

### Template Lembar Kerja Individu dan Kelompok

| Nama & NPM  | Topik:             | Tanggal:          |
|---|--------------------|-------------------|
| Sindi Putri Utami<br>G1F024053  | IF dan SWITCH java | 19 september 2024 |
| <b>[1] Identifikasi Masalah:</b>  |                    |                   |
| <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)?<br/>(Petunjuk: <math>\text{hitung} = \text{nilai} \% 2</math> jika <math>\text{hitung} = 0</math> maka bilangan genap, jika <math>\text{hitung} = 1</math> maka bilangan ganjil)</p> <p>1.2 Cermati contoh 2, analisa kondisi pada IF bersarang!<br/>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?<br/>Periksa satu kondisi mana yang paling tepat menggantikan ketiga kondisi itu!</p> <ol style="list-style-type: none"> <li>IF (<math>\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80</math>)</li> <li>IF (<math>\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \ \&amp;\&amp; \text{nilaiQ} \geq 80</math>)</li> <li>IF (<math>\text{nilaiU} \geq 80 \ \&amp;\&amp; \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80</math>)</li> <li>IF (<math>\text{nilaiU} \geq 80 \ \&amp;\&amp; \text{nilaiT} \geq 80 \ \&amp;\&amp; \text{nilaiQ} \geq 80</math>)</li> </ol> <p>1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!</p>            |                    |                   |
| <b>[1] Analisis dan Argumentasi</b>   |                    |                   |
| <p>1.1. Analisa kode program yang tepat untuk menghitung masukan pengguna termasuk bilangan genap atau bilangan ganjil.</p> <ol style="list-style-type: none"> <li>Impor Package:<br/><b>import java.util.Scanner;</b><br/>Ini memanggil package Scanner yang digunakan untuk membaca input dari pengguna.</li> <li>Deklarasi Kelas:<br/><b>public class PercabanganIf {</b><br/>Kode ini mendefinisikan kelas PercabanganIf.</li> <li>Metode main:<br/><b>public static void main(String[] args) {</b><br/>Ini adalah metode utama yang dijalankan saat program dieksekusi.</li> <li>Inisialisasi Scanner:<br/><b>Scanner masuk = new Scanner(System.in);</b><br/>Membuat objek Scanner bernama masuk untuk membaca input dari sistem.</li> <li>Membaca Input Pengguna:<br/><b>System.out.print("Masukkan Angka Anda: ");</b><br/><b>int nilai = masuk.nextInt(); // Menggunakan nextInt untuk tipe data integer</b><br/>Di sini, program meminta pengguna untuk memasukkan angka, yang kemudian disimpan dalam variabel nilai.</li> <li>Percabangan untuk Cek Nilai:<br/><b>if (nilai == 10) {</b><br/>    <b>System.out.println("Sepuluh");</b><br/><b>} else {</b><br/>    <b>System.out.println("Nilai Bukan Sepuluh");</b><br/><b>}</b><br/>Kode ini memeriksa apakah nilai yang dimasukkan sama dengan 10 dan mencetak pesan yang sesuai.</li> <li>Menghitung Genap atau Ganjil:<br/><b>int hitung = nilai % 2;</b><br/><b>if (hitung == 0) {</b></li> </ol> |                    |                   |

```

        System.out.println("Bilangan Genap");
    } else {
        System.out.println("Bilangan Ganjil");
    }
}

```

- nilai % 2 menghitung sisa pembagian nilai dengan 2.
- Jika sisa (hitung) sama dengan 0, maka angka tersebut adalah genap.
- Jika sisa sama dengan 1, maka angka tersebut adalah ganjil.

Kode ini dapat menentukan apakah input pengguna adalah bilangan genap atau ganjil dengan menggunakan operator modulus/sisa bagi. Pengguna diminta untuk memasukkan angka, lalu program mengecek jika angka itu 20 dan memberi tahu apakah angka tersebut genap atau ganjil berdasarkan sisa bagi dengan 2.

1.2. Menganalisa kondisi pada IF bersarang, menambahkan 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.

Program ini meminta pengguna untuk memasukkan nilai ujian, tugas, dan kuis. Jika nilai ujian, kuis, dan tugas semua 80 atau lebih, program akan mencetak "Anda mendapatkan nilai A." Jika nilai ujian kurang dari 80, program akan mencetak "Anda TIDAK mendapatkan nilai A."

1. Input Nilai

Tiga nilai diminta dari pengguna:

- Nilai Ujian (nilaiU): Nilai yang diperoleh dari ujian.
- Nilai Tugas (nilaiT): Nilai yang diperoleh dari tugas.
- Nilai Kuis (nilaiQ): Nilai yang diperoleh dari kuis.

2. Kondisi If Bersarang:

- Nilai Ujian: program memeriksa apakah nilaiU lebih besar atau sama dengan 80. Jika true, program melanjutkan ke pemeriksaan nilai tugas.
- Nilai Tugas: Selanjutnya, jika nilai ujian memenuhi syarat, program memeriksa nilaiT. Jika nilai tugas juga memenuhi syarat, program melanjutkan ke pemeriksaan nilai kuis.
- Nilai Kuis: Jika kedua nilai sebelumnya benar, program memeriksa nilaiQ. Jika semua nilai (ujian, tugas, dan kuis) memenuhi syarat, program mencetak "Anda mendapatkan nilai A". Jika tidak, program mencetak "Anda TIDAK mendapatkan nilai A".

3. Logika:

- Setiap kondisi harus dievaluasi secara terpisah.
- Jika salah satu nilai tidak memenuhi syarat, program langsung mencetak hasil tanpa memeriksa nilai lainnya.

1.3. Meringkas ketiga kondisi IF pada Contoh 1.2

Kode yang paling tepat menggantikan ketiga kondisi pada codingan ini adalah **IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)**

Kode blok pernyataan if yang mengevaluasi tiga kondisi:

- Kondisi: nilaiU >= 80, nilaiT >= 80, dan nilaiQ >= 80.  
Ini memeriksa apakah nilai ujian, tugas, dan kuis semua lebih besar atau sama dengan 80.
- Logika &&:  
Operator logika && berarti "dan". Semua kondisi harus benar agar keseluruhan pernyataan menjadi true.

- Eksekusi:

Jika semua kondisi terpenuhi (true), program mencetak "Anda mendapatkan nilai A".  
Jika salah satu dari kondisi tersebut tidak terpenuhi (false), program mencetak "Anda TIDAK mendapatkan nilai A".

## [1] Penyusunan Algoritma dan Kode Program

### 1.1. Menentukan bilangan genap atau bilangan ganjil

#### 1. Algoritma

1. Mulai.
2. Impor package Scanner untuk membaca input pengguna.
3. Inisialisasi objek Scanner dengan nama masuk.
4. Tampilkan pesan "Masukkan Angka Anda : " kepada pengguna.
5. Baca input dari pengguna dan simpan ke dalam variabel nilai.
6. Jika nilai dibagi 2 menghasilkan sisa 0 ( $\text{nilai} \% 2 == 0$ ):  
Tampilkan "Nilai Genap".
7. Jika tidak (else):  
Tampilkan "Nilai Ganjil".
8. 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.nextInt(); //menyimpan masukan pengguna
            ke tipe data
8
9         if (nilai % 2 == 0) { //percabangan yang memeriksa kondisi
10             System.out.println("Nilai Genap"); //baris kode yang dieksekusi bila
                benar
11         }
12     else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan
                salah
13         System.out.println("Nilai Ganjil");
14     }
15 }
```

```
java -cp . /tmp/zlmv6rYx1/PercabanganIf
Masukkan Angka Anda : 20
Nilai Genap
--- Code Execution Successful ---
```

Luaran sudah sesuai dengan program yang disusun.

### 1.2. Kondisi pada IF bersarang

#### 1. Algoritma

1. Mulai.
2. Impor package Scanner untuk membaca input pengguna.
3. Inisialisasi objek Scanner dengan nama varU.
4. Tampilkan pesan "Masukkan Angka Ujian Anda : ".
5. Baca nilai ujian dari pengguna dan simpan ke dalam variabel nilaiU.
6. Inisialisasi objek Scanner dengan nama varT.
7. Tampilkan pesan "Masukkan Angka Tugas Anda : ".
8. Baca nilai tugas dari pengguna dan simpan ke dalam variabel nilaiT.
9. Inisialisasi objek Scanner dengan nama varQ.
10. Tampilkan pesan "Masukkan Angka Quiz Anda : ".
11. Baca nilai kuis dari pengguna dan simpan ke dalam variabel nilaiQ.
12. Jika nilaiU  $\geq 80$ :  
Jika nilaiQ  $\geq 80$ :  
Jika nilaiT  $\geq 80$ :
13. Jika semua kondisi di atas terpenuhi:  
Tampilkan "Anda mendapatkan nilai A".
14. Jika salah satu kondisi di atas tidak terpenuhi:  
Tampilkan "Anda TIDAK mendapatkan nilai A".
15. Selesai.

#### 2. Kode program dan luaran

```

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 }

```

```

java -cp /tmp/76aYXBtWkr/IfBersarang
Masukkan Angka UTS Anda : 88
Masukkan Angka Tugas Anda : 90
Masukkan Angka Quiz Anda : 80
Anda mendapatkan nilai A

=== Code Execution Successful ===

```

Luaran sudah sesuai dengan program yang disusun.

### 1.3. Meringkas ketiga kondisi IF pada Contoh 1.2.

#### 1. Algoritma

1. Mulai.
2. Impor package Scanner untuk membaca input pengguna.
3. Inisialisasi objek Scanner dengan nama varU.
4. Tampilkan pesan "Masukkan Angka UTS Anda : ".
5. Baca nilai UTS dari pengguna dan simpan ke dalam variabel nilaiU.
6. Inisialisasi objek Scanner dengan nama varT.
7. Tampilkan pesan "Masukkan Angka Tugas Anda : ".
8. Baca nilai tugas dari pengguna dan simpan ke dalam variabel nilaiT.
9. Inisialisasi objek Scanner dengan nama varQ.
10. Tampilkan pesan "Masukkan Angka Quiz Anda : ".
11. Baca nilai kuis dari pengguna dan simpan ke dalam variabel nilaiQ.
12. Jika nilaiU >= 80 dan nilaiT >= 80 dan nilaiQ >= 80:  
Tampilkan "Anda mendapatkan nilai A".
13. Jika tidak:  
Tampilkan "Anda TIDAK mendapatkan nilai A".
14. Selesai.

#### 2. Kode program dan luaran

```

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

```

```

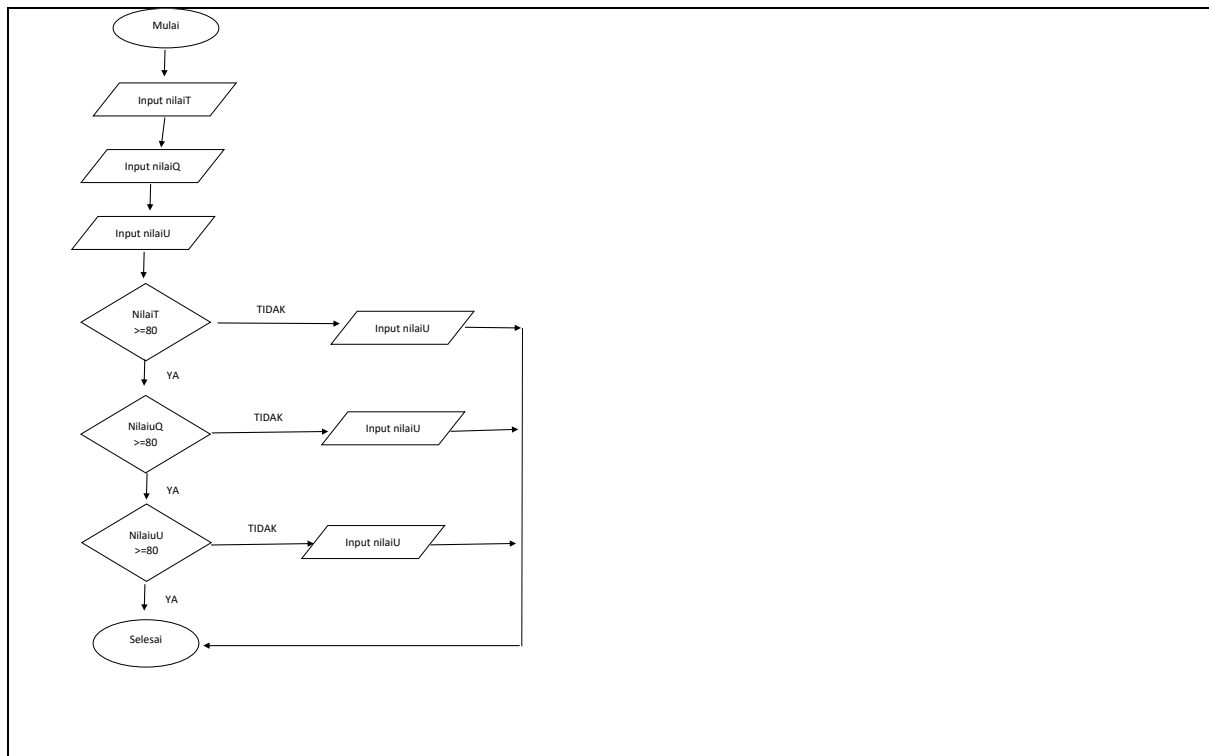
java -cp /tmp/76aYXBtWkr/IfBersarang
Masukkan Angka UTS Anda : 85
Masukkan Angka Tugas Anda : 90
Masukkan Angka Quiz Anda : 78
Anda TIDAK mendapatkan nilai A

=== Code Execution Successful ===

```

Luaran sudah sesuai dengan program yang disusun.

### 1.4. Gambar diagram flowchart dari Latihan 1.2



### [1] Kesimpulan

Program pertama menggunakan Scanner untuk membaca input dan memeriksa apakah angka yang dimasukkan genap atau ganjil dengan operator modulus. Program kedua mengevaluasi nilai akademik (UTS, Tugas, dan Kuis) menggunakan kondisi if bersarang. Program ketiga mengevaluasi nilai akademik (UTS, Tugas, dan Kuis) menggunakan kondisi if yang digabung menjadi satu pernyataan.

### [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!

### [2] Analisis dan Argumentasi

- 2.1. Analisis perbedaan hasil luaran ketika kode menggunakan break, ketika kode break baris 1 dihapus, dan ketika kode break baris 2 dihapus.
  - Kegunaan break untuk memastikan bahwa setelah mengeksekusi case yang sesuai, eksekusi akan berakhir. Ini mencegah fall through ke case berikutnya. Misalnya, jika pengguna memilih A, pernyataan break setelah case A akan

menghentikan eksekusi lebih lanjut, sehingga program tidak akan melanjutkan ke case B atau default

- Kegunaan default untuk menangani kasus ketika tidak ada case yang cocok, yaitu case (A atau B).

1. Kode dengan break (Asli)

- Ketika pengguna memilih A, outputnya adalah:  
Anda sudah rajin belajar
- Ketika pengguna memilih B, outputnya adalah:  
Anda perlu kurangi main game
- Ketika pengguna memilih karakter lain, outputnya adalah:  
Pilihan anda diluar A atau B

2. Kode tanpa break pada baris 1

- Tanpa break, eksekusi akan terus melanjutkan ke case berikutnya yaitu case B Setelah mengeksekusi case A, sehingga menghasilkan output dari kedua case.
- Jika menghapus break dari case A, dan pengguna memilih A, outputnya adalah:  
Anda sudah rajin belajar Anda perlu kurangi main game

3. Kode tanpa break pada baris 2

- Tanpa break eksekusi akan melanjutkan ke blok default setelah case B, sehingga menghasilkan output dari case B dan default.
- Jika menghapus break dari case B, dan pengguna memilih B, outputnya adalah:  
Anda perlu kurangi main game Pilihan anda diluar A atau B

2.2. Menambahkan sampai bulan ke-12 pada baris ke-1 dan baris ke-2.

1. Input Pengguna: Menggunakan Scanner untuk membaca nomor bulan.

2. Switch Case:

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

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

Februari: Menghitung berdasarkan tahun kabisat (29 hari jika tahun habis dibagi 4, 28 hari jika tidak).

Output: Menampilkan jumlah hari yang sesuai, atau pesan kesalahan jika input tidak valid (bukan 1-12).

2.3. Mengubah codingan contoh 3 menjadi perintah IF

Program ini meminta pengguna untuk memilih antara 'A' atau 'B' dan memberikan respon sesuai pilihan tersebut.

1. Input Pengguna:

- Program meminta pengguna untuk memasukkan karakter, yang dapat berupa 'A' atau 'B'.

2. Percabangan dengan if-else:

- Struktur if-else digunakan untuk memeriksa nilai input:
  - Jika input adalah 'A', program mencetak "Anda sudah rajin belajar".
  - Jika input adalah 'B', program mencetak "Anda perlu kurangi main game".
  - Jika input tidak valid (tidak 'A' atau 'B'), program mencetak pesan kesalahan.

## Perbandingan antara if-else dan switch

### 1. Struktur dan Keterbacaan

- if-else:
  - Lebih fleksibel karena dapat menangani kondisi yang kompleks dan lebih dari sekadar perbandingan nilai.
  - Membuatnya lebih sulit dibaca ketika ada banyak kondisi, karena setiap kondisi harus ditulis secara terpisah.
- switch:
  - Lebih terstruktur dan mudah dibaca ketika ada banyak opsi tetap.
  - Hanya dapat menangani perbandingan nilai tetap, seperti integer, karakter, atau string, dan tidak mendukung kondisi yang lebih kompleks.

### 2. Kinerja

- if-else:

Evaluasi dilakukan secara berurutan, jadi jika ada banyak kondisi, ini dapat menjadi kurang efisien karena semua kondisi diperiksa satu per satu.
- switch:

Dioptimalkan untuk pemilihan di antara banyak nilai, sehingga dapat lebih efisien daripada banyak pernyataan if-else.

### 3. Kasus Penggunaan

Cocok untuk situasi di mana Anda memiliki sejumlah opsi yang terbatas dan tetap. Misalnya, memilih tindakan berdasarkan kode atau menu.

## [2] Penyusunan Algoritma dan Kode Program

### 2.1 kode break; pada //baris 1 dan kode break; pada //baris 2.

#### 1. Menghapus kode break; pada //baris 1

##### 1. Algoritma

1. Mulai
2. Deklarasi Variabel:
  - Deklarasikan masukData sebagai objek Scanner.
  - Deklarasikan data sebagai karakter.
3. Ambil Input:
  - Tampilkan pesan: "Pilih A atau B: ".
  - Baca karakter input dari pengguna dan simpan ke dalam variabel data.
4. Struktur Kontrol Switch:

Switch data:

  - Kasus 'A':
    - Tampilkan pesan: "Anda sudah rajin belajar".
    - (Tidak ada break, lanjut ke kasus berikutnya).
  - Kasus 'B':
    - Tampilkan pesan: "Anda perlu kurangi main game".
    - break untuk keluar dari switch.
  - Default:
    - Tampilkan pesan: "Pilihan anda diluar A atau B".
5. Selesai

#### 2. Kode program dan luaran

|   |   |
|---|---|
| <pre> 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                 // break; // baris 1 // kode break pada baris ini dihapus 13             case 'B': 14                 System.out.print(" Anda perlu kurangi main game"); 15                 break; // baris 2 16             default: 17                 System.out.print(" Pilihan anda diluar A atau B "); 18         } 19     } 20 } </pre> | <pre> java -cp ./tmp/CRGKF0B0n6/SwitchBersarang Pilih A atau B : A Anda sudah rajin belajar Anda perlu kurangi main game --- Code Execution Successful --- </pre> |
|---|---|

Luaran sudah sesuai dengan program yang disusun.

## 2. Menghapus kode break; pada //baris 2

### 1. Algoritma

1. Mulai
2. Inisialisasi: Buat objek Scanner untuk membaca input dari pengguna.
3. Tampilkan pesan: Minta pengguna untuk memilih 'A' atau 'B'.
4. Baca input: Simpan input karakter pengguna ke dalam variabel data.
5. Struktur Kontrol: Gunakan switch untuk memeriksa nilai data.
  - Kasus 'A':
    - Tampilkan pesan "Anda sudah rajin belajar".
    - Keluar dari switch (gunakan break).
  - Kasus 'B':
    - Tampilkan pesan "Anda perlu kurangi main game".
    - (Tanpa break): Eksekusi berlanjut ke kasus default.
  - Kasus default:
    - Tampilkan pesan "Pilihan anda diluar A atau B".
6. Selesai

### 2. Kode program dan luaran

|   |  |
|---|--|
| <pre> 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                 break; // baris 1 13             case 'B': 14                 System.out.print(" Anda perlu kurangi main game"); 15                 // break; // baris 2 // kode break pada baris ini dihapus 16             default: 17                 System.out.print(" Pilihan anda diluar A atau B "); 18         } 19     } 20 } </pre> | <pre> java -cp ./tmp/p0d0c1q0t/SwitchBersarang Pilih A atau B : B Anda perlu kurangi main game Pilihan anda diluar A atau B --- Code Execution Successful --- </pre> |
|---|--|

Luaran sudah sesuai dengan program yang disusun.

## 2.2. Menambahkan sampai bulan ke-12 pada baris ke-1 dan baris ke-2.

### 1. Algoritma

1. Mulai
2. Deklarasi Variabel:
  - bulan (tipe byte)
  - tahun (tipe int dan set nilainya ke 2022)
  - jumlahHari (tipe int dan set nilainya ke 0)
3. Tampilkan Pesan: "Masukkan data bulan (dalam angka):"
4. Baca Input:
  - Masukkan nilai ke dalam variabel bulan.
5. Struktur Kontrol (Switch):
  - Jika bulan = 1, 3, 5, 7, 8, 10, 12:  
Set jumlahHari = 31



- Jika bulan = 4, 6, 9, 11:  
Set jumlahHari = 30
  - Jika bulan = 2:  
Jika tahun % 4 = 0:  
Set jumlahHari = 29  
Jika tidak:  
Set jumlahHari = 28
  - Jika tidak ada kasus di atas (default):  
Tampilkan pesan: "Maaf bulan hanya sampai 12."
6. Tampilkan Hasil: "Jumlah hari = " + jumlahHari
  7. Selesai

## 2. Kode program dan luaran

The screenshot shows a Java IDE with a code editor on the left and a console on the right. The code is a Java program that calculates the number of days in a month based on the month number and the year. It uses a switch statement for months 1-12 and an if-else statement for leap years. The output in the console shows the user inputting 6 for the month, resulting in 30 days.

```

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:
15        case 12: //baris 1
16            jumlahHari = 31;
17            break;
18        case 4: case 6: case 9: case 11: //baris 2
19            jumlahHari = 30;
20            break;
21        case 2:
22            if (tahun % 4 == 0)
23                jumlahHari = 29;
24            else
25                jumlahHari = 28;
26            break;
27        default:
28            System.out.println("Maaf bulan hanya sampai 12.");
29            break;
30    }
31    System.out.println("Jumlah hari = " + jumlahHari);
32 }
33 }

```

Output in console:

```

C:\java> java -cp .\bin\main\MainSwitchBersejarah
Masukkan data bulan (dalam angka): 6
Jumlah hari = 30
=== Code Execution Successful ===

```

Luaran sudah sesuai dengan program yang disusun.

## 2.3. Mengubah codingan contoh 3 menjadi perintah IF

1. Algoritma
  1. Mulai
  2. Deklarasi Variabel:
    - Deklarasikan variabel data bertipe char.
  3. Tampilkan Pesan:
    - "Pilih A atau B : "
  4. Baca Input:
    - Masukkan karakter dari pengguna dan simpan dalam variabel data.
  5. Struktur Kontrol (if-else):
    - Jika data sama dengan 'A':  
Tampilkan pesan: "Anda sudah rajin belajar".
    - Jika data sama dengan 'B':  
Tampilkan pesan: "Anda perlu kurang main game".
    - Jika tidak:  
Tampilkan pesan: "Pilihan anda diluar A atau B".
  6. Selesai

## 2. Kode program dan luaran

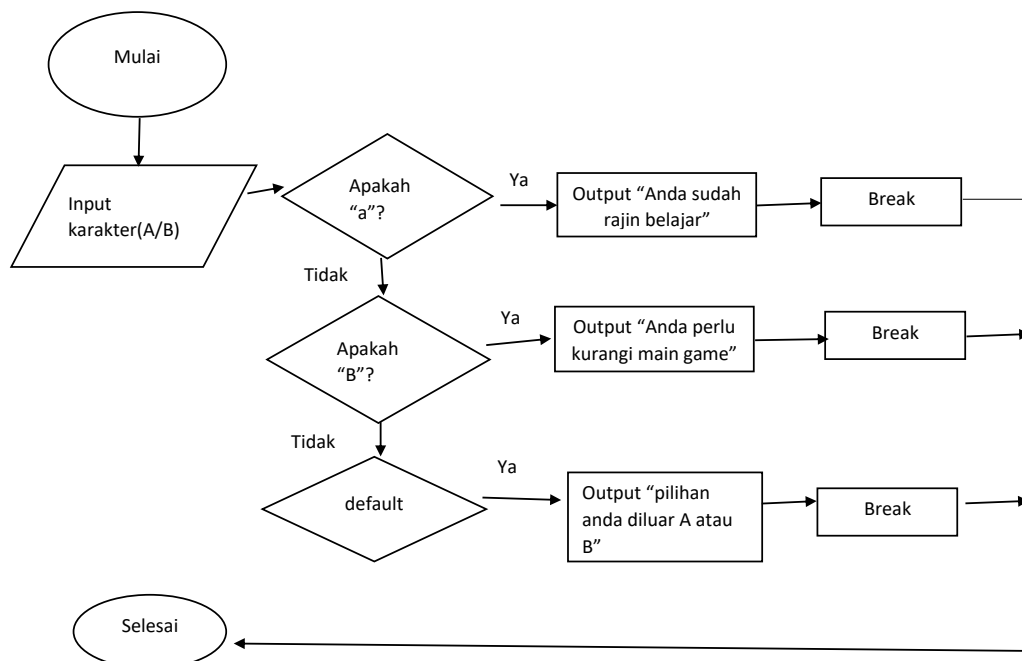
```
1- import java.util.Scanner;
2
3 public class SwitchBersarang {
4     public static void main(String[] args) {
5         Scanner masukData = new Scanner(System.in);
6         // mengambil input
7         System.out.print("Pilih A atau B : ");
8         char data = masukData.next().charAt(0);
9
10        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/jlszf0BUPK/switchbersarang
Pilih A atau B : A
Anda sudah rajin belajar
== Code Execution Successful ==
```

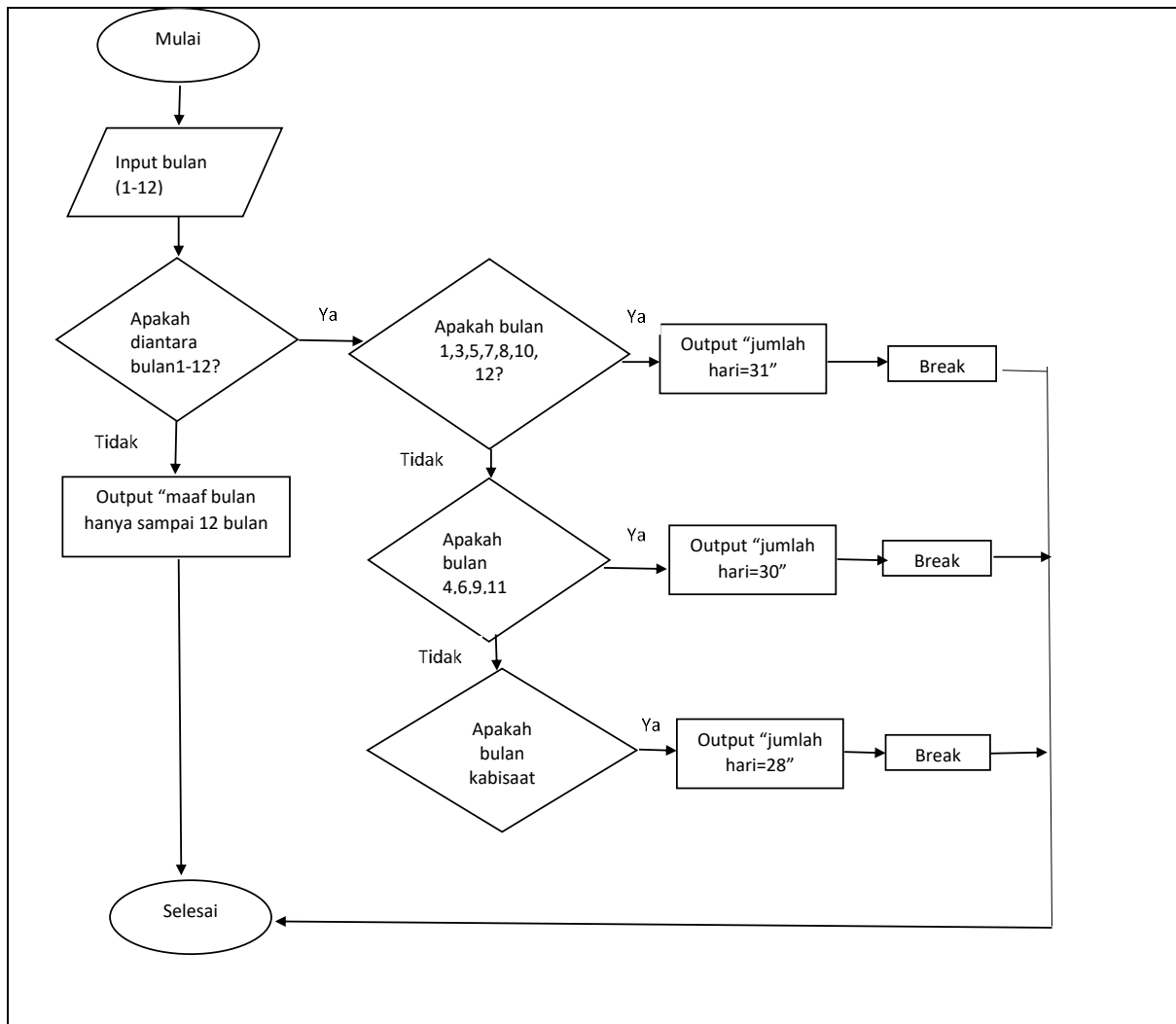
Luaran sudah sesuai dengan program yang disusun.

## 2.4. gambar flowchart dari Latihan 2.1. dan Latihan 2.2

### 1. flowchart latihan 2.1



### 2. Gambar flowchart latihan 2.2



## [2] Kesimpulan

Dua contoh program menggunakan pernyataan switch dalam Java untuk mengendalikan alur berdasarkan input pengguna. Contoh 3 memilih antara dua pilihan ('A' atau 'B') dan memberikan respon sesuai pilihan, dengan penanganan untuk kasus yang tidak valid, sementara Contoh 4 menentukan jumlah hari dalam bulan berdasarkan input numerik, termasuk penanganan tahun kabisat untuk Februari. Keuntungan penggunaan switch adalah memudahkan struktur kode dan meningkatkan keterbacaan, serta lebih efisien untuk menangani banyak pilihan tetap dibandingkan dengan banyak pernyataan if-else. Secara keseluruhan, penggunaan switch membuat pengendalian alur program lebih sederhana dan fleksibel.

## Refleksi

saya belajar tentang penggunaan if dan switch di Java. If bisa menangani berbagai kondisi yang mungkin muncul, sedangkan switch lebih terstruktur untuk memilih di antara beberapa pilihan. Tantangan yang saya hadapi adalah bingung kapan harus pakai if dan kapan pakai switch. Kadang, kondisi yang rumit bikin saya ragu.

