<p>1.1. Bilangan genap merupakan bilangan yang habis dibagi 2. Bilangan ganjil adalah bilangan yang tidak habis dibagi 2. Analisa kode program yang tepat untuk menghitung masukan pengguna termasuk bilangan genap atau bilangan ganjil (lihat Contoh 1)? (Petunjuk: $\text{hitung} = \text{nilai} \% 2$ jika $\text{hitung} = 0$ maka bilangan genap, jika $\text{hitung} = 1$ maka bilangan ganjil)</p> <p>1.2. Cermati contoh 2, analisa kondisi pada IF bersarang! Tambahkan satu kondisi IF dengan satu nilai input Quiz (nilaiQ). Jika nilai UTS, Tugas, dan Quiz lebih besar sama dengan 80 maka siswa mendapat nilai A.</p> <p>1.3. Apakah ketiga kondisi IF pada Contoh 1.2. dapat diringkas menjadi satu kondisi? Periksa satu kondisi mana yang paling tepat menggantikan ketiga kondisi itu!</p> <ol style="list-style-type: none"> IF ($\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80$) IF ($\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \ \&\& \ \text{nilaiQ} \geq 80$) IF ($\text{nilaiU} \geq 80 \ \&\& \ \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80$) IF ($\text{nilaiU} \geq 80 \ \&\& \ \text{nilaiT} \geq 80 \ \&\& \ \text{nilaiQ} \geq 80$) <p>1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!</p>		
[No.1] Analisis dan Argumentasi		
<p>1.1. Analisis kode Kode latihan 1.1 bertujuan untuk menentukan apakah sebuah bilangan yang dimasukkan oleh pengguna adalah genap atau ganjil. Dimulai dengan mengimpor kelas Scanner dari paket java.util, program ini membuat objek Scanner untuk membaca input dari konsol. Pengguna diminta untuk memasukkan sebuah angka, yang kemudian disimpan dalam variabel nilai. Dengan menggunakan operator modulus (%), program menghitung sisa bagi dari nilai ketika dibagi 2 dan menyimpannya dalam variabel hitung. Selanjutnya, percabangan if digunakan untuk memeriksa nilai hitung: jika sama dengan 0, program mencetak "Bilangan Genap"; jika tidak, program mencetak "Bilangan Ganjil".</p> <p>1.2. Analisis kondisi if bersarang Kode 1.2 meminta pengguna untuk memasukkan nilai tugas, quiz, dan UTS, kemudian menggunakan struktur if bersarang untuk menentukan apakah siswa mendapatkan nilai A. Jika semua nilai di atas 80, program mencetak "Anda mendapatkan nilai A"; jika tidak, ia mencetak "Anda TIDAK mendapatkan nilai A". Struktur bersarang membantu memeriksa setiap kondisi secara berurutan, tetapi bisa lebih ringkas menggunakan operator logika.</p> <p>1.3. Meringkas Ketiga Kondisi IF Ketiga kondisi IF tersebut dapat diringkas menjadi satu kondisi yang lebih sederhana. Jawaban yang tepat adalah: d. IF ($\text{nilaiU} \geq 80 \ \&\& \ \text{nilaiT} \geq 80 \ \&\& \ \text{nilaiQ} \geq 80$) Ini karena semua tiga nilai harus memenuhi syarat untuk mendapatkan nilai A.</p>		

[No.1] Penyusunan Algoritma dan Kode Program

1.1 Program untuk Mengecek Bilangan Genap atau Ganjil

Algoritma:

- 1) Mulai
- 2) Impor Scanner
- 3) Buat objek Scanner untuk membaca input
- 4) Tampilkan pesan: "Masukkan Angka Anda: "
- 5) Baca input dari pengguna dan simpan ke variabel nilai
- 6) Hitung sisa bagi 2 dari nilai dan simpan ke variabel hitung
- 7) Jika hitung == 0
 - a. Tampilkan "Bilangan Genap"
- 8) Jika tidak
 - a. Tampilkan "Bilangan Ganjil"
- 9) Selesai

Kode Program:

```
1 import java.util.Scanner; // Mengimport Scanner untuk input pengguna
2
3 public class GenapGanjil {
4     public static void main(String[] args) { //method utama
5         Scanner input = new Scanner(System.in); // Membuat Scanner untuk input
6         System.out.print("Masukkan Angka Anda: "); // Meminta input dari
           pengguna
7         int nilai = input.nextInt(); // Menyimpan input ke variabel 'nilai'
8
9         int hitung = nilai % 2; // Menghitung sisa bagi 2
10
11         if (hitung == 0) { // Cek apakah genap
12             System.out.println("Bilangan Genap"); // Output jika genap
13         } else { // Jika tidak genap
14             System.out.println("Bilangan Ganjil"); // Output jika ganjil
15         }
16     }
17 }
```

```
java -cp /tmp/DZ1CS1qswE/GenapGanjil
Masukkan Angka Anda: 4
Bilangan Genap
=== Code Execution Successful ===
```

1.2. program tambahkan satu kondisi IF

Algoritma

- 1) Mulai
- 2) Inisialisasi Scanner:
 - Buat objek Scanner untuk membaca input pengguna.
- 3) Input Nilai:
 - Tampilkan pesan "Masukkan Angka Tugas Anda:" dan simpan input ke variabel nilaiT.
 - Tampilkan pesan "Masukkan Angka Quiz Anda:" dan simpan input ke variabel nilaiQ.
 - Tampilkan pesan "Masukkan Angka UTS Anda:" dan simpan input ke variabel nilaiU.
- 4) Cek Kondisi:
 - Jika nilaiU >= 80 dan nilaiT >= 80 dan nilaiQ >= 80, lakukan langkah 5.
 - Jika tidak, lakukan langkah 6.
- 5) Output Nilai A:
 - Tampilkan "Anda mendapatkan nilai A".
- 6) Output Tidak Mendapatkan Nilai A:
 - Tampilkan "Anda TIDAK mendapatkan nilai A".
- 7) Selesai

Code program:

```
1- import java.util.Scanner;
2
3- public class IfBersarang {
4-     public static void main(String[] args) {
5         Scanner varT = new Scanner(System.in);
6         System.out.print("Masukkan Angka Tugas Anda: ");
7         int nilaiT = varT.nextByte();
8
9         Scanner varQ = new Scanner(System.in);
10        System.out.print("Masukkan Angka Quiz Anda: ");
11        int nilaiQ = varQ.nextByte();
12
13        Scanner varU = new Scanner(System.in);
14        System.out.print("Masukkan Angka UTS Anda: ");
15        int nilaiU = varU.nextByte();
16
17        if (nilaiU >= 80) {
18            if (nilaiT >= 80) {
19                if (nilaiQ >= 80) {
20                    System.out.println("Anda mendapatkan nilai A");
21                } }
22        } else {
23            System.out.println("Anda TIDAK mendapatkan nilai A");
24        } }
```

```
java -cp /tmp/6YmT407eXa/IfBersarang
Masukkan Angka Tugas Anda: 80
Masukkan Angka Quiz Anda: 85
Masukkan Angka UTS Anda: 90
Anda mendapatkan nilai A

=== Code Execution Successful ===
```

1.3.program Meringkas Ketiga Kondisi IF

Algoritma

- 1) Mulai
- 2) Inisialisasi Scanner:
 - Buat objek Scanner untuk membaca input pengguna.
- 3) Input Nilai:
 - Tampilkan pesan "Masukkan Angka Tugas Anda:" dan simpan input ke variabel nilaiT.
 - Tampilkan pesan "Masukkan Angka Quiz Anda:" dan simpan input ke variabel nilaiQ.
 - Tampilkan pesan "Masukkan Angka UTS Anda:" dan simpan input ke variabel nilaiU.
- 4) Cek Kondisi:
 - Jika nilaiU >= 80 dan nilaiT >= 80 dan nilaiQ >= 80, lakukan langkah 5.
 - Jika tidak, lakukan langkah 6.
- 5) Output Nilai A:
 - Tampilkan "Anda mendapatkan nilai A".
- 6) Output Tidak Mendapatkan Nilai A:
 - Tampilkan "Anda TIDAK mendapatkan nilai A".
- 7) Selesai

Code program:

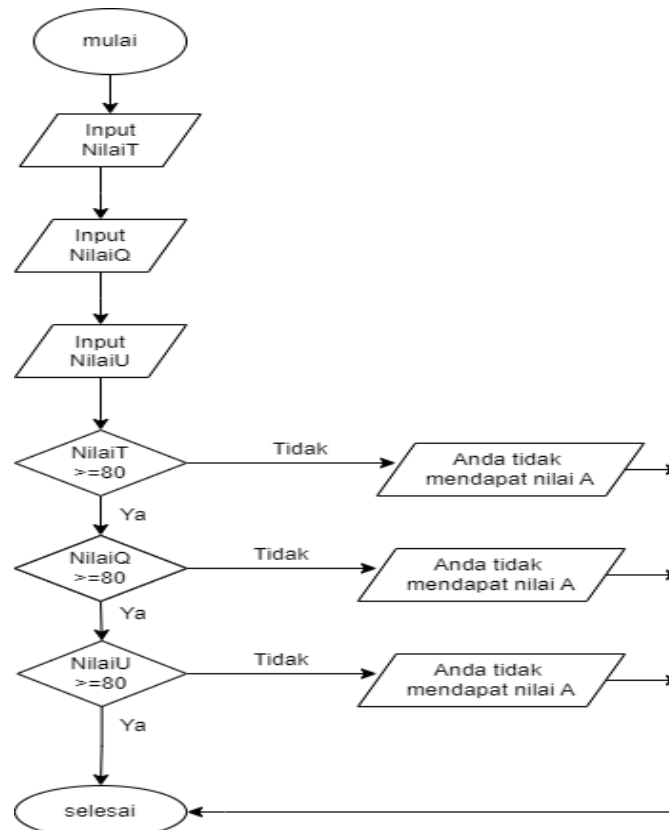
```
1- import java.util.Scanner; // Mengimpor kelas Scanner untuk membaca input
   pengguna
2
3- public class IfBersarang {
4-     public static void main(String[] args) {
5-         Scanner varT = new Scanner(System.in); // Membuat objek Scanner
6-         System.out.print("Masukkan Angka Tugas Anda: ");
7-         int nilaiT = varT.nextInt(); // Membaca input nilai tugas
8-
9-         System.out.print("Masukkan Angka Quiz Anda: ");
10-        int nilaiQ = varQ.nextInt(); // Membaca input nilai quiz
11-
12-        System.out.print("Masukkan Angka UTS Anda: ");
13-        int nilaiU = varU.nextInt(); // Membaca input nilai UTS
14-
15-        // Memeriksa apakah semua nilai di atas 80 untuk mendapatkan nilai A
16-        if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80) {
17-            System.out.println("Anda mendapatkan nilai A"); // Semua kondisi
               terpenuhi
18-        } else {
19-            System.out.println("Anda TIDAK mendapatkan nilai A"); // Salah satu
               kondisi tidak terpenuhi
20-        }
21-    }
22 }
```

```
java -cp /tmp/QpT111eLPw/IfBersarang
Masukkan Angka Tugas Anda: 89
Masukkan Angka Quiz Anda: 90
Masukkan Angka UTS Anda: 95
Anda mendapatkan nilai A

=== Code Execution Successful ===
```

1.4 Uraian Diagram Flowchart untuk Latihan 1.2

- 1) Mulai
- 2) Input nilaiU
- 3) Input nilaiT
- 4) Input nilaiQ
- 5) Cek Kondisi: nilaiU >= 80
Cek kondisi: nilaiT >= 80
Cek kondisi: nilaiQ >= 80
Ya: Output "Anda mendapatkan nilai A"
Tidak: Output "Anda TIDAK mendapatkan nilai A"
- 6) Selesai



[No.1] Kesimpulan

Latihan 1.1 dan 1.2 memperkenalkan konsep dasar pemrograman menggunakan percabangan dalam Java. Dalam latihan 1.1, kita menggunakan operator modulus untuk menentukan apakah sebuah bilangan genap atau ganjil, sementara dalam latihan 1.2, kita mengevaluasi nilai siswa berdasarkan input nilai tugas, quiz, dan UTS.

Penggunaan if bersarang dalam latihan 1.2 memungkinkan pemeriksaan kondisi secara berurutan, tetapi dapat menjadi rumit dan kurang efisien. Oleh karena itu, merangkum kondisi dengan menggunakan operator logika `&&` lebih disarankan untuk membuat kode lebih ringkas dan mudah dibaca.

Flowchart yang diuraikan memberikan gambaran visual tentang alur logika program, memudahkan pemahaman tentang bagaimana input dan kondisi diolah. Secara keseluruhan, pembahasan ini menekankan pentingnya struktur dan efisiensi dalam penulisan kode, serta pemahaman logika dasar dalam pemrograman.

[No.2] Identifikasi Masalah:

2.1. Cermati kode pada Contoh 3.

Hapuslah kode break; pada //baris 1, eksekusi kembali.

Kemudian hapuslah kode break; pada //baris 2, eksekusi kembali.

Analisis perbedaan hasil luaran ketika kode menggunakan break, ketika kode break baris 1 dihapus, dan ketika kode break baris 2 dihapus!

Analisa kegunaan baris kode dengan break dan kata kunci default!

2.2. Cermati kode pada Contoh 4 yang menampilkan jumlah hari sesuai dengan bulannya.

Namun kode tersebut baru sampai bulan ke-4. Tambahkan sampai bulan ke-12 pada baris ke-1 dan baris ke-2.

2.3. Cermati permasalahan yang dipecahkan pada Contoh 3.

Apakah masalah ini bisa diubah menjadi perintah IF?

Jika bisa, rincikan analisa Anda!

Bandingkan masalah yang dapat diselesaikan percabangan dengan IF atau SWITCH !

2.4. Buatlah dokumentasi gambar flowchart dari Latihan 2.1. dan Latihan 2.2!

[No.1] Analisis dan Argumentasi

2.1. Analisis Perbedaan Hasil Luaran:

1. Dengan break: Setiap case hanya dieksekusi satu kali. Misalnya, jika input 'A', hanya "Anda sudah rajin belajar" yang dicetak.
2. Tanpa break pada Baris 1: Jika break dihapus, setelah mencetak "Anda sudah rajin belajar", eksekusi akan terus berlanjut ke case 'B', mencetak "Anda perlu kurangi main game".
3. Tanpa break pada Baris 2: Jika break dihapus, setelah mencetak "Anda perlu kurangi main game", program akan melanjutkan ke default, mencetak "Pilihan anda diluar A atau B".
4. Kegunaan break dan default:
5. break menghentikan eksekusi setelah sebuah case diproses, mencegah "fall-through" ke case berikutnya.
6. default menangani kasus ketika tidak ada nilai yang cocok dengan case yang ada, memberikan pengertian bahwa input tidak valid.

2.3. Analisis Permasalahan Contoh 3: Masalah yang dipecahkan pada Contoh 3 adalah menentukan respon berdasarkan pilihan pengguna (A atau B). Dengan menggunakan pernyataan switch, program dapat dengan cepat memberikan respon yang sesuai. Apakah Bisa Diubah Menjadi IF? Ya, masalah ini dapat diubah menjadi pernyataan if.

Perbandingan Penggunaan IF dan SWITCH:

- **SWITCH:**
 - Lebih terstruktur dan mudah dibaca saat menangani banyak nilai tetap (case).
 - Secara visual lebih jelas dalam menunjukkan pilihan yang mungkin.
- **IF:**
 - Lebih fleksibel; bisa menangani kondisi yang lebih kompleks (misalnya, rentang nilai atau kombinasi kondisi).
 - Dapat digunakan untuk tipe data yang lebih beragam (tidak terbatas pada nilai tetap).

[No.2] Penyusunan Algoritma dan Kode Program

Latihan 2.2

Algoritma

- 1). Mulai
- 2). Deklarasi Variabel:
 - Deklarasikan byte bulan
 - Deklarasikan int tahun dan set ke 2022
 - Deklarasikan int jumlahHari dan set ke 0
- 3).Tampilkan Pesan:
 - Tampilkan "Masukkan data bulan (dalam angka):"
- 4).Baca Input:
 - Baca input bulan dari pengguna dan simpan ke variabel bulan
- 5).Pernyataan Switch:
 - Lakukan switch berdasarkan bulan
 - Case 1, 3, 5, 7, 8, 10, 12:
 - Set jumlahHari = 31
 - Lanjutkan dengan break
 - Case 4, 6, 9, 11:
 - Set jumlahHari = 30
 - Lanjutkan dengan break
 - Case 2:
 - Jika tahun % 4 == 0
 - Set jumlahHari = 29
 - Jika tidak
 - Set jumlahHari = 28
 - Lanjutkan dengan break
 - Default:
 - Tampilkan "Maaf bulan hanya sampai 12."
 - Lanjutkan dengan break
- 6).Tampilkan Hasil:
 - Tampilkan "Jumlah hari = " diikuti dengan nilai jumlahHari
- 7).Selesai

Code program:

```
1- import java.util.Scanner;
2
3- public class SwitchBersarang {
4
5-     public static void main(String[] args) {
6         byte bulan;
7         int tahun = 2022;
8         int jumlahHari = 0;
9         System.out.print("Masukkan data bulan (dalam angka): ");
10        Scanner masukData = new Scanner(System.in);
11        bulan = masukData.nextByte();
12
13-        switch (bulan) {
14            case 1: case 3: case 5: case 7: case 8: case 10: case 12: // baris 1
15                jumlahHari = 31;
16                break;
17            case 4: case 6: case 9: case 11: // baris 2
18                jumlahHari = 30;
19                break;
20            case 2:
21                if (tahun % 4 == 0)
22                    jumlahHari = 29;
23                else
24                    jumlahHari = 28;
25                break;
26            default:
27                System.out.println("Maaf bulan hanya sampai 12.");
28                break;
29        }
30        System.out.println("Jumlah hari = " + jumlahHari);
31    } }
```

java -cp ./tmp/zXE2KS8gA4/SwitchBersarang
Masukkan data bulan (dalam angka): 2
Jumlah hari = 28
=== Code Execution Successful ===

Latihan 2.3

Algoritma

1. Mulai
2. Deklarasi Variabel:
 - Deklarasikan char data untuk menyimpan pilihan pengguna.
3. Tampilkan Pesan:
 - Tampilkan "Pilih A atau B :"
4. Baca Input:
 - Baca input karakter dari pengguna dan simpan ke variabel data.
5. Cek Kondisi:
 - Jika data == 'A':
 - Tampilkan "Anda sudah rajin belajar"
 - Jika data == 'B':
 - Tampilkan "Anda perlu kurangi main game"
 - Jika tidak:
 - Tampilkan "Pilihan anda diluar A atau B"
6. Selesai

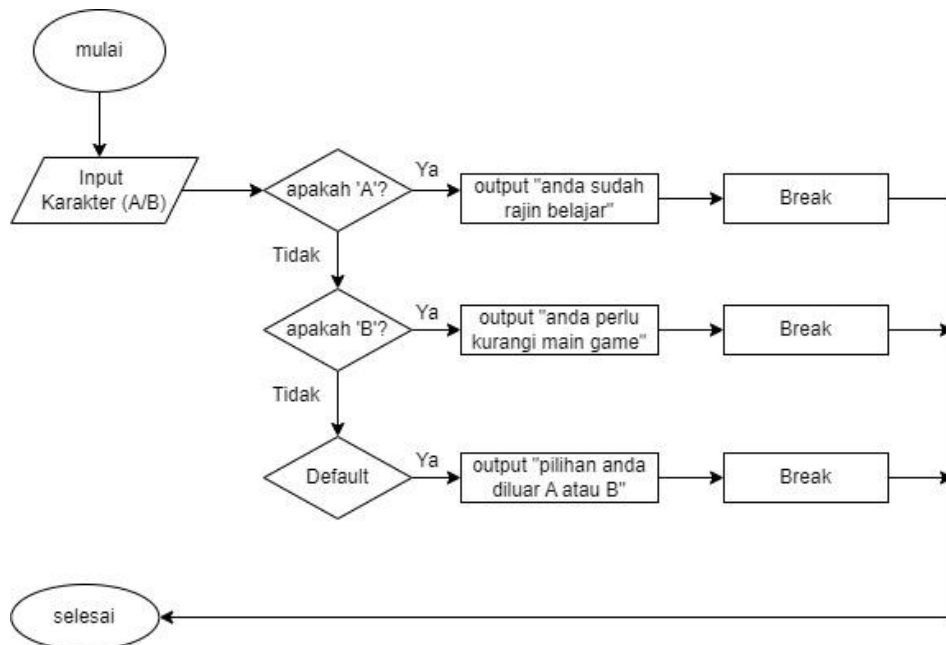
Code program

```
1- import java.util.Scanner;
2
3- public class SwitchBersarang {
4-     public static void main(String[] args) {
5-         Scanner masukData = new Scanner(System.in);
6-         // mengambil input
7-         System.out.print("Pilih A atau B : ");
8-         char data = masukData.next().charAt(0);
9
10-        if (data == 'A') {
11-            System.out.print("Anda sudah rajin belajar");
12-        } else if (data == 'B') {
13-            System.out.print("Anda perlu kurangi main game");
14-        } else {
15-            System.out.print("Pilihan anda diluar A atau B");
16-        }
17-    }
18- }
```

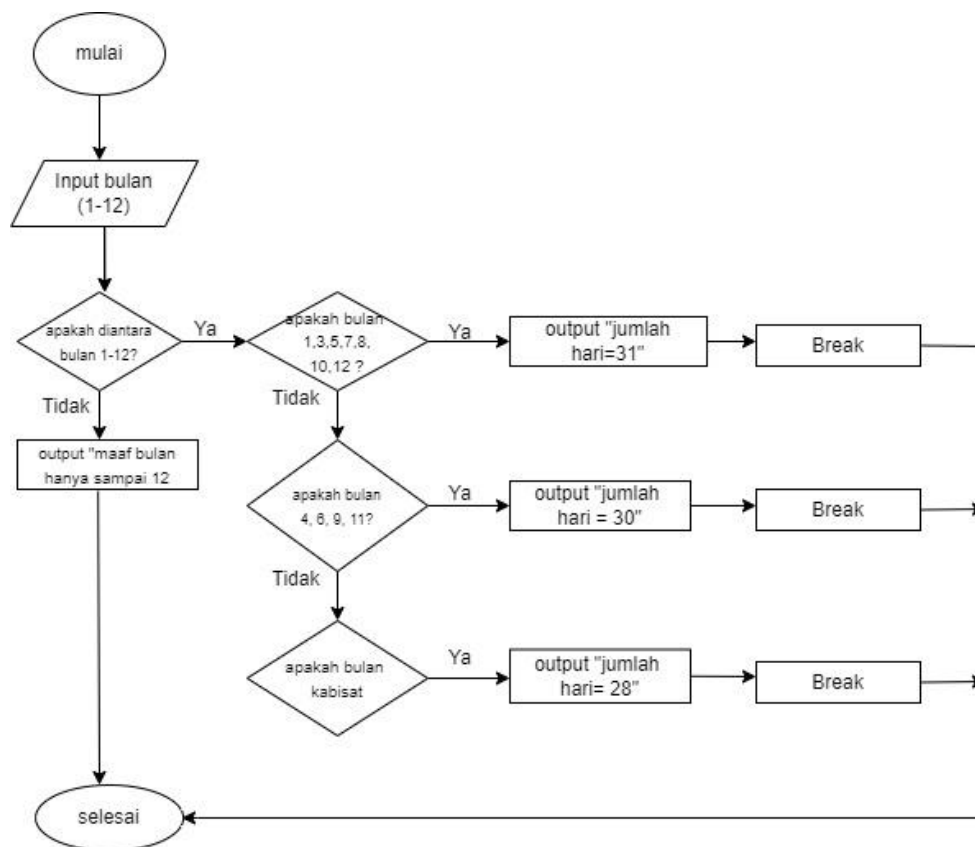
```
java -cp /tmp/KHy6MBJbba/SwitchBersarang
Pilih A atau B : A
Anda sudah rajin belajar
=== Code Execution Successful ===
```

2.4 flowchart dari Latihan 2.1. dan Latihan 2.2

Flowchart 2.1



Flowchart 2.2



[no.2] Kesimpulan:

Kesimpulan dari Latihan 2 menunjukkan bahwa perintah switch digunakan untuk menangani berbagai pilihan berdasarkan nilai dari suatu ekspresi, dan pentingnya pernyataan break untuk menghentikan eksekusi setelah sebuah case terpenuhi. Tanpa break, program akan melanjutkan eksekusi ke case berikutnya meskipun kondisinya tidak sesuai, yang dapat menyebabkan hasil yang tidak diinginkan. Perintah default juga sangat berguna untuk menangani input yang tidak sesuai dengan case yang ada, sehingga membantu program lebih andal dalam menangani kesalahan input.

Refleksi: Setelah mempelajari materi IF dan SWITCH, saya menyadari pentingnya percabangan dalam pemrograman. IF sangat fleksibel untuk memeriksa kondisi logika yang kompleks, sementara SWITCH lebih efisien untuk menangani banyak pilihan diskrit seperti penentuan bulan atau input karakter. Saya juga memahami pentingnya pernyataan break dalam switch untuk mencegah eksekusi lanjutan yang tidak diinginkan.

Melalui latihan, saya belajar bahwa visualisasi logika program dengan flowchart sangat membantu dalam memahami alur keputusan sebelum menulis kode. Pemahaman akan IF dan SWITCH memberikan keterampilan dasar yang penting dalam menyusun alur logika dan menyelesaikan masalah pemrograman secara efisien.