

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Rivan Alfatoni G1F024047</b>	<b>IF ELSE &amp; SWITCH</b>	<b>21 September 2024</b>

### [No. 1] Identifikasi Masalah:

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. Mencari bilangan genap dapat menggunakan operator aritmatika, yaitu modulus atau sisa hasil bagi, jika sisa 0 maka bilangan tersebut adalah genap dan jika tidak maka ganjil

1.2. Menambahkan kondisi if dengan nilai UTS berarti menambah 1 cabang if lagi.

1.3. Dari ketiga cabang if yang digunakan dapat disederhanakan menjadi 1 if yaitu dengan operator Boolean AND yaitu && dan pada pilihan soal tepatnya pada jawaban d.

1.4. Flowchart dapat dilihat pada gambar dibawah.

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

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

Algoritma IF ELSE sebuah program

- Mulai
- Mengisi inputan atau mendapatkan sebuah value dalam variable
- Mengecek apakah inputan terpenuhi atau tidak
- Jika terpenuhi jalankan kode program yang terpenuhi
- Jika tidak jalankan kode program yang tidak terpenuhi
- Selesai

2) Kode program dan luaran

- Screenshot/ Capture potongan kode dan hasil luaran  
(Coding 1.1 Mencari bilangan ganjil genap)

```

1  import java.util.Scanner; // Memasukkan Library Scanner
2
3  class ganjilgenap { // Membuat Class ganjilgenap
4      public static void main(String[] args) { // Deklarasi method main
5          Scanner angka = new Scanner(System.in); // Membuat Scanner
6          System.out.print("Masukkan angka : "); // Membuat tampilan input
7          int a = angka.nextInt(); // Menginputkan angka
8
9          if(a % 2 == 0){ // Mengecek apakah angka genap
10             System.out.println("Genap");
11         }else{ // Jika bukan genap
12             System.out.println("Ganjil");
13         }
14     }
15 }
16

```

Luaran:

```

Masukkan angka : 200
Genap

```

( Coding 1.2 Mencari nilai Tugas, Quiz dan UTS )

```

2  // Mengimport Scanner untuk menginputkan nilai
3  import java.util.Scanner;
4
5  // Membuat class IFBersarang
6  public class IFBersarang {
7      // Membuat method main
8      public static void main(String[] args) {
9          // Membuat Scanner untuk menginputkan nilai Tugas
10         Scanner varT = new Scanner(System.in);
11         System.out.print("Masukkan Angka Tugas Anda : ");
12         int nilaiT = varT.nextByte();
13
14         // Membuat Scanner untuk menginputkan nilai Quiz
15         Scanner varQ = new Scanner(System.in);
16         System.out.print("Masukkan Angka Quiz Anda : ");
17         int nilaiQ = varQ.nextByte();
18
19         // Membuat Scanner untuk menginputkan nilai UTS
20         Scanner varU = new Scanner(System.in);
21         System.out.print("Masukkan Angka UTS Anda : ");
22         int nilaiU = varU.nextByte();
23
24         // Mengecek apakah nilai Quiz >= 80
25         if (nilaiQ >= 80) {
26             // Mengecek apakah nilai Tugas >= 80
27             if (nilaiT >= 80) {
28                 // Mengecek apakah nilai UTS >= 80
29                 if (nilaiU >= 80) {
30                     // Jika ketiga nilai di atas >= 80 maka mendapatkan nilai A
31                     System.out.println("Anda mendapatkan nilai A");
32                 }
33             }
34         } else {
35             // Jika tidak maka tidak mendapatkan nilai A
36             System.out.println("Anda TIDAK mendapatkan nilai A");
37         }
38     }
39 }
40

```

Luaran

```
Masukkan Angka Tugas Anda : 90
Masukkan Angka Quiz Anda : 80
Masukkan Angka UTS Anda : 90
Anda mendapatkan nilai A
```

( Coding 1.3 Efisiensi If Bersarang)

```
1
2 // Mengimport Scanner untuk menginputkan nilai
3 import java.util.Scanner;
4
5 // Membuat class IFBersarang
6 public class IFBersarang {
7     // Membuat method main
8     public static void main(String[] args) {
9         // Membuat Scanner untuk menginputkan nilai Tugas
10        Scanner varT = new Scanner(System.in);
11        System.out.print("Masukkan Angka Tugas Anda : ");
12        int nilaiT = varT.nextByte();
13
14        // Membuat Scanner untuk menginputkan nilai Quiz
15        Scanner varQ = new Scanner(System.in);
16        System.out.print("Masukkan Angka Quiz Anda : ");
17        int nilaiQ = varQ.nextByte();
18
19        // Membuat Scanner untuk menginputkan nilai UTS
20        Scanner varU = new Scanner(System.in);
21        System.out.print("Masukkan Angka UTS Anda : ");
22        int nilaiU = varU.nextByte();
23
24        // Mengecek apakah nilai Quiz,Tugas dan UTS >= 80
25        if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80) {
26            // Jika ketiga nilai di atas >= 80 maka mendapatkan nilai A
27            System.out.println("Anda mendapatkan nilai A");
28        }
29        else {
30            // Jika tidak maka tidak mendapatkan nilai A
31            System.out.println("Anda TIDAK mendapatkan nilai A");
32        }
33    }
34 }
35
```

Luaran:

```
Masukkan Angka Tugas Anda : 90
Masukkan Angka Quiz Anda : 90
Masukkan Angka UTS Anda : 80
Anda mendapatkan nilai A
```

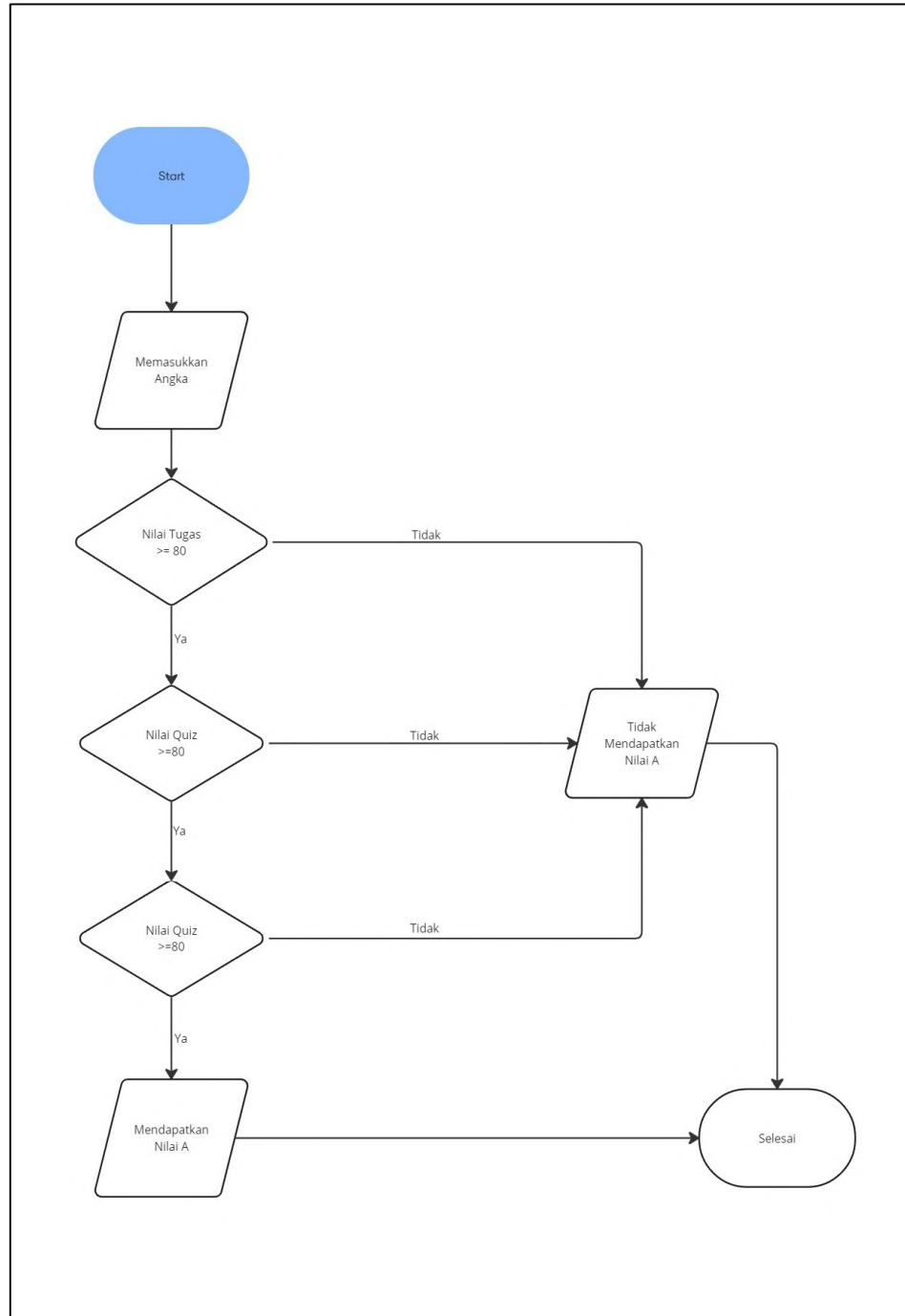
b) Analisa luaran yang dihasilkan

(1.1) Luaran pada code 1.1 menghasilkan genap karena modulus 200 dibagi 2 adalah 0.

Karena jika bilangan habis dibagi 2 maka bilangan tersebut adalah bilangan genap, dan jika tidak habis dibagi dengan 2 maka bilangan ganjil

(1.2) Luaran pada code 1.2 dan code 1.3 sama, yaitu mencari kondisi ketiga nilai dari Tugas, Quiz dan Uts. Jika ketiga nilai tersebut lebih atau sama dengan 80 maka mendapatkan nilai A dan jika salah satu nilai kurang dari 80 maka tidak mendapatkan nilai A.

c) Flowchart dari code 1.2



( Gambar Flowchart soal 1.4 Membuat di website <https://miro.com/> )

**[No.1] Kesimpulan**

Kode program untuk mencari bilangan ganjil dan genap dapat dilakukan dengan menggunakan operator aritmatika yaitu modulus atau sisa bagi, karena jika sisa bagi 0 maka bilangan tersebut adalah bilangan genap dan jika tidak habis atau sisa 1 adalah bilangan ganjil. Pada Latihan 1.2 If bercabang dapat dipersingkat dengan menggunakan operator AND(&&) karena kondisi if tersebut sama yaitu

sama sama meminta nilai lebih atau sama dengan 80. Flowchart yang Digambar sudah sesuai dengan kode program 1.2 karena kita melakukan pengecekan satu satu setiap nilai dan jika salah satu nilai tidak terpenuhi maka tidak akan mendapatkan nilai A atau sebaliknya jika seluruh nilai terpenuhi maka akan mendapatkan nilai A.

---

## [No. 2] Identifikasi Masalah:

- 2.1 Cermati kode pada Contoh 3.
  1. Hapuslah kode break; pada //baris 1, eksekusi kembali.
  2. Kemudian hapuslah kode break; pada //baris 2, eksekusi kembali.
  3. Analisis perbedaan hasil luaran ketika kode menggunakan break, Ketika kode break baris 1 dihapus, dan ketika kode break baris 2 dihapus!
  4. 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.
  - a. Apakah masalah ini bisa diubah menjadi perintah IF?
  - b. Jika bisa, rincikan analisa Anda!
  - c. 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 **Analisis:** Ketika kode break; pada baris 1 dihapus, eksekusi akan terus berlanjut ke case-case berikutnya tanpa keluar dari blok switch. Ini akan mengakibatkan semua statement di bawah case yang cocok dieksekusi, termasuk case lain yang tidak relevan.  
**Kesimpulan:** Hapusnya break pada baris 1 akan menyebabkan eksekusi berantai (fall-through) ke case-case di bawahnya.
- 2.2 Pada Contoh 4, bahwa mengecek bulan belum lengkap bahwa Ketika menginput bulan > 4 maka memunculkan bulan hanya sampai 12, maka saya menambahkan Solusi menambahkan case yang sesuai sampai dengan case 12
- 2.3 Permasalahan pada contoh 3 dapat dipecahkan dengan fungsi if, karena fungsi if juga bisa untuk mengecek kondisi jika menginput nilai A maka akan menampilkan “Anda sudah rajin belajar” jika menginput nilai B maka akan “Anda perlu kurangi main game” dan jika inputan diluar itu maka “Pilihan anda diluar A atau B “.
- 2.4 Flowchart dapat dilihat di poin penyusunan algoritma dan kode program.

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

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma SWITCH CASE Sebuah program
  1. Mulai
  2. Masukkan input atau value yang akan diperiksa
  3. Masukkan input atau value ke dalam struktur switch.
  4. Periksa nilai:
    - a. Jika sesuai dengan case tertentu, jalankan perintah dalam case tersebut.
    - b. Gunakan break untuk keluar dari switch setelah satu case terpenuhi.
  5. Jika tidak ada case yang cocok, jalankan blok default (opsional).

6. Selesai.

2) Kode program dan luaran

a) Kode program

( Kode Contoh 3 )

```
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:

Pilih A atau B : A

Anda sudah rajin belajar

a) Hapus break; pada baris 1:

case 'A':

System.out.print("Anda sudah rajin belajar");

b) Hapus break; pada baris 2:

case 'B':

System.out.print(" Anda perlu kurangi main game");

c) **Analisis:** Tanpa break, program akan terus masuk ke kasus bulan 2 dan nilai jumlahHari akan ditimpa dengan hasil bulan Februari.

d) **Analisis kegunaan break dan default:**

- **break** berfungsi untuk mencegah eksekusi **fall-through**, di mana kode dalam kasus berikutnya juga dieksekusi meskipun tidak sesuai dengan kondisi. Setiap kasus harus diakhiri dengan break agar program berhenti pada kondisi yang sesuai.
- **default** digunakan untuk menangani kondisi ketika tidak ada kasus yang sesuai dengan input. Ini berguna untuk memastikan bahwa program dapat menanggapi input yang tidak valid atau di luar jangkauan.

b) Screenshot/ Capture potongan kode dan hasil luaran

( Koding soal 2.2 Melengkapi 12 Bulan )

```

1  import java.util.Scanner;
2
3  /*
4   * Program untuk menghitung jumlah hari dalam bulan
5   */
6  public class SwitchBersarang {
7
8      /*
9       * Method main untuk mengeksekusi program
10      */
11     public static void main(String[] args) {
12         byte bulan; // Deklarasi variabel bulan
13         int tahun = 2022; // Deklarasi variabel tahun
14         int jumlahHari = 0; // Deklarasi variabel jumlahHari
15         System.out.print("Masukkan data bulan (dalam angka): "); // Tampilan input
16         Scanner masukData = new Scanner(System.in); // Deklarasi Scanner
17         bulan = masukData.nextByte(); // Menginputkan bulan
18         switch (bulan) { // Membuat switch untuk menentukan jumlah hari
19             case 1:case 3:case 5:case 7:case 8:case 10:case 12:
20                 jumlahHari = 31; // Jumlah hari 31
21                 break;
22             case 4:case 6:case 9:case 11:
23                 jumlahHari = 30; // Jumlah hari 30
24                 break;
25             case 2:
26                 if (tahun % 4 == 0) // Jika tahun kabisat
27                     jumlahHari = 29; // Jumlah hari 29
28                 else
29                     jumlahHari = 28; // Jumlah hari 28
30                 break;
31             default:
32                 System.out.println("Maaf bulan hanya sampai 12."); // Tampilan jika bulan tidak valid
33                 break;
34         }
35         System.out.println("Jumlah hari = " + jumlahHari); // Tampilan jumlah hari
36     }
37 }
38

```

Luaran Kode Program:

```

Masukkan data bulan (dalam angka): 8
Jumlah hari = 31

```

(Koding soal 2.3 Mengubah Masalah switch dengan If)

```

1  import java.util.Scanner;
2
3  public class IfBersarang2 {
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 }

```

Luaran kode

```

Pilih A atau B : A
Anda sudah rajin belajar

```

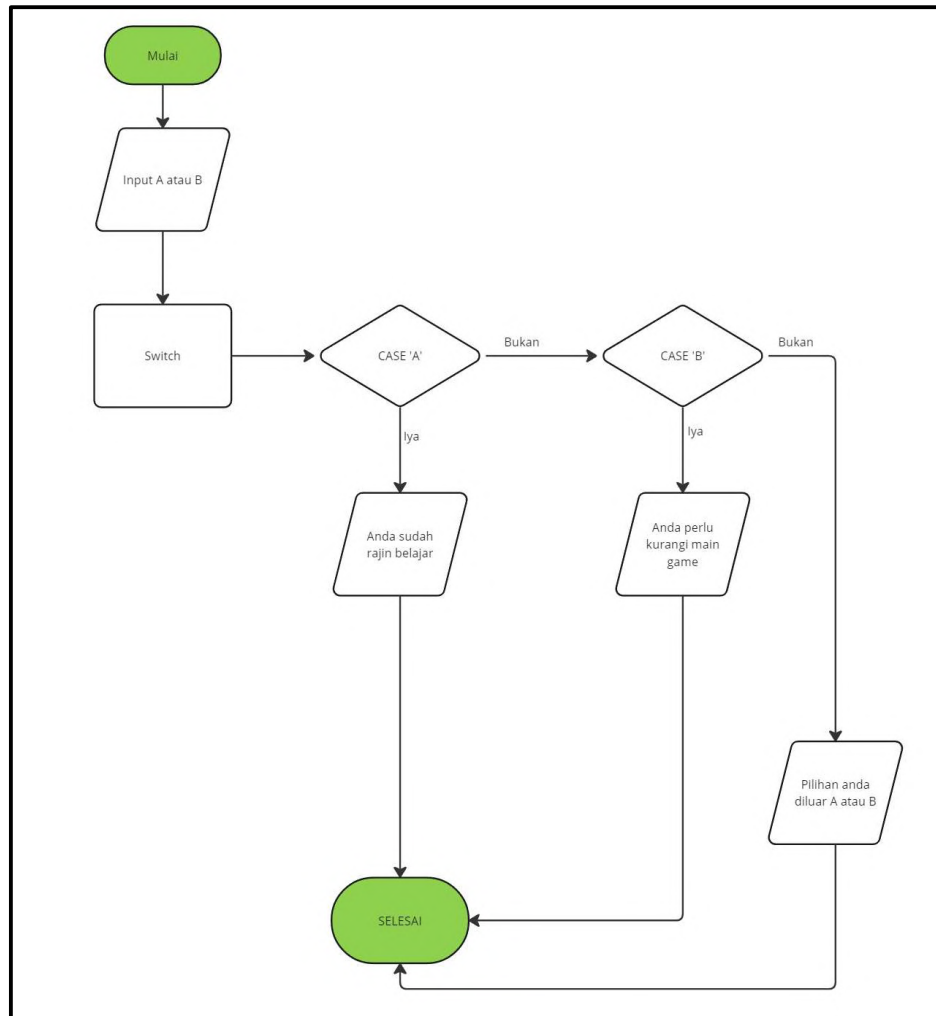
c) Analisa luaran yang dihasilkan

Luaran program 2.2 Kita hanya melengkapi kode pada contoh 4 menjadi 12 bulan dan Ketika memasukkan bulan 12 maka akan memunculkan 31 hari dan jika lebih 12 maka akan memunculkan maaf bulan hanya sampai 12.

Luaran pada program 2.3 sama seperti contoh 3 yaitu memilih 2 pilihan, fungsinya sama switch lebih cocok jika ada banyak kemungkinan nilai yang tetap (seperti enum, tipe char, atau tipe int). if-else lebih fleksibel karena dapat digunakan dengan kondisi yang lebih kompleks (seperti perbandingan logis atau gabungan beberapa syarat).

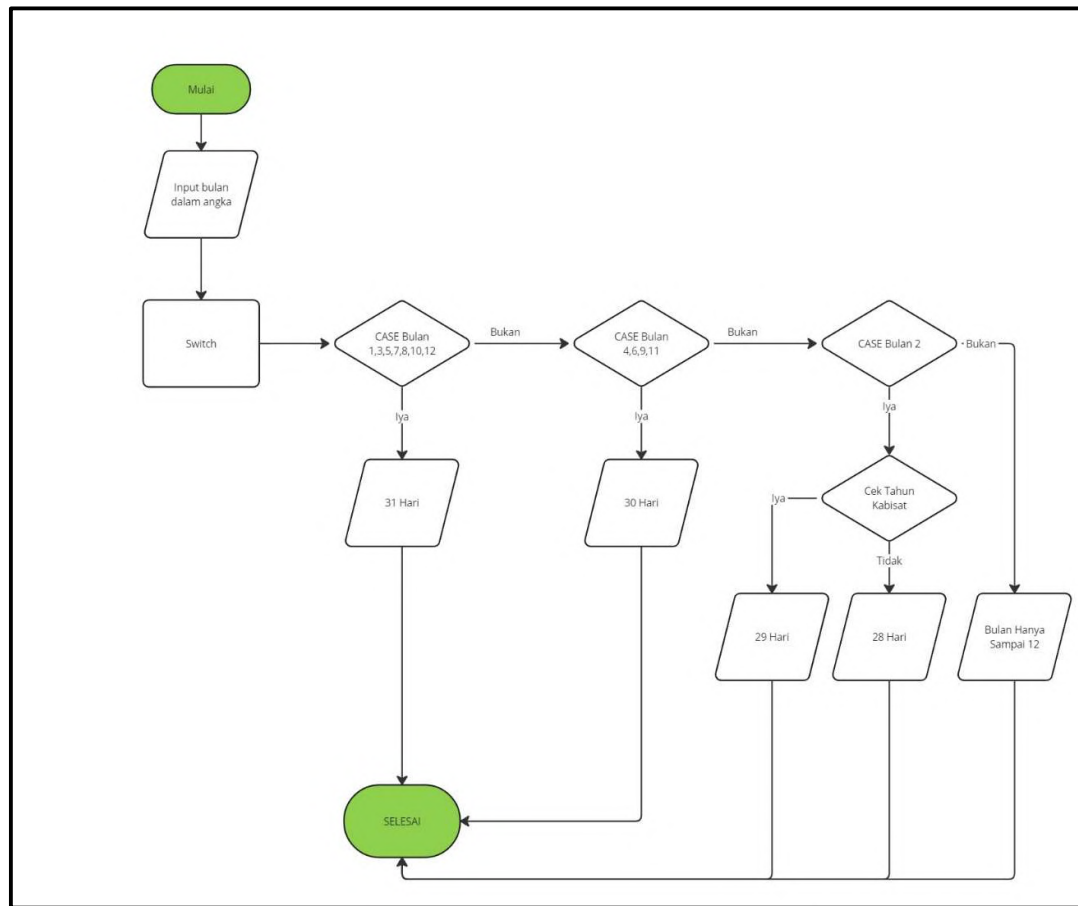
d) Flowchart

Flowchart Soal No 2.1



(Flowchart Soal 2.2)





## [No.2] Kesimpulan

Kesimpulan dari kode diatas untuk mencari bulan dan pilihan A atau B selain menggunakan switch juga dapat menggunakan if. Switch digunakan untuk mencari pilihan diatas karena pilihan sudah pasti ada.

## Refleksi

Praktikum minggu ini sangat menyenangkan karena pada materi if else dan switch saya memiliki basic nya. Namun tidak tau mengapa tingkatan saya turun ke Tingkat dasar.