

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>HANIFAH AZIZAH G1F024037</b>	<b>IF DAN SWITCH JAVA</b>	<b>21/09/2024</b>

### [No.1] Identifikasi Masalah:

#### 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: hitung = nilai % 2 jika hitung = 0 maka bilangan genap, jika 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 (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)**

1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!

### [No.1] Analisis dan Argumentasi

1.1 Kode program yang tepat untuk masalah pada nomor satu adalah sebagai berikut:

```
import java.util.Scanner;
public class Percabangan {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Masukkan Angka Anda : ");
        int nilai = scanner.nextByte();
        int hitung = nilai % 2;

        if (hitung == 0) {
            System.out.println("Bilangan Genap");
        }
        else {
            System.out.println("Bilangan Ganjil");
        }
    }
}
```

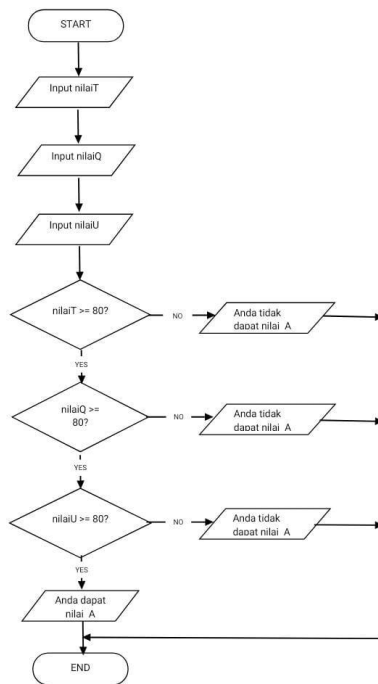
Dengan menambahkan variabel hitung menggunakan tipe data int sehingga variabel tersebut menyimpan rumus operasi perhitungan nilai % 2. Jadi Ketika pengguna memasukkan suatu nilai di luaran, kode akan otomatis memproses menggunakan rumus yang telah disimpan dalam variabel dan memutuskan kondisi apa yang akan dieksekusi. Jika pengguna memasukkan nilai genap, maka program akan membacanya sebagai angka yang habis dibagi 2 (bersisa 0) dan menampilkan luaran "bilangan genap". Begitupula sebaliknya.

1.2 Pada masalah nomor 2, diharuskan menambah satu lagi kondisi IF di mana ketika nilai dimasukkan maka harus memenuhi >= 80 untuk semua kondisi. Jadi jika pengguna ingin mendapatkan luaran "mendapatkan nilai A" maka harus memasukkan nilai >= 80 untuk semua kondisi: \*kode dapat dilihat pada bagian penyusunan algoritma dan kode program. Dengan ditambahkannya satu lagi kondisi IF bersarang, kini terdapat 3 kondisi IF sehingga pengguna perlu memasukkan 3 kondisi untuk mendapatkan luaran.

1.3 Kondisi yang paling tepat untuk menggantikan 3 IF bersarang tersebut adalah yang d. **IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80).**

Karena sesuai dengan syarat pada Latihan 1.2, siswa harus memiliki nilai >= 80 untuk semua Tugas, UTS, dan Quiz baru bisa mendapatkan nilai A. Jika menggunakan kondisi a, b, c maka siswa masih bisa mendapatkan nilai A walaupun salah satu dari kondisi tidak memenuhi nilai >= 80 atau bahkan bisa mendapatkan nilai A walaupun hanya memiliki satu nilai >= 80.

1.4 Kode dari Latihan 1.2 dapat diimplementasikan menjadi gambar flowchart sebagai berikut:



Flowchart nilai UTS, Tugas, Quiz dengan word

### [No.1] Penyusunan Algoritma dan Kode Program

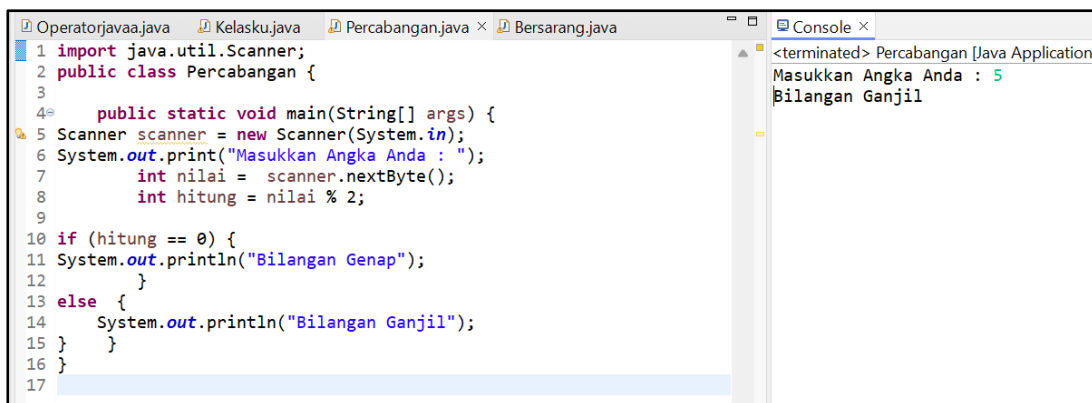
#### Algoritma Latihan 1.1

1. Mulai
2. Deklarasikan dengan membuat objek Scanner untuk input dari pengguna.
3. Tampilkan pesan "Masukkan Angka Anda : ".
4. Simpan input pengguna dalam variabel nilai.
5. Hitung sisa bagi dari nilai dengan 2, simpan hasilnya dalam variabel hitung.
6. Jika hitung sama dengan 0: Tampilkan "Bilangan Genap".
7. Jika tidak: Tampilkan "Bilangan Ganjil".
8. Selesai.

```
1 import java.util.Scanner;
2 public class Percabangan {
3
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Masukkan Angka Anda : ");
7         int nilai = scanner.nextByte();
8         int hitung = nilai % 2;
9
10        if (hitung == 0) {
11            System.out.println("Bilangan Genap");
12        }
13        else {
14            System.out.println("Bilangan Ganjil");
15        }
16    }
17 }
```

Console output: <terminated> Percabangan [Java Applic  
Masukkan Angka Anda : 10  
Bilangan Genap

### Kode Java dengan luaran bilangan genap untuk bilangan yang habis dibagi 2.



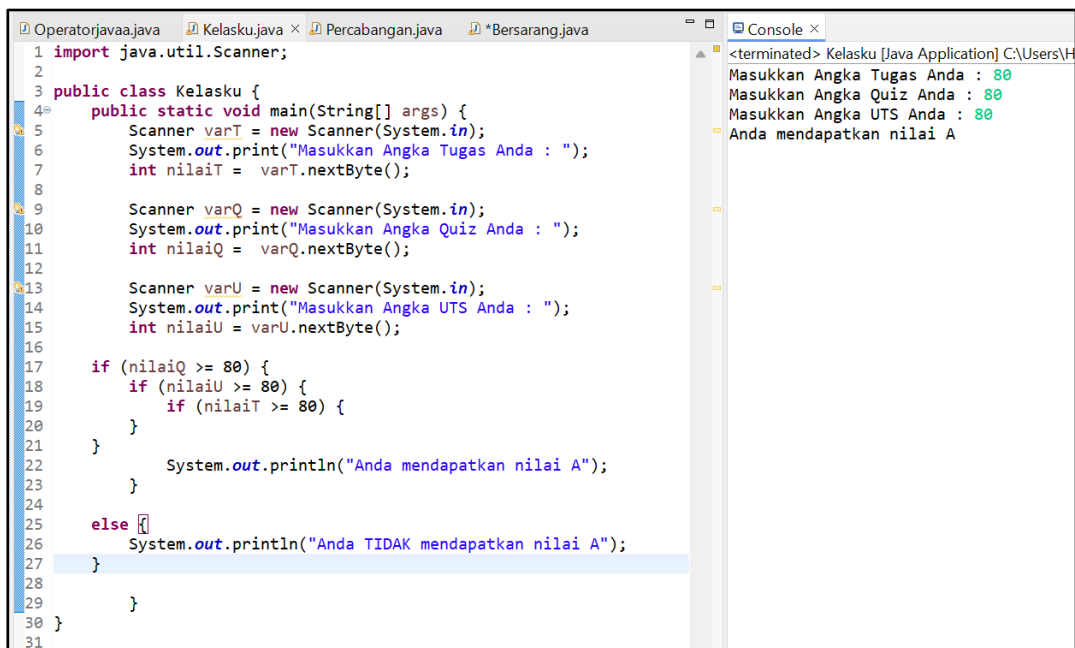
```
1 import java.util.Scanner;
2 public class Percabangan {
3
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Masukkan Angka Anda : ");
7         int nilai = scanner.nextByte();
8         int hitung = nilai % 2;
9
10        if (hitung == 0) {
11            System.out.println("Bilangan Genap");
12        }
13        else {
14            System.out.println("Bilangan Ganjil");
15        }
16    }
17 }
```

Console: <terminated> Percabangan [Java Application]  
Masukkan Angka Anda : 5  
Bilangan Ganjil

### Kode Java dengan luaran bilangan ganjil untuk bilangan yang tidak habis dibagi 2.

#### Algoritma 1.2

1. Mulai
2. Deklarasikan dengan membuat objek Scanner untuk input dari pengguna.
3. Tampilkan pesan "Masukkan Angka Tugas Anda : ".
4. Simpan input pengguna dalam variabel nilaiT.
5. Tampilkan pesan "Masukkan Angka Quiz Anda : ".
6. Simpan input pengguna dalam variabel nilaiQ.
7. Tampilkan pesan "Masukkan Angka UTS Anda : ".
8. Simpan input pengguna dalam variabel nilaiU.
9. Cek Nilai untuk A: Jika nilaiQ >= 80; Jika nilaiU >= 80; Jika nilaiT >= 80:
10. Jika semua nilai memenuhi, Tampilkan "Anda mendapatkan nilai A".
11. Jika tidak, Tampilkan "Anda TIDAK mendapatkan nilai A".
12. Selesai.



```
1 import java.util.Scanner;
2
3 public class Kelasku {
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 (nilaiQ >= 80) {
18            if (nilaiU >= 80) {
19                if (nilaiT >= 80) {
20                }
21            }
22            System.out.println("Anda mendapatkan nilai A");
23        }
24        else {
25            System.out.println("Anda TIDAK mendapatkan nilai A");
26        }
27    }
28 }
29
30 }
31 }
```

Console: <terminated> Kelasku [Java Application] C:\Users\H...  
Masukkan Angka Tugas Anda : 80  
Masukkan Angka Quiz Anda : 80  
Masukkan Angka UTS Anda : 80  
Anda mendapatkan nilai A

### Kode Java dengan luaran Anda Mendapatkan nilai A jika semua kondisi >= 80 terpenuhi.

```
Operatorjavaa.java Kelasku.java Percabangan.java *Bersarang.java Console x
1 import java.util.Scanner;
2
3 public class Kelasku {
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 (nilaiQ >= 80) {
18            if (nilaiU >= 80) {
19                if (nilaiT >= 80) {
20                }
21            }
22            System.out.println("Anda mendapatkan nilai A");
23        }
24        else {
25            System.out.println("Anda TIDAK mendapatkan nilai A");
26        }
27    }
28
29 }
30
31
```

```
<terminated> Kelasku [Java Application] C:\Users\HP
Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 70
Masukkan Angka UTS Anda : 80
Anda TIDAK mendapatkan nilai A
```

Kode Java dengan luaran Anda Tidak Mendapatkan nilai A jika salah satu kondisi  $\geq 80$  tidak terpenuhi.

### Algoritma Latihan 1.3

1. Mulai
2. Deklarasikan dengan membuat objek Scanner untuk input dari pengguna.
3. Tampilkan pesan "Masukkan Angka Tugas Anda : ".
4. Simpan input pengguna dalam variabel nilaiT.
5. Tampilkan pesan "Masukkan Angka Quiz Anda : ".
6. Simpan input pengguna dalam variabel nilaiQ.
7. Tampilkan pesan "Masukkan Angka UTS Anda : ".
8. Simpan input pengguna dalam variabel nilaiU.
9. Cek Nilai untuk A: Jika nilaiQ  $\geq 80$  dan nilaiU  $\geq 80$  dan nilaiT  $\geq 80$ :
10. Tampilkan "Anda mendapatkan nilai A".
11. Jika tidak: Tampilkan "Anda TIDAK mendapatkan nilai A".
12. Selesai.

```
Operatorjavaa.java Kelasku.java Percabangan.java Bersarang.java Console x
1 import java.util.Scanner;
2 public class Bersarang {
3
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 (nilaiQ >= 80 && nilaiU >= 80 && nilaiT >= 80) {
18            System.out.println("Anda mendapatkan nilai A");
19        }
20        else {
21            System.out.println("Anda TIDAK mendapatkan nilai A");
22        }
23    }
24
25 }
```

```
<terminated> Bersarang [Java Application] C:\Users\H
Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 70
Masukkan Angka UTS Anda : 90
Anda TIDAK mendapatkan nilai A
```

Kode Java dengan luaran Anda Tidak Mendapatkan nilai A jika salah satu kondisi  $\geq 80$  tidak terpenuhi dengan IF bersarang digabungkan menjadi satu kondisi dengan operator  $\&\&$ .

```

1 import java.util.Scanner;
2 public class Bersarang {
3
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 (nilaiQ >= 80 && nilaiU >= 80 && nilaiT >= 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 }

```

Console Output:

```

<terminated> Bersarang [Java Application] C:\Users
Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 80
Masukkan Angka UTS Anda : 80
Anda mendapatkan nilai A

```

Kode Java dengan luaran Anda Mendapatkan nilai A jika semua kondisi  $\geq 80$  terpenuhi dengan IF bersarang digabungkan menjadi satu kondisi dengan operator  $\&\&$ .

#### [No.1] Kesimpulan

1. Kesimpulan: Saya harus dapat memodifikasi kode yang sudah ada dengan perintah yang serupa tetapi menginginkan luaran yang berbeda. Seperti permasalahan pada Latihan 1.1 dan Latihan 1.2. Kemudian tujuan penyusunan kode harus diketahui secara matang agar dapat menentukan modifikasi apa yang tepat seperti pada Latihan 1.3. Kemampuan untuk menganalisa kode juga diperlukan dalam pembuatan flowchart
2. Dasar pengambilan Keputusan: Dasar pengambilan Keputusan saya adalah materi IF SWITCH yang telah diberikan.

Nama & NPM

Topik:

Tanggal:

HANIFAH AZIZAH G1F024037	IF DAN SWITCH JAVA	21/09/2024
-----------------------------	--------------------	------------

## [No.2] Identifikasi Masalah:

### 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!

## [No.2] Analisis dan Argumentasi

- 2.1 Ketika kode `break` pada baris 1 dihapus, kode tetap berjalan tanpa adanya error. Tetapi ketika pengguna memasukkan nilai A, luaran yang ditampilkan menampilkan luaran untuk case A dan juga luaran untuk case B. Sehingga luaran tidak lagi sesuai dengan case yang diinginkan. Sama halnya ketika kode `break` pada baris 2 dihapus, ketika pengguna memasukkan nilai A luaran yang ditampilkan menampilkan luaran untuk case A, case B, dan default. Kode memang berjalan tanpa error tetapi terjadi ketidakteraturan pada luaran.  
Untuk itu diketahui bahwa kegunaan kode `break` adalah untuk memisahkan atau menghentikan switch statement antar case yang tersedia. Jika kode `break` dihapus, maka program akan terus menjalankan luaran dari case berikutnya. Sedangkan dalam switch statement, default berfungsi ketika terdapat suatu kondisi yang tidak cocok dengan case yang ditentukan. Jika tidak ada case yang cocok dengan kondisi yang diberikan, maka pernyataan default akan dieksekusi.
- 2.3 Permasalahan pada contoh 3 dapat diselesaikan dengan perintah `IF` dengan tetap menghasilkan luaran yang sama.

The screenshot shows a Java IDE with a file named 'Bulan.java'. The code is as follows:

```

1 import java.util.Scanner;
2 public class Breakremove {
3     public static void main(String[] args) {
4         Scanner masukData = new Scanner(System.in);
5
6         System.out.print("Pilih A atau B atau Z : ");
7         char data = masukData.next().charAt(0);
8
9
10        if (data == 'A') {
11            System.out.println("Anda Sudah Rajin Belajar");
12        } else if (data == 'B') {
13            System.out.println("Anda Perlu Kurangi Main Game");
14        } else if (data == 'Z') {
15            System.out.println("Anda Perlu Makan");
16        } else {
17            System.out.println("Pilihan Anda di luar A atau B");
18        }
19    }
20 }
21

```

The console output on the right shows the execution result for input 'Z':

```

<terminated> Breakremove [Java Application]
Pilih A atau B atau Z : Z
Anda Perlu Makan

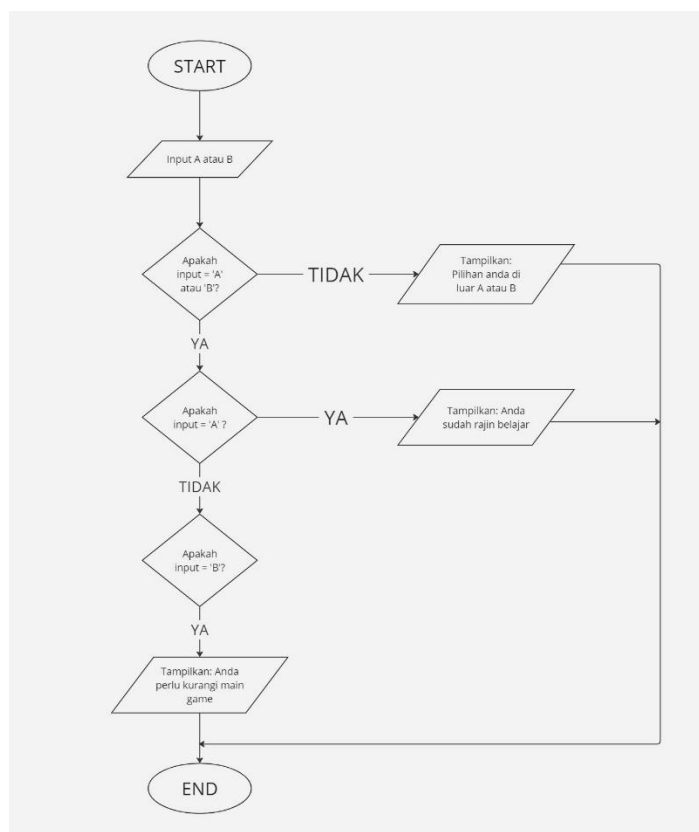
```

Ketika menggunakan perintah switch, maka perlu menetapkan case-case yang relevan dan luaran sesuai dengan case yang ditetapkan, selain itu perlu menambahkan kode break agar setiap case tidak tercampur dengan case satu sama lain. Jika perintah switch menggunakan banyak case untuk banyak kondisi, perintah IF dapat menggunakan sebanyak else if untuk banyak kondisi.

Permasalahan yang dapat diselesaikan dengan perintah IF juga lebih fleksibel karena dapat menangani kondisi kombinasi logika dan evaluasi Boolean. Tetapi switch lebih efisien untuk menyelesaikan banyak kondisi yang dibandingkan.

## [No.2] Penyusunan Algoritma dan Kode Program

### Flowchart Latihan 2.1



Flowchart untuk kode case break dan default yang dibuat di website miro.

### Algoritma Latihan 2.2

1. Mulai
2. Inisialisasi 7variable bulan bertipe byte
3. Inisialisasi 7variable tahun bertipe int dan set nilai awalnya ke 2024.
4. Inisialisasi 7variable jumlahHari bertipe int dan set nilainya ke 0.
5. Tampilkan pesan "Masukkan data bulan (dalam angka):".
6. Ambil input pengguna dan simpan ke dalam 7variable bulan.
7. Gunakan struktur switch untuk memeriksa nilai bulan: Kasus 1, 3, 5, 7, 8, 10, 12: Set jumlahHari menjadi 31.
8. Kasus 4, 6, 9, 11: Set jumlahHari menjadi 30.
9. Kasus 2: Cek apakah tahun habis dibagi 4.
10. Jika ya, set jumlahHari menjadi 29 (tahun kabisat).
11. Jika tidak, set jumlahHari menjadi 28.
12. Kasus Default: Tampilkan pesan "Maaf bulan hanya sampai 12.".
13. Tampilkan jumlah hari dengan pesan "Jumlah hari = " diikuti dengan nilai jumlahHari.

```

1 import java.util.Scanner;
2
3 public class Bulan {
4     public static void main(String[] args) {
5         byte bulan;
6         int tahun = 2024;
7         int jumlahHari = 0;
8         System.out.print("Masukkan data bulan (dalam angka): ");
9         Scanner masukData = new Scanner(System.in);
10        bulan = masukData.nextByte();
11
12        switch (bulan) {
13            case 1: case 3: case 5: case 7: case 9: case 11:
14                jumlahHari = 31;
15                break;
16            case 4: case 6: case 8: case 10: case 12:
17                jumlahHari = 30;
18                break;
19            case 2:
20                if (tahun % 4 == 0)
21                    jumlahHari = 29;
22                else
23                    jumlahHari = 28;
24                break;
25            default:
26                System.out.println("Maaf bulan hanya sampai 12.");
27                break;
28        }
29        System.out.println("Jumlah hari = " + jumlahHari);
30    }
31 }

```

Console x

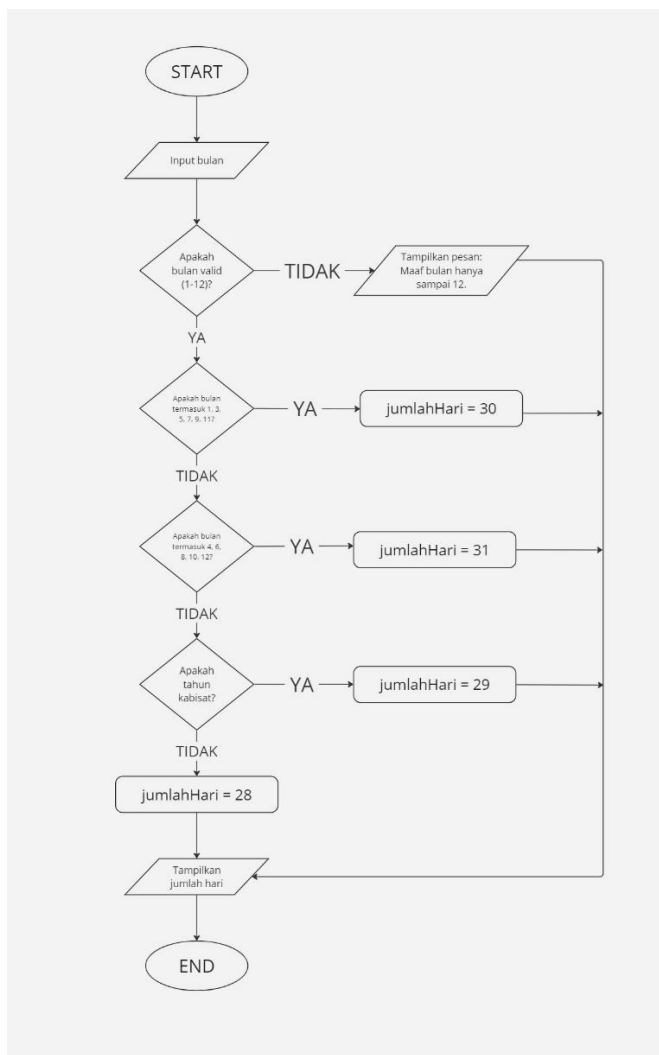
```

<terminated> Bulan [Java Application] C:\Users\HP\p2\pool\plugi
Masukkan data bulan (dalam angka): 12
Jumlah hari = 30

```

Kode Java dengan perintah witch yang memuat 12 case untuk jumlah hari setiap bulannya dan memasukkan perintah IF ELSE untuk case 2, bulan februari sesuai tahun kabisat.

## Flowchart Latihan 2.2



Flowchart kode case penentuan jumlah hari untuk case 12 bulan menggunakan website miro.



<b>[No.2] Kesimpulan</b>
<ol style="list-style-type: none"><li>1. Kesimpulan: Dapat disimpulkan jika kode break sangat penting untuk perintah SWITCH agar masing-masing pernyataan dari masing-masing case tidak saling bertabrakan pada luaran. Kode Default juga penting untuk mengecek apabila input yang dimasukkan pengguna tidak sesuai dengan case yang ditentukan. Perintah SWITCH dapat diganti dengan perintah IF ELSE. Masing-masing perintah SWITCH dan IF sama bergunanya untuk berbagai masalah yang memiliki prioritas berbeda.</li><li>2. Dasar pengambilan Keputusan: Dasar pengambilan Keputusan saya adalah materi SWITCH, BREAK, DEFAULT, IF ELSE yang telah diberikan dan implementasi pada flowchart.</li></ol>

## **Refleksi**

Dikarenakan tugas dikerjakan tanpa adanya tekanan waktu, saya dapat lebih memahami apa yang dibutuhkan oleh soal dan dapat lebih menganalisa jawaban dari yang saya buat. Saya juga menemukan cara-cara baru yang lebih efektif dan rapi dalam Menyusun jawaban untuk setiap soal yang diberikan. Materi kali ini sangat membuka wawasan dan pengetahuan saya dalam Menyusun kode program yang bermanfaat bagi pengguna. Tentu saya perlu mempelajari lebih dalam tentang kode-kode dan variasi perintah agar dapat mengimplementasikannya lebih efektif.

Tantangan yang dihadapi berupa kemampuan Analisa dan menguraikan soal sangat diuji karena soal-soal yang diberikan menjurus pada studi kasus dan bagaimana suatu kondisi dapat memengaruhi luaran yang akan ditampilkan.

Pada minggu ini saya belajar tentang kode operasi nilai dengan input pengguna, IF bersarang, kode logika `&&` dan `||`, kode break dan default dalam SWITCH, mengganti SWITCH dengan IF ELSE, dan flowchart