

Nama & NPM	Topik:	Tanggal:
Michelia Erza Annadhira G1F024035	IF Java	22 September 2024

[1] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

Contoh 1: Salin dan tempel kode program berikut ke Eclipse.

```
import java.util.Scanner; //memanggil impor package yang membaca masukan pengguna

public class PercabanganIf {
    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
        int nilai = masuk.nextByte(); //menyimpan masukan pengguna ke tipe data

        if (nilai == 10) { //percabangan yang memeriksa kondisi
            System.out.println("Sepuluh"); //baris kode yang dieksekusi bila benar
        }
        else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
            System.out.println("Nilai Bukan Sepuluh");
        }
    }
}
```

Luaran Contoh 1:

Masukkan Angka Anda : 8
Nilai Bukan Sepuluh

Contoh 2: Salin dan tempel kode program berikut ke Eclipse.

```
import java.util.Scanner;

public class IfBersarang {
    public static void main(String[] args) {
        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();

        if (nilaiU >= 80) {
            if (nilaiT >= 80) {
                System.out.println("Anda mendapatkan nilai A");
            }
        }
        else{
            System.out.println("Anda TIDAK mendapatkan nilai A");
        }
    }
}
```

Luaran Contoh 2:

Masukkan Angka Tugas Anda : 70
Masukkan Angka Quiz Anda : 70
Masukkan Angka UTS Anda : 70
Anda TIDAK mendapatkan nilai A

Latihan 1:

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)

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.

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!

- IF** ($\text{nilaiU} \geq 80 \parallel \text{nilaiT} \geq 80 \parallel \text{nilaiQ} \geq 80$)
- IF** ($\text{nilaiU} \geq 80 \parallel \text{nilaiT} \geq 80 \&\& \text{nilaiQ} \geq 80$)
- IF** ($\text{nilaiU} \geq 80 \&\& \text{nilaiT} \geq 80 \parallel \text{nilaiQ} \geq 80$)
- IF** ($\text{nilaiU} \geq 80 \&\& \text{nilaiT} \geq 80 \&\& \text{nilaiQ} \geq 80$)

1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!

2) Rincikan sumber informasi yang relevan (buku / webpage)

Video penjelasan pembelajaran dapat diakses pada **Chanel Youtube Rumah Ilmu Raflesia**
<https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMyIw>

Video Materi 1 tentang IF – <https://www.youtube.com/watch?v=G0dfdAFa9iM>

Video Materi 2 tentang SWITCH – <https://www.youtube.com/watch?v=RB4nz4xkisM>

[2] Analisis dan Argumentasi

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)

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan operator modulus (%) untuk membagi angka yang dimasukkan oleh pengguna dengan dengan 2. Kemudian, bias menggunakan percabangan if-else untuk mengecek hasil bagi itu
- Alasan solusi ini karena ketika kita membagi sebuah bilangan dengan 2 dan hasil bagi itu tidak menyisakan angka, maka bilangan tersebut merupakan bilangan genap. Sebaliknya, jika ada sisa, bilangan tersebut merupakan bilangan ganjil.
- Perbaiki kode program dengan cara menambahkan logika pengecekan menggunakan operator modulus (%) dan percabangan if-else seperti berikut :

```
if(nilai % 2 == 0 { //percabangan yan memeriksa kondisi
System.out.println("merupakan bilangan genap"); //baris kode yang dieksekusi
bila benar
}
else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
System.out.println("merupakan bilangan ganjil");
}
```

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.

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan kondisi tambahan di dalam percabangan if untuk memeriksa nilai Quiz. Setelah memeriksa nilai UTS dan Tugas, kita perlu memastikan nilai quiz juga memenuhi syarat agar siswa bias mendapatkan nilai A.
- Alasan solusi ini karena kalau hanya memeriksa nilai UTS dan tugas saja, kita tidak tau secara pasti apakah siswa benar-benar layak mendapatkan nilai A. nilai Quiz juga penting dan harus diperiksa.
- Perbaiki kode program dengan cara menambah satu if lagi di dalam percabangan yang sudah ada untuk memeriksa apakah nilai quiz juga memenuhi syarat.

```

Scanner varU = new Scanner(System.in);
System.out.print("Masukkan Angka UTS Anda : ");
int nilaiU = varU.nextByte();

Scanner varU = new Scanner(System.in);
System.out.print("Masukkan Angka UTS Anda : ");
int nilaiU = varU.nextByte();

Scanner varU = new Scanner(System.in);
System.out.print("Masukkan Angka UTS Anda : ");
int nilaiU = varU.nextByte();

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");
}

```

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!

- `IF (nilaiU >= 80 || nilaiT >= 80 || nilaiQ >= 80)`
 - `IF (nilaiU >= 80 || nilaiT >= 80 && nilaiQ >= 80)`
 - `IF (nilaiU >= 80 && nilaiT >= 80 || nilaiQ >= 80)`
 - `IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)`
- Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah struktur kode yang bersarang menjadi satu kondisi if. Ini membuat logika program lebih sederhana dan lebih mudah dipahami.
 - Alasan solusi ini karena dengan merangkum semua syarat dalam satu baris, kita tidak hanya mempercepat eksekusi program, tetapi juga membuatnya lebih mudah untuk dibaca dan dikelola.
 - Perbaiki kode program dengan cara mengganti semua kondisi if yang bersarang menjadi satu kondisi yang lebih jelas, seperti `if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)`. Dengan cara ini, kita langsung tau apa yang harus dipenuhi agar siswa mendapatkan nilai A.

[3] Penyusunan Algoritma dan Kode Program

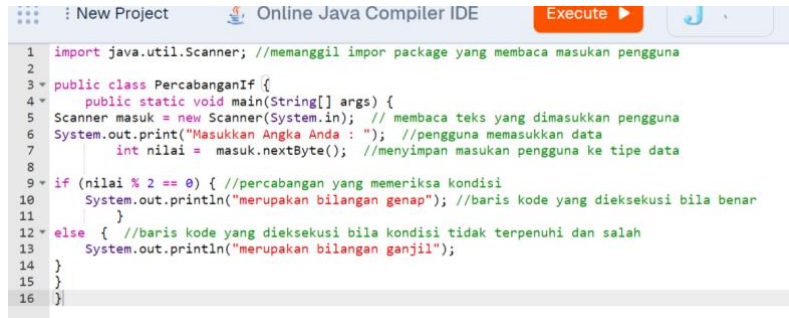
1.1 Bilangan genap merupakan bilangan yang habis dibagi 2. Bilangan ganjil adalah bilangan yang tidak habis dibagi 2. Analisa kode program yang tepat untuk menghitung masukan pengguna termasuk bilangan genap atau bilangan ganjil (lihat Contoh 1)?
(Petunjuk: `hitung = nilai % 2` jika `hitung = 0` maka bilangan genap, jika `hitung = 1` maka bilangan ganjil)

❖ Algoritma

- Mulai
- Kita buat variabel nilai untuk menyimpan angka yang dimasukkan pengguna
- Tampilkan pesan "masukkan angka anda :"
- Gunakan percabangan if-else untuk mengecek apakah angka tersebut genap atau ganjil :
 - If : jika angka tersebut habis dibagi 2, maka program akan menampilkan "bilangan merupakan genap"
 - Else : jika angka tersebut tidak habis dibagi 2, maka program akan

menampilkan “bilangan merupakan ganjil”

5. Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan
6. Simpan kode yang telah diperbaiki dan catat perubahan yang telah dilakukan
7. Selesai



```
1 import java.util.Scanner; //memanggil impor package yang membaca masukan pengguna
2
3 public class PercabanganIf {
4     public static void main(String[] args) {
5         Scanner masuk = new Scanner(System.in); // membaca teks yang dimasukkan pengguna
6         System.out.print("Masukkan Angka Anda : "); //pengguna memasukkan data
7         int nilai = masuk.nextByte(); //menyimpan masukan pengguna ke tipe data
8
9         if (nilai % 2 == 0) { //percabangan yang memeriksa kondisi
10             System.out.println("merupakan bilangan genap"); //baris kode yang dieksekusi bila benar
11         }
12     else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
13         System.out.println("merupakan bilangan ganjil");
14     }
15 }
16 }
```

Luaran(output) :



Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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

❖ Algoritma

1. Mulai
2. Kita buat 3 variabel, nilaiT untuk nilai tugas, nilaiQ untuk nilai quiz, dan nilaiU untuk nilai UTS. Ini akan membantu kita menyimpan nilai-nilai yang dimasukkan oleh siswa
3. Program akan meminta pengguna untuk memasukkan nilai UTS dengan menampilkan pesan “masukkan angka UTS anda : ” dan simpan dalam nilaiU, minta nilai tugas dengan pesan “masukkan angka tugas anda : “ dan simpan dalam nilaiT, dan terakhir minta nilai quiz dengan pesan “masukkan angka quiz anda : “ dan simpan dalam nilaiQ.
4. Kita cek nilai UTS dulu, jika nilaiU lebih besar atau sama dengan 80, kita periksa nilai tugas dan quiz. Jika nilaiT dan nilaiQ juga lebih besar atau sama dengan 80, maka kita tampilkan pesan “Anda mendapatkan nilai A”. jika salah satu dari ketiga variabel itu tidak memenuhi syarat, maka kita tampilkan pesan “Anda TIDAK mendapatkan nilai A”
5. Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan
6. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
7. Selesai

```

1 import java.util.Scanner;
2
3 public class IfBersarang {
4     public static void main(String[] args) {
5         Scanner varU = new Scanner(System.in);
6         System.out.print("Masukkan Angka UTS Anda : ");
7         int nilaiU = varU.nextByte();
8
9         Scanner varT = new Scanner(System.in);
10        System.out.print("Masukkan Angka Tugas Anda : ");
11        int nilaiT = varT.nextByte();
12
13        Scanner varQ = new Scanner(System.in);
14        System.out.print("Masukkan Angka Quiz Anda : ");
15        int nilaiQ = varQ.nextByte();
16
17        if (nilaiU >= 80) {
18            if (nilaiT >= 80) {
19                if (nilaiQ >= 80) {
20                    System.out.println("Anda mendapatkan nilai A");
21                }
22            }
23        }
24        else{
25            System.out.println("Anda TIDAK mendapatkan nilai A");
26        }
27    }
28 }

```

Luaran(output) :

Output	Generated Files	Output	Generated Files
<pre> Masukkan Angka UTS Anda : 90 Masukkan Angka Tugas Anda : 80 Masukkan Angka Quiz Anda : 86 Anda mendapatkan nilai A </pre>		<pre> Masukkan Angka UTS Anda : 78 Masukkan Angka Tugas Anda : 60 Masukkan Angka Quiz Anda : 80 Anda TIDAK mendapatkan nilai A </pre>	
Compiled and executed in 12.26 sec(s)		Compiled and executed in 14.883 sec(s)	

Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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!

- IF (nilaiU >= 80 || nilaiT >= 80 || nilaiQ >= 80)**
- IF (nilaiU >= 80 || nilaiT >= 80 && nilaiQ >= 80)**
- IF (nilaiU >= 80 && nilaiT >= 80 || nilaiQ >= 80)**
- IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)**

❖ Algoritma

- Mulai
- Buat variabel untuk menyimpan nilai-nilai yang akan dimasukkan, yaitu nilai UTS (nilaiU), nilai tugas (nilaiT), dan nilai quiz (nilaiQ)
- Kita akan meminta pengguna untuk memasukkan nilai UTS mereka, setelah pengguna memasukkan nilainya, kita simpan di variabel nilaiU. Lalu, kita minta pengguna memasukkan nilainya tugas, nilai ini kita simpan di variabel nilaiT. terakhir, kita minta pengguna memasukkan nilainya quiz, nilai ini kita simpan di variabel nilaiQ
- Cek syaratnya,
- Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan
- Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan
- Selesai

```

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 (nilaiT >= 80 && nilaiQ >= 80 && nilaiU >= 80) {
18            System.out.println("Anda mendapatkan nilai A");
19        }
20
21        else {
22            System.out.println("Anda TIDAK mendapatkan nilai A");
23        }
24    }
25 }

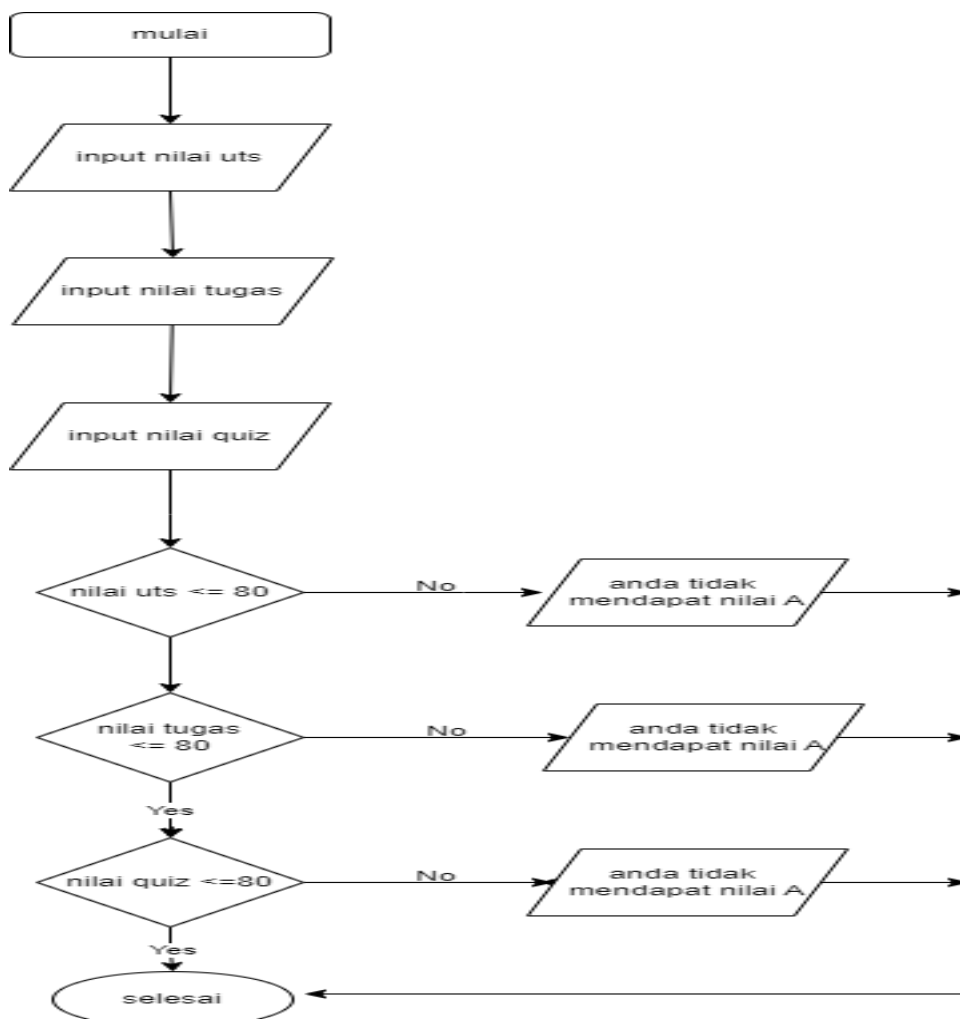
```

Luaran(output) :

Output	Generated Files
<pre> Masukkan Angka Tugas Anda : 90 Masukkan Angka Quiz Anda : 86 Masukkan Angka UTS Anda : 82 Anda mendapatkan nilai A </pre>	<pre> Masukkan Angka Tugas Anda : 70 Masukkan Angka Quiz Anda : 60 Masukkan Angka UTS Anda : 75 Anda TIDAK mendapatkan nilai A </pre>
Compiled and executed in 15.775 sec(s)	Compiled and executed in 15.249 sec(s)

Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

1.4 Uraikan gambar diagram flowchart dari Latihan 1.2!



[4] Kesimpulan

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Permasalahan ini melibatkan 2 hal, pertama menentukan apakah angka genap atau ganjil, kedua apakah nilai UTS, tugas dan quiz memenuhi syarat untuk mendapatkan nilai A.
 - Latihan 1.1, saya menggunakan modulus (%) untuk mengecek apakah angka bias dibagi habis oleh 2. Jika iya, angka tersebut genap, jika tidak, berarti ganjil. Serta, percabangan sederhana if-else digunakan untuk membedakan keduanya.
 - Latihan 1.2, dengan menggunakan if bersarang, program mengecek apakah nilai UTS, tugas, dan quiz semuanya ≥ 80 . Setelah pengecekan nilai quiz ditambahkan, kita bisa memastikan siswa mendapat nilai A jika ketiga syarat terpenuhi.
 - Latihan 1.3, kondisi dapat disederhanakan menggunakan AND (&&) untuk memeriksa semua syarat dalam satu kali pengecekan. Ini membuat program lebih singkat
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
 - Latihan 1.1, karena modulus % langsung menentukan genap atau ganjil dengan cara yang sederhana.
 - Latihan 1.2, karena if bersarang memastikan semua nilai (UTS, tugas, quiz) memenuhi syarat sebelum memberikan hasil.
 - Latihan 1.3, karena && memeriksa semua syarat sekaligus, membuat program lebih ringkas dan efisien

Nama & NPM	Topik:	Tanggal:
Michelia Erza Annadhira G1F024035	SWICTH Java	22 September 2024

[1] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

Contoh 3: Salin dan tempel kode program berikut ke Eclipse.

```
import java.util.Scanner;

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

Luaran Contoh 3:

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

Contoh 4: Salin dan tempel kode program berikut ke Eclipse.

```
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: //baris 1
                jumlahHari = 31;
                break;
            case 4: //baris 2
                jumlahHari = 30;
                break;
            case 2:
                if (tahun % 4 == 0)
                    jumlahHari = 29;
                else
                    jumlahHari = 28;
                break;
            default:
                System.out.println("Maaf bulan hanya sampai 12.");
                break;
        }
        System.out.println("Jumlah hari = " + jumlahHari);
    }
}
```



```
}  
}
```

Luaran Contoh 4:

Masukkan data bulan (dalam angka): 7

Jumlah hari = 31

Latihan 2:

- 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!

2) Rincikan sumber informasi yang relevan (buku / webpage)

Video penjelasan pembelajaran dapat diakses pada **Chanel Youtube Rumah Ilmu Raflesia**
<https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMYIw>

Video Materi 1 tentang IF – <https://www.youtube.com/watch?v=G0dfdAFa9iM>

Video Materi 2 tentang SWITCH – <https://www.youtube.com/watch?v=RB4nz4xkisM>

[2] Analisis dan Argumentasi

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**!
 - Analisis perbedaan hasil luaran ketika kode menggunakan **break**, ketika kode **break** baris 1 dihapus, dan ketika kode **break** baris 2 dihapus.
 - Jika **break** ada di setiap case, program akan mencetak pesan sesuai pilihan yang tersedia.
 - Tanpa **break** setelah case 'A'
hasil ketika input 'A' : "Anda sudah rajin belajar Anda perlu kurangi main game input 'B' dan lainnya tetap sama
 - Tanpa **break** setelah case 'B'
hasil ketika input 'B' : 'Anda perlu kurangi main gamePilihan anda diluar A atau B input 'A' tetap sama
 - Analisis kegunaan baris kode dengan **break** dan kata kunci **default**
 - **Break**, dapat menghentikan eksekusi agar hanya satu case yang dijalankan, mencegah hasil yang tidak diinginkan.
 - **Default**, menangani input yang tidak sesuai dengan case yang ada, memberikan umpan balik yang jelas kepada pengguna
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
 - a. Saya mengusulkan permasalahan ini dengan menambahkan semua bulan dari Januari sampai Desember dalam switch agar program bisa menghitung jumlah hari untuk setiap bulan dengan benar.

- b. Alasan solusi ini karena kode yang ada hanya mencakup sampai bulan ke 4, sehingga belum lengkap. Selain itu, perlu penyesuaian untuk menghitung jumlah hari di bulan februari, terutama ketika tahun kabisat.
 - c. Perbaiki kode program dilakukan dengan menambahkan case untuk semua bulan yang belum ada dan memperbaiki pengecekan tahun kabisat di bulan februari, supaya jumlah hari dihitung dengan lebih akurat
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** !
 - a. Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengganti penggunaan switch-case dengan if-else. Ini karena pilihan yang ada sangat sederhana dan bisa dengan mudah dikelola dengan kondisi logika.
 - b. Alasan solusi ini karena dengan menggunakan if-else, kita bisa membuat kode menjadi lebih jelas dan mudah dibaca. Ketika hanya ada dua pilihan (A dan B), penggunaan if-else membuat alur program lebih langsung, tanpa perlu menggunakan break yang mungkin membingungkan.
 - c. Perbaiki kode program dengan cara mengubah struktur switch-else menjadi if-else. Ini akan membuat program lebih mudah dimengerti.

[3] Penyusunan Algoritma dan Kode Program

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**!
 - ❖ Algoritma
 - a. Mulai
 - b. Buat input (Scanner)
 - c. Tampilkan pesan “pilih A atau B”
 - d. Ambil input karakter dari pengguna
 - e. Gunakan switch untuk memeriksa input :
 - Jika ‘A’, tampilkan “Anda sudah rajin belajar” dan hentikan dengan break
 - Jika ‘B’, tampilkan “Anda perlu kurang main game” dan hentikan dengan break
 - Jika bukan ‘A’ atau ‘B’, tampilkan “Pilihan anda diluar A atau B”
 - f. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
 - g. Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan
 - h. Selesai

```

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         switch(data) {
10             case 'A':
11                 System.out.print("Anda sudah rajin belajar");
12
13             case 'B':
14                 System.out.print(" Anda perlu kurang main game");
15                 break; // baris 2
16             default:
17                 System.out.print(" Pilihan anda diluar A atau B ");
18         }
19     }
20 }

```

```

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         switch(data) {
10             case 'A':
11                 System.out.print("Anda sudah rajin belajar");
12
13             case 'B':
14                 System.out.print(" Anda perlu kurang main game");
15
16             default:
17                 System.out.print(" Pilihan anda diluar A atau B ");
18         }
19     }
20 }

```

Luaran (Output) :



Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran

(output) yang sesuai dengan kode program dan tidak terjadi error.

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

❖ Algoritma

- Mulai
- Buat variabel bulan, tahun, dan jumlahHari
- Minta pengguna memasukkan angka bulan (1-12) [input]
- Periksa bulan :
 - Bulan 1, 3, 5, 7, 8, 10, 12 [jumlahHari = 31]
 - Bulan 4, 6, 9, 11 [jumlahHari = 30]
 - Bulan 2 [kabisat, jumlahHari = 29, bukan jumlahHari = 28]
- Jika bulan di luar 1-12, tampilkan pesan error
- [output] tampilkan jumlah hari sesuai bulan
- Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
- Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan
- Selesai

```
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    }
32 }
```

Luaran (Output) :



Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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** !

❖ Algoritma

- Mulai
- Tampilkan pesan “Pilih A atau B”
- Input pengguna dan simpan di data
- Jika data == ‘A’
 - Tampilkan [“Anda sudah rajin belajar”]
- Jika tidak, jika data == ‘B’
 - Tampilkan [“Anda perlu kurang main game”]
- Jika tidak :

- Tampilkan ["Pilihan anda diluar A atau B"]
- g. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
- h. Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan
- i. Selesai

```

1 import java.util.Scanner;
2
3 public class IfBersarang {
4     public static void main(String[] args) {
5         Scanner masukData = new Scanner(System.in);
6
7         // Mengambil input
8         System.out.print("Pilih A atau B : ");
9         char data = masukData.next().charAt(0);
10
11         // Gunakan struktur if-else mengganti switch
12         if (Character.toUpperCase(data) == 'A') {
13             System.out.println("Anda sudah rajin belajar");
14         } else if (Character.toUpperCase(data) == 'B') {
15             System.out.println("Anda perlu kurangi main game");
16         } else {
17             System.out.println("Pilihan anda diluar A dan B");
18         }
19     }
20 }

```

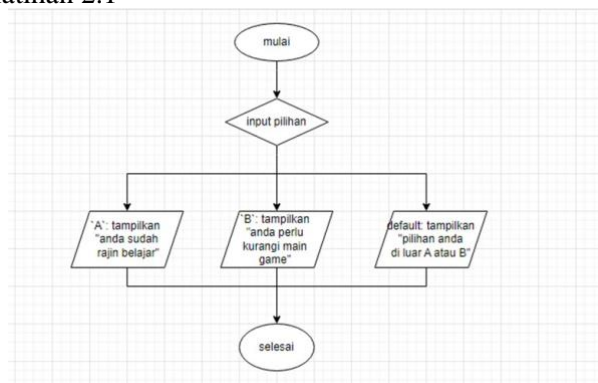
Luaran (Output) :



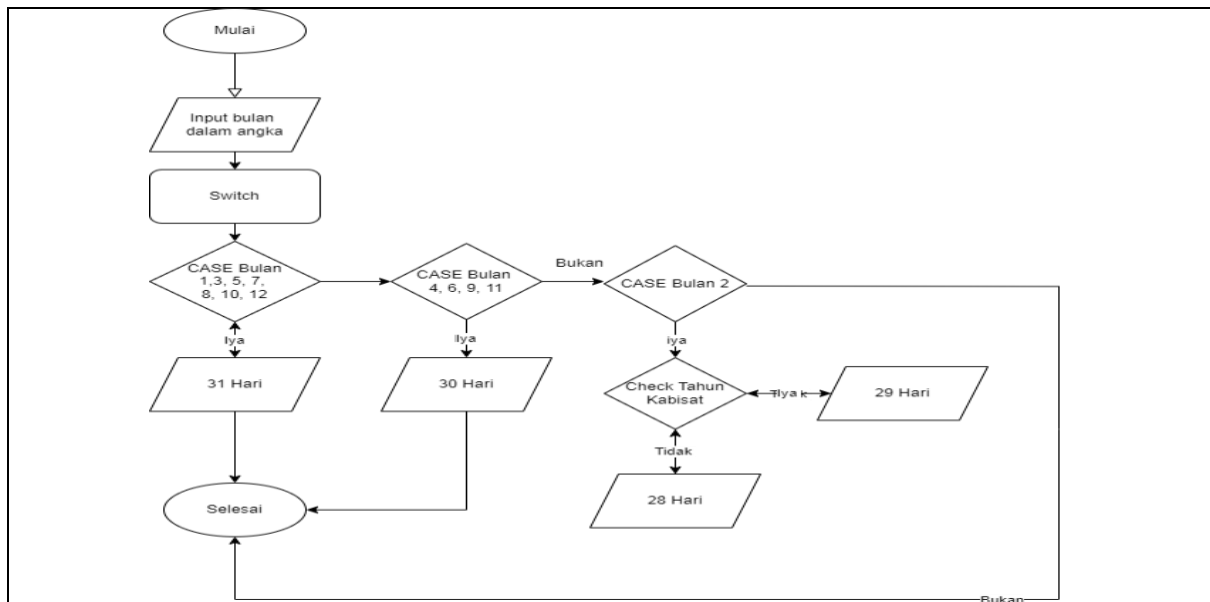
Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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

❖ Flowchart latihan 2.1



❖ Flowchart latihan 2.2



[4] Kesimpulan

- Pada contoh 3, saya menggunakan struktur control switch-case karena memudahkan dalam menentukan respon berdasarkan pilihan pengguna, yaitu A dan B. Perbaiki program dengan menambahkan pernyataan break pada setiap case struktur Java mengharuskan penggunaan break untuk mencegah “fall-through” ke case berikutnya, yang bisa menghasilkan output yang tidak diinginkan.
- Pada contoh 4, saya menggunakan struktur control switch-case karena memudahkan dalam menghitung jumlah hari berdasarkan bulan yang dimasukkan oleh pengguna. Perbaiki program dengan menambahkan case untuk bulan 5 sampai 12 karena struktur Java mengharuskan penanganan semua agar program dapat memberikan hasil yang akurat dan lengkap.

Refleksi

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)

Minggu ini, saya belajar banyak tentang percabangan dalam Java, khususnya penggunaan if dan switch untuk mengevaluasi input pengguna. Saya menghadapi tantangan dalam mengatur kondisi bersarang dan memahami pentingnya pernyataan break. Proses merangkum beberapa kondisi menjadi satu juga mengasah kemampuan berpikir kritis saya. Secara keseluruhan, pengalaman ini meningkatkan pemahaman dan kepercayaan diri saya dalam pengendalian alur pemrograman.