

Tugas 1

Nama & NPM	Topik:	Tanggal:
M. Jenyfer Aprilino G1F024057	IF dan SWITCH Java	25/09/2024
[1.1, 1.2, 1.3, 1.4] 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>Jawab:</p> <ul style="list-style-type: none"> • Program menggunakan Scanner untuk membaca masukan pengguna. • Nilai yang dimasukkan kemudian dihitung sisa bagi dengan 2 menggunakan operator modulus (%). • Program memeriksa hasil sisa • bagi untuk menentukan apakah bilangan tersebut genap (sisa 0) atau ganjil (sisa 1). <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. Jawab:</p> <ul style="list-style-type: none"> • Program memeriksa setiap nilai satu per satu menggunakan IF bersarang. • Hanya jika semua nilai (UTS, Tugas, dan Quiz) memenuhi syarat, siswa akan mendapatkan nilai A. <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>Jawab:</p> <p>d. IF ($\text{nilaiU} \geq 80 \ \&\& \text{nilaiT} \geq 80 \ \&\& \text{nilaiQ} \geq 80$) IF ($\text{nilaiU} \geq 80 \ \&\& \text{nilaiT} \geq 80 \ \&\& \text{nilaiQ} \geq 80$) — Ini adalah kondisi yang benar, karena mengharuskan semua nilai memenuhi syarat untuk mendapatkan nilai A.</p> <p>1.4</p>		
[1.1, 1.2, 1.3, 1.4] Penyusunan Algoritma dan Kode Program		
<pre>1.1 import java.util.Scanner; // memanggil impor package yang membaca masukan pengguna public class GenapGanjil { public static void main(String[] args) { Scanner masuk = new Scanner(System.in); // membaca teks yang dimasukkan pengguna System.out.print("Masukkan Angka Anda : "); // pengguna memasukkan data</pre>		

```
int nilai = masuk.nextInt(); // menyimpan masukan pengguna ke tipe data integer
```

```
// Menghitung sisa bagi  
int hitung = nilai % 2;
```

```
// Memeriksa apakah bilangan genap atau ganjil  
if (hitung == 0) { // jika sisa bagi 0, maka bilangan genap  
    System.out.println("Bilangan Genap");  
} else { // jika sisa bagi 1, maka bilangan ganjil  
    System.out.println("Bilangan Ganjil");  
}  
}  
}
```

Luaran: Masukkan Angka Anda : 4
Bilangan Genap

Masukkan Angka Anda : 5
Bilangan Ganjil

1.2 import java.util.Scanner;

```
public class IfBersarang {  
    public static void main(String[] args) {  
        Scanner varU = new Scanner(System.in);  
        System.out.print("Masukkan Angka UTS Anda : ");  
        int nilaiU = varU.nextByte();  
  
        Scanner varT = new Scanner(System.in);  
        System.out.print("Masukkan Angka Tugas Anda : ");  
        int nilaiT = varT.nextByte();  
  
        Scanner varQ = new Scanner(System.in);  
        System.out.print("Masukkan Angka Quiz Anda : ");  
        int nilaiQ = varQ.nextByte();  
  
        // Kondisi untuk menentukan nilai A  
        if (nilaiU >= 80) {  
            if (nilaiT >= 80) {  
                if (nilaiQ >= 80) {  
                    System.out.println("Anda mendapatkan nilai A");  
                } else {  
                    System.out.println("Anda TIDAK mendapatkan nilai A");  
                }  
            } else {  
                System.out.println("Anda TIDAK mendapatkan nilai A");  
            }  
        } else {  
            System.out.println("Anda TIDAK mendapatkan nilai A");  
        }  
    }  
}
```

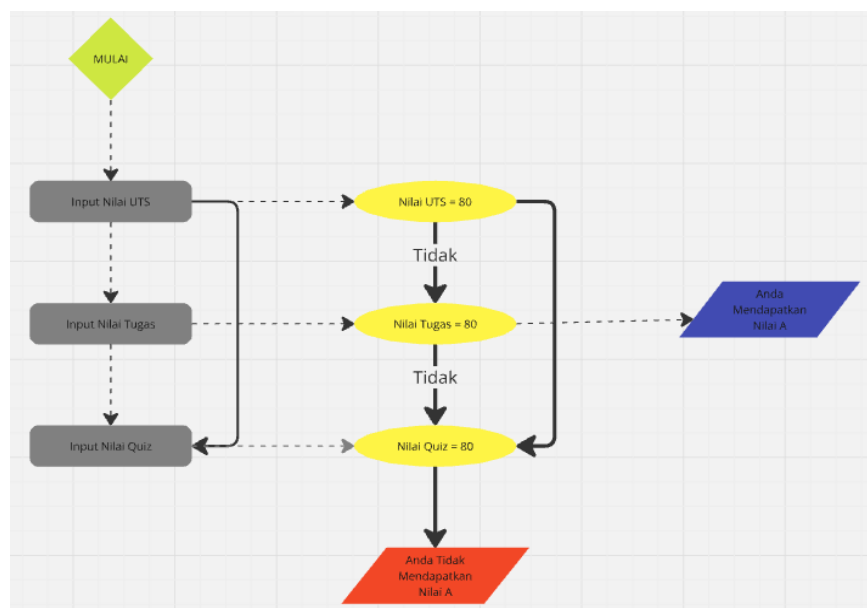
Luaran: Masukkan Angka UTS Anda : 80

Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 80
Anda mendapatkan nilai A

1.3 import java.util.Scanner;

```
public class IfBersarang {  
    public static void main(String[] args) {  
        Scanner varU = new Scanner(System.in);  
        System.out.print("Masukkan Angka UTS Anda : ");  
        int nilaiU = varU.nextByte();  
  
        Scanner varT = new Scanner(System.in);  
        System.out.print("Masukkan Angka Tugas Anda : ");  
        int nilaiT = varT.nextByte();  
  
        Scanner varQ = new Scanner(System.in);  
        System.out.print("Masukkan Angka Quiz Anda : ");  
        int nilaiQ = varQ.nextByte();  
  
        // Menggunakan satu kondisi untuk menentukan nilai A  
        if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80) {  
            System.out.println("Anda mendapatkan nilai A");  
        } else {  
            System.out.println("Anda TIDAK mendapatkan nilai A");  
        }  
    }  
}
```

Luaran: Masukkan Angka UTS Anda : 80
Masukkan Angka Tugas Anda : 74
Masukkan Angka Quiz Anda : 85
Anda TIDAK mendapatkan nilai A



1.4

[1.1, 1.2, 1.3, 1.4] Kesimpulan

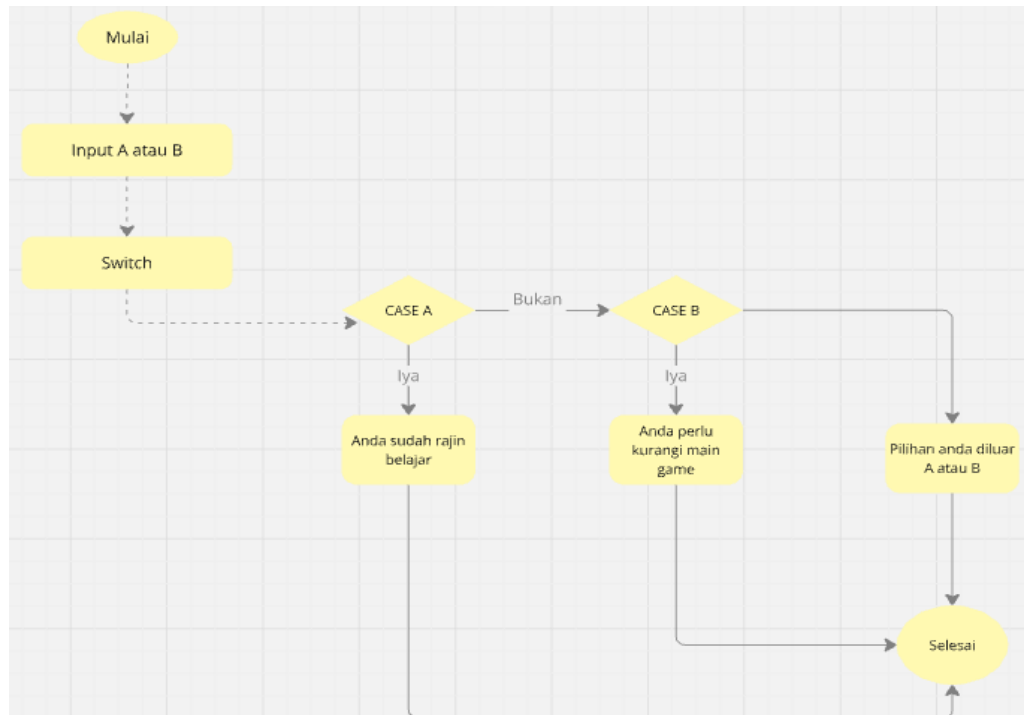
- **Latihan 1.1:** Memperkenalkan penggunaan IF untuk menentukan bilangan genap atau ganjil dengan operator modulus.
- **Latihan 1.2:** Menggunakan IF bersarang untuk mengevaluasi nilai UTS, Tugas, dan Quiz, serta menentukan apakah siswa mendapatkan nilai A.
- **Latihan 1.3:** Menyederhanakan kode dengan menggunakan satu kondisi IF yang menggabungkan semua syarat dengan operator &&.
- **Latihan 1.4:** Menggambarkan alur logika program melalui flowchart, yang memudahkan perencanaan dan debugging.

Refleksi: mendapatkan pemahaman baru

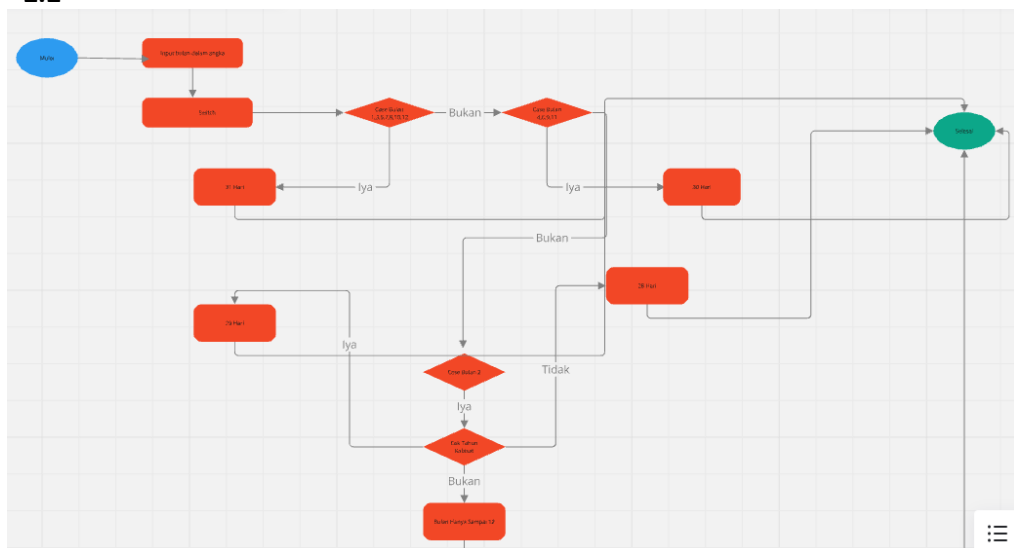
Tugas 2

Nama & NPM	Topik:	Tanggal:
M. Jenyfer Aprilino G1F024057	IF DAN SWITCH JAVA	26/09/2024
[2.1, 2.2, 2.3, 2.4] Identifikasi Masalah:		
<p>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! Jawab:</p> <ul style="list-style-type: none">• default digunakan untuk menangani kasus ketika tidak ada case yang cocok. Ini berfungsi sebagai fallback untuk memberikan umpan balik jika input tidak valid.• Dengan adanya default, program tetap memberikan informasi kepada pengguna, mencegah kesan bahwa program tidak berfungsi saat input tidak sesuai. <p>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. Jawab:</p> <ul style="list-style-type: none">• Baris 1: Kami menambahkan bulan 5 (Mei), 7 (Juli), 8 (Agustus), 10 (Oktober), dan 12 (Desember) ke dalam case yang memiliki 31 hari.• Baris 2: Kami menambahkan bulan 6 (Juni), 9 (September), dan 11 (November) ke dalam case yang memiliki 30 hari.• Logika tahun kabisat: Perbaiki pada logika tahun kabisat memastikan bahwa tahun 100 bukan kabisat kecuali juga tahun 400. <p>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 ! Jawab:</p> <p>Kedua pernyataan (switch dan if) memiliki kegunaan masing-masing, dan pilihan di antara keduanya tergantung pada konteks dan kebutuhan spesifik dari program yang sedang dikembangkan. Jika hanya perlu memeriksa nilai tertentu, switch lebih rapi. Namun, jika perlu menangani kondisi kompleks, if lebih sesuai.</p> <p>2.4. Buatlah dokumentasi gambar flowchart dari Latihan 2.1. dan Latihan 2.2! Jawab:</p>		

2.1



2.2



[2.1, 2.2, 2.3] Penyusunan Algoritma dan Kode Program

```

2.1 switch(data) {
    case 'A':
        System.out.print("Anda sudah rajin belajar");
        break; // baris 1
    case 'B':
        System.out.print(" Anda perlu kurangi main game");
        break; // baris 2
    default:
        System.out.print(" Pilihan anda diluar A atau B ");
}
  
```

```

2.2 import java.util.Scanner;
  
```

```

public class SwitchBersarang {
  
```

```

public static void main(String[] args) {
    byte bulan;
    int tahun = 2022;
    int jumlahHari = 0;
    System.out.print("Masukkan data bulan (dalam angka): ");
    Scanner masukData = new Scanner(System.in);
    bulan = masukData.nextByte();

    switch (bulan) {
        case 1: case 3: case 5: case 7: case 8: case 10: case 12: // baris 1
            jumlahHari = 31;
            break;
        case 4: case 6: case 9: case 11: // baris 2
            jumlahHari = 30;
            break;
        case 2:
            if (tahun % 4 == 0 && (tahun % 100 != 0 || tahun % 400 == 0))
                jumlahHari = 29; // Memperbaiki logika tahun kabisat
            else
                jumlahHari = 28;
            break;
        default:
            System.out.println("Maaf bulan hanya sampai 12.");
            break;
    }
    System.out.println("Jumlah hari = " + jumlahHari);
}
}

```

Luaran : Masukkan data bulan (dalam angka): 7
 Jumlah hari = 31

2.3 import java.util.Scanner;

```

public class IfExample {
    public static void main(String[] args) {
        Scanner masukData = new Scanner(System.in);
        System.out.print("Pilih A atau B: ");
        char data = masukData.next().charAt(0);

        if (data == 'A') {
            System.out.print("Anda sudah rajin belajar");
        } else if (data == 'B') {
            System.out.print("Anda perlu kurangi main game");
        } else {
            System.out.print("Pilihan anda diluar A atau B");
        }
    }
}

```

Luaran: Pilih A atau B: A
 Anda sudah rajin belajar

[2.1, 2.2, 2.3, 2.4] Kesimpulan

- Switch:

Lebih mudah dibaca dan lebih terstruktur ketika menangani banyak pilihan tetap.

Dapat lebih cepat dalam beberapa implementasi karena optimasi pencarian.
Ideal untuk perbandingan nilai tetap (seperti karakter atau bilangan bulat).

- If:

Lebih fleksibel dan dapat menangani kondisi kompleks (seperti rentang nilai atau kombinasi logika).

Cocok untuk situasi di mana kondisi tidak hanya berupa kesetaraan.

Mungkin lebih lambat jika banyak kondisi yang perlu diperiksa secara berurutan.

Refleksi: Mendapatkan Pemahaman yang baru

