

Contoh Jawaban:

| Nama & NPM                   | Topik:      | Tanggal:          |
|------------------------------|-------------|-------------------|
| Yulinda Samosir<br>G1F024020 | If dan Else | 25 September 2024 |

### Latihan 1. IF

#### [No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel
  - 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 1.2, 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!
    - a.  $\text{IF} (\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80)$
    - b.  $\text{IF} (\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \ \&\& \ \text{nilaiQ} \geq 80)$
    - c.  $\text{IF} (\text{nilaiU} \geq 80 \ \&\& \ \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80)$
    - d.  $\text{IF} (\text{nilaiU} \geq 80 \ \&\& \ \text{nilaiT} \geq 80 \ \&\& \ \text{nilaiQ} \geq 80)$
  - 1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!

#### [No.1] Analisis dan Argumentasi

- 1) analisis kode program pada contoh 1 bertujuan untuk menentukan apakah bilangan yang dimasukkan oleh pengguna termasuk bilangan genap atau ganjil. Pengguna diminta untuk memasukkan sebuah angka, disini saya masukkan angka 6. Lalu program akan memeriksa apakah angka tersebut habis dibagi 2 menggunakan angka  $\% 2$ . Jika hasil angka  $\% 2 == 0$ , maka bilangan tersebut adalah genap. Jika hasil angka  $\% 2 != 0$ , maka bilangan tersebut adalah ganjil. Karena disini saya masukkan angka 6, maka 6 adalah bilangan genap (karena  $6 \% 2 = 0$ ). Jadi, apakah bilangan tersebut genap atau ganjil akan bergantung pada input yang dimasukkan oleh pengguna.
- 2) Pada Contoh 2, terdapat kondisi if bersarang untuk mengecek apakah nilai tugas (nilaiT) dan nilai UTS (nilaiU) dan nilai Quiz (nilaiQ) lebih besar atau sama dengan 80. Jika iya, maka siswa mendapatkan nilai A. Disini saya masukkan nilai Tugas 82, nilai UTS 85, dan nilai Quiz 85. Karena disini ketiga nilainya terpenuhi, maka mencetak siswa mendapatkan nilai A.
- 3) Kita bisa menyederhanakan ketiga kondisi if ini menjadi satu if dengan menggunakan operator logika  $\&\&$  (dan). Dengan  $\&\&$ , kita menyatakan bahwa semua syarat harus benar untuk menghasilkan nilai true. Jadi Pilihan if yang paling tepat untuk menyederhanakan tiga kondisi if tersebut adalah: **d.**  $\text{if} (\text{nilaiU} \geq 80 \ \&\& \ \text{nilaiT} \geq 80 \ \&\& \ \text{nilaiQ} \geq 80)$ . Karena pilihan ini memastikan bahwa ketiga nilai (nilaiU, nilaiT, dan nilaiQ) harus lebih besar atau sama dengan 80 untuk mendapatkan nilai A.

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

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (a) Input nilai UTS, nilai Tugas, dan nilai Quiz

- (b) Jika nilai UTS  $\geq 80$ :
- (c) Lanjut ke langkah 4
- (d) Jika tidak, tampilkan "Nilai tidak cukup untuk A"
- (e) Jika nilai Tugas  $\geq 80$ :
- (f) Lanjut ke langkah 5
- (g) Jika tidak, tampilkan "Nilai tidak cukup untuk A"
- (h) Jika nilai Quiz  $\geq 80$ :
- (i) Tampilkan "Siswa mendapat nilai A"
- (j) Jika tidak, tampilkan "Nilai tidak cukup untuk A"
- (k) Selesai

## 2) Kode program dan luaran

```

1 import java.util.Scanner; //mengambil 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         int hitung = nilai % 2;
9
10        if (hitung == 0) { // jika sisa bagi 2 sama dengan 0
11            System.out.println("Bilangan Genap");
12        }
13        else { // jika sisa bagi 2 sama dengan 1
14            System.out.println("Bilangan Ganjil");
15        }
16
17        if (nilai % 2 == 0) { //percabangan yang memeriksa kondisi
18            System.out.println("0"); //baris kode yang dieksekusi bila benar
19        }
20        else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
21            System.out.println("1");
22        }
23    }
}

```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Output Generated Files

Masukkan Angka Anda:6  
Bilangan Genap

Compiled and executed in 13.435 sec(s)

codingan 1.2

```

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

```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

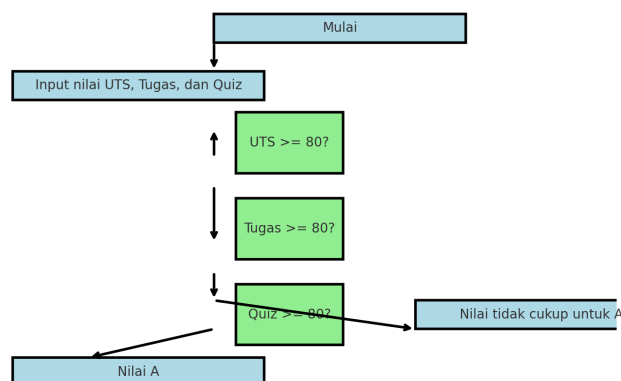
Output Generated Files

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

Compiled and executed in 21.243 sec(s)

Codingan 1.2

Flowchart Latihan 1.2: Penilaian Siswa Berdasarkan Nilai



- a) Analisa luaran yang dihasilkan  
Luaran yang dihasilkan sudah sesuai dengan program yang disusun.  
Kondisi If yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

### [No.1] Kesimpulan

#### Analisa

- a) Susunlah kesimpulan!  
Kesimpulan dari semua kondisi if yang telah dibahas adalah
  1. percabangan if sederhana  
Percabangan ini di gunakan untuk membuat keputusan berdasarkan satu kondisi. Percabangan if sederhana memungkinkan program menjalankan kode blok tertentu, misalnya jika kondisi true dan kode blok lain jika false.
  2. Percabangan if Bersarang  
Percabangan ini digunakan untuk keputusan kompleks di mana satu kondisi tergantung pada kondisi lain. If bersarang melakukan pengecekan berlapis, di mana sebuah kondisi bisa dievaluasi jika kondisi sebelumnya terpenuhi. Jika salah satu kondisi tidak terpenuhi maka kondisi lain tidak bisa di evaluasi. Pada contoh 2 di atas, misalkan disana salah satu kondisi nilai Quiz tidak memenuhi syarat, maka siswa tersebut tidak akan mendapatkan nilai A, karena konsep if bersarang itu semua syarat harus terpenuhi.
  3. Penyederhanaan If Bersarang  
Penyederhanaan ini membuat kode lebih ringkas dan mudah dibaca. Pada contoh 2 tadi, menyederhanakan ketiga kondisi if bersarang supaya menjadi satu menggunakan operator logika && (dan). Operator && memastikan bahwa semua kondisi harus terpenuhi untuk menjalankan blok if.  
**Note:** Untuk kasus di mana semua kondisi harus benar, gunakan &&. Jika hanya salah satu kondisi yang harus benar, gunakan ||.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  
Alasannya supaya bisa menganalisis kondisi dari masing masing kode program yang dibuat dan dijalankan seperti pada contoh contoh di atas yang telah saya bahas.

### Latihan 2. ELSE

#### [No. 1] Identifikasi Masalah:

Uraikan permasalahan dan variabel!

##### 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

Note: break digunakan untuk menghentikan eksekusi switch setelah sebuah case selesai. Jika break dihapus, switch akan terus mengeksekusi kode di case berikutnya meskipun case tersebut tidak sesuai.

## 2.1 Analisis menghapus break pada contoh 3

- Jika input data adalah 'A', maka akan mencetak: Anda sudah rajin belajar Anda perlu kurangi main game. Hal ini karena tanpa break, program tidak berhenti di case 'A' dan melanjutkan ke case 'B'.
- Jika input data adalah 'B', maka akan mencetak: Anda perlu kurangi main game Pilihan anda diluar A atau B. Hal ini karena tanpa break, program tidak berhenti di case 'B' dan melanjutkan ke default.

Kegunaan kode break dan kata kunci default adalah break itu digunakan untuk menghentikan eksekusi switch setelah satu case selesai, sedangkan default menangani semua input yang tidak sesuai dengan case yang ada.

## 2.2 Menambahkan kode program hingga bulan ke-12 pada baris ke -1 dan baris ke-2

Penambahan Kode untuk Bulan Mei hingga Desember:

Bulan dengan 31 hari: Januari, Maret, Mei, Juli, Agustus, Oktober, Desember.

Bulan dengan 30 hari: April, Juni, September, November.

Februari: 28 hari pada tahun biasa, 29 hari pada tahun kabisat.

### **Bulan dengan 31 hari**

- Januari (case 1), Maret (case 3), Mei (case 5), Juli (case 7), Agustus (case 8), Oktober (case 10), dan Desember (case 12) semuanya memiliki 31 hari. Semua kode ini diatur pada satu blok yang sama.

### **Bulan dengan 30 hari**

- April (case 4), Juni (case 6), September (case 9), dan November (case 11) semuanya memiliki 30 hari. Semua kode ini diatur pada blok yang sama.

### **Bulan dengan 28 hari pada tahun biasa, dan 29 hari pada tahun kabisat**

- Februari (case 2). Nah untuk bulan februari, diperlukan pemeriksaan apakah tahun tersebut adalah tahun kabisat atau tidak, dengan cara; jika tahun bisa di bagi 4 ( $\text{tahun} \% 4 == 0$ ) dan bisa di bagi 400 ( $\text{tahun} \% 400 == 0$ ) maka ini adalah tahun kabisat, yang artinya februari memiliki 29 hari. Namun jika tahun tidak bisa di bagi 4 dan tidak bisa di bagi 400, maka ini adalah bukan tahun kabisat yang artinya februari memiliki 28 hari

## 2.3

Switch pada contoh 3 bisa diubah menjadi if, karena hal tersebut sama sama berguna untuk memeriksa beberapa variabel, meskipun switch lebih disukai jika kondisi yang diperiksa adalah nilai konstan tertentu seperti angka atau karakter, sedangkan if-else if lebih fleksibel untuk menangani berbagai jenis ekspresi logika.

### **• Pemeriksaan bulan dengan 31 hari**

Hal ini mengecek apakah bulan bernilai 1, 3, 5, 7, 8, 10, atau 12. Jika iya, variabel jumlahHari diatur ke 31.

### **• Pemeriksaan bulan dengan 30 hari**

Hal ini mengecek apakah bulan bernilai 4, 6, 9, atau 11. Jika iya, jumlahHari diatur ke 30.

- **Pemeriksaan bulan Februari**

Hal ini mengecek apakah bulan ini bernilai 2. Jika benar, maka dilakukan pengecekan apakah tahun adalah tahun kabisat atau tidak:

- **Default**

Jika input bulan tidak valid (lebih dari 12 atau kurang dari 1), maka cetak pesan "Maaf bulan hanya sampai 12".

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

### 1). Algoritma

#### Algoritma 2.1

- a. Input kode switch dengan beberapa case.
- b. Eksekusi kode dengan break pada setiap case.
- c. Hapus break pada baris 1.
- d. Eksekusi kembali
- e. Hapus break pada baris 2.
- f. Eksekusi kembali

#### Algoritma 2.2

- a. Mulai

- a. deklarasikan variabel untuk menyimpan input bulan.
- b. memasukkan nomor bulan (1-12).
- c. gunakan struktur switch untuk menentukan jumlah hari:

- Jika bulan = 1 (Januari): Output "31 hari"

- Jika bulan = 2 (Februari): Output "28 hari" (atau "29 hari" untuk tahun kabisat)

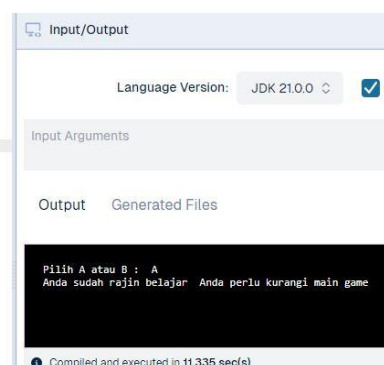
dan seterusnya

- d. selesai

### 2). Kode program dan luaran

```
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");
            case 'B':
                System.out.print(" Anda perlu kurangi main game");
                break; // baris 2
            default:
                System.out.print(" Pilihan anda diluar A atau B ");
        }
    }
}
```



## Codingan 2.1

```
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();

        if (bulan == 1 || bulan == 3 || bulan == 5 || bulan == 7 || bulan == 8 || bulan == 10)
            jumlahHari = 31;
        } else if (bulan == 4 || bulan == 6 || bulan == 9 || bulan == 11) { // baris 2
            jumlahHari = 30;
        } else if (bulan == 2) {
            if (tahun % 4 == 0) {
                jumlahHari = 29;
            } else {
                jumlahHari = 28;
            }
        } else {
            System.out.println("Maaf bulan hanya sampai 12.");
        }

        System.out.println("Jumlah hari = " + jumlahHari);
    }
}
```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Output Generated Files

Masukkan data bulan (dalam angka): 10  
Jumlah hari = 31

Compiled and executed in 17.396 sec(s)

## Codingan 2.2 bulan dengan 31 hari

```
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 9: case 11: //baris 1
                jumlahHari = 31;
                break;
            case 4: case 6: case 8: case 10: case 12: //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);
    }
}
```

Input/Output

Language Version: JDK 21.0.0 ☒ Inter

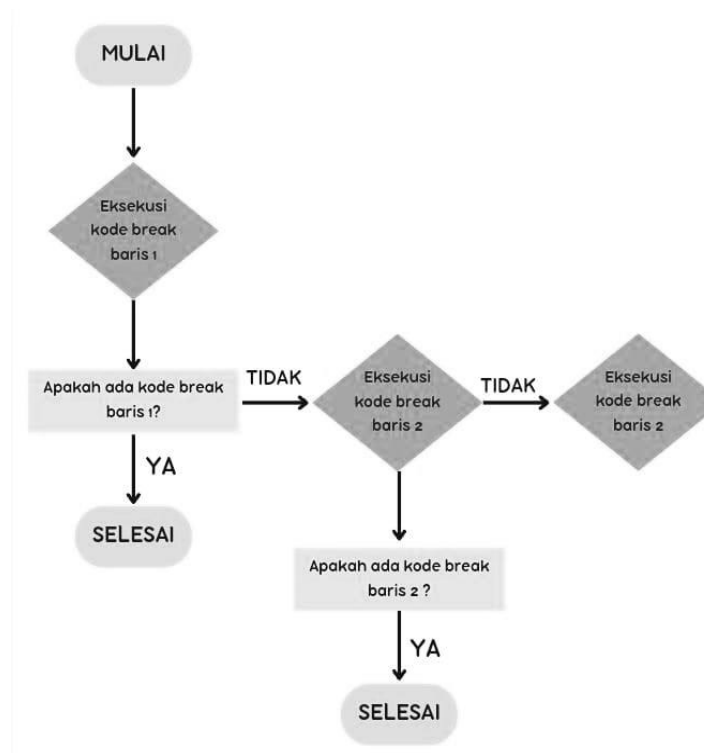
Input Arguments

Output Generated Files

Masukkan data bulan (dalam angka): 12  
Jumlah hari = 30

Compiled and executed in 4.771 sec(s)

## Codingan 2.2 dengan bulan 30 hari



Flowchart 2.1

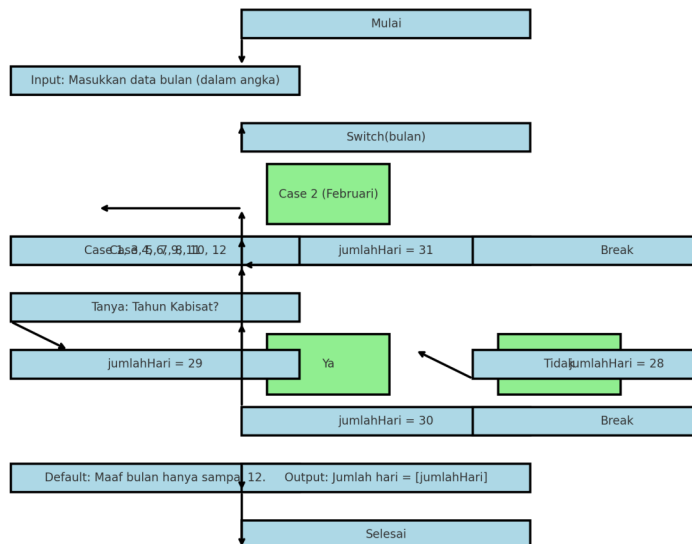
a. Analisa luaran yang dihasilkan:

Luaran yang dihasilkan sudah sesuai dengan program yang disusun.

Kondisi Switch yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

### [No.1] Kesimpulan

Flowchart Latihan 2.2: Menghitung Jumlah Hari Sampai Bulan ke-12



### Analisa

A. Susunlah kesimpulan!

Kesimpulannya adalah

#### 1. Analisis Penghapusan break

break digunakan untuk menghentikan eksekusi switch setelah sebuah case selesai. Jika break dihapus, switch akan terus mengeksekusi kode di case berikutnya meskipun case tersebut tidak sesuai. Kegunaan kode break dan kata kunci default adalah break itu digunakan untuk menghentikan eksekusi switch setelah satu case selesai, sedangkan default menangani semua input yang tidak sesuai dengan case yang ada.

#### 2. Penambahan kode program sampai bulan 12

Nah disini, menambahkan kode program bulan sampai dengan bulan 12. Disini menambahkan kode program bulan dengan mengelompokkan jumlah hari setiap bulan tersebut. Untuk bulan yang jumlah harinya 31, ada dibilang Januari, Maret, Mei, Juli, Agustus, Oktober, Desember. Untuk bulan yang jumlah harinya 30, ada di bulan April, Juni, September, November. Dan yang terakhir yaitu bulan dengan jumlah hari 28 atau 29, yaitu bulan Februari. Untuk bulan Februari, perlu dilakukan pemeriksaan apakah tahun kabisat atau tidak.

#### 3. apakah bisa diubah menjadi if?

Switch pada contoh 3 bisa diubah menjadi if, karena hal tersebut sama sama berguna untuk memeriksa beberapa variabel, meskipun switch lebih disukai jika kondisi yang diperiksa adalah nilai konstan tertentu seperti angka atau karakter, sedangkan if-else if lebih fleksibel untuk menangani berbagai jenis ekspresi logika.

B. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Alasannya supaya bisa menganalisis kondisi dari masing masing kode program yang dibuat dan dijalankan seperti pada contoh contoh di atas yang telah saya bahas.