|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Bagas Satrio Winata** |  | **17 September 2024** |

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

**Contoh 1:**

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

**public class PercabanganIf {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  // membaca teks yang dimasukkan pengguna  
        System.out.print("Masukkan Angka Anda : ");  //pengguna memasukkan data  
        nilai =  masuk.nextByte();  //menyimpan masukan pengguna ke tipe data  
          
        if (nilai = 1000) {  //percabangan yang memeriksa kondisi  
        System.out.println("Seribu");  //baris kode yang dieksekusi bila benar  
                }  
        else  {  //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah  
            System.out.println("Nilai Bukan Seribu");  
        }    }  
}**

**contoh 2:**

**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");  
    }  
    }  
}**

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

1. Saya mengusulkan permasalahan ini :

1.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

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!

a. IF (nilaiU >= 80 || nilaiT >= 80 || nilaiQ >= 80)

b. IF (nilaiU >= 80 || nilaiT >= 80 && nilaiQ >= 80)

c. IF (nilaiU >= 80 && nilaiT >= 80 || nilaiQ >= 80)

d. IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)

1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!

Jawaban:

* 1. Contoh 1:
* variable nilai belum dideklarasikan menggunakan byte
* kata masuk pada “masuk.byte” tidak sesuai dengan nama yang ada di scanner input
* pada operator perbandingan hanya menggunakan satu (=) yang merupakan operator penugasan
* nextbyte yang mana memiliki batas nilai tertentu

contoh 2:

* variable nilaiU yang tidak sesuai
* blum mempunyai operator nilai UTS
  1. Pada contoh ini, kondisi IF bersarang memeriksa satu per satu apakah nilai UTS, nilai Tugas, dan nilai Quiz lebih besar atau sama dengan 80. Jika semua syarat tersebut terpenuhi, maka siswa mendapatkan nilai A.
  2. d. if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)

Ini karena kondisi memeriksa apakah **ketiga nilai** (UTS, Tugas, dan Quiz) harus lebih besar atau sama dengan 80 untuk mendapatkan nilai A. Kondisi lainnya memungkinkan kelulusan jika hanya salah satu atau kombinasi dua nilai yang memenuhi syarat, padahal syarat harus ketiganya.

* 1. Flowchard ditampilkan dengan gambar

**[No.1] Algoritma dan Kode Program**

1. Algoritma

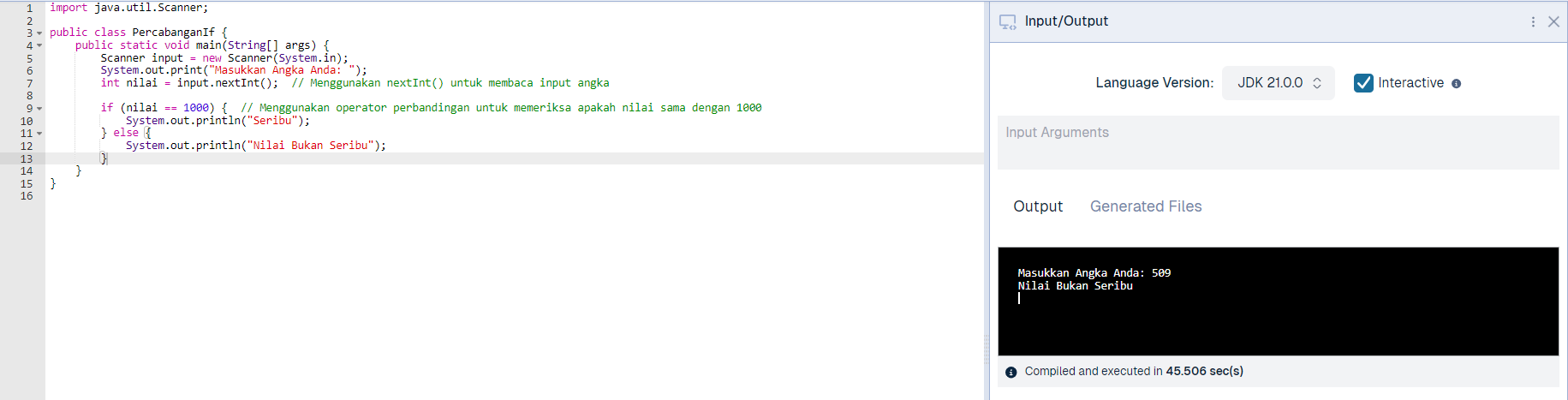
Contoh 1:

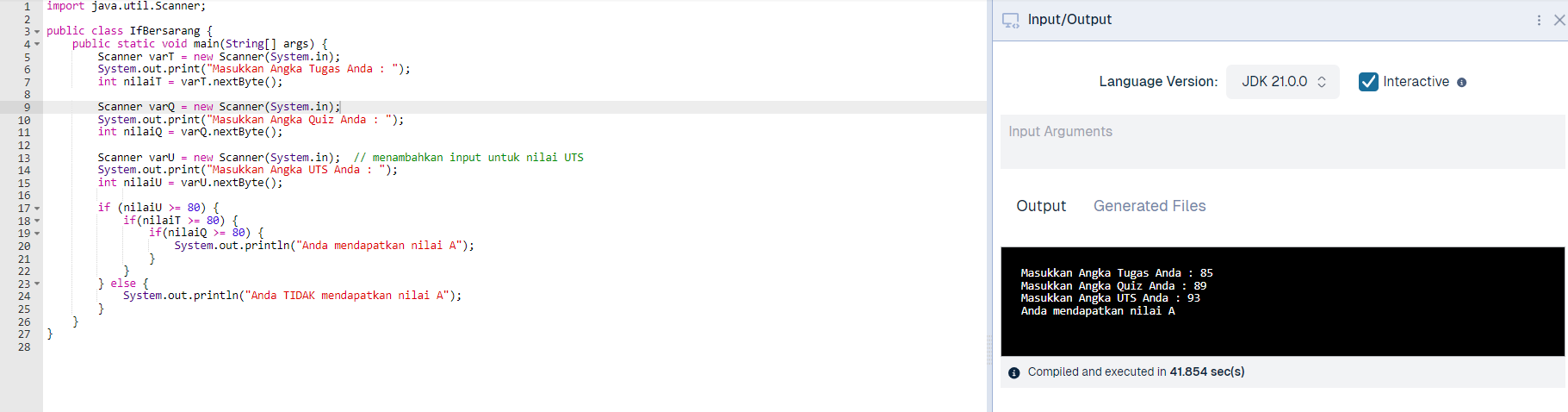
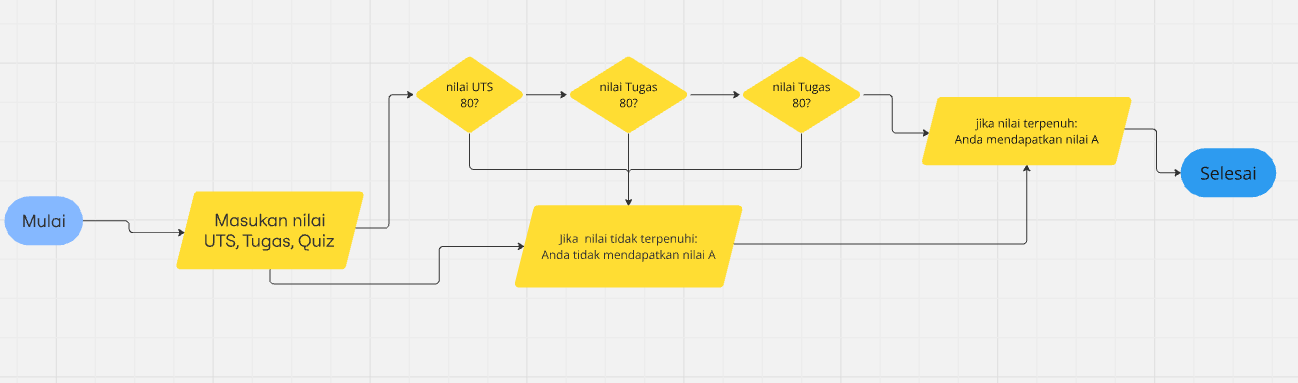
* Mulai program
* Mendeklarasikan import java.util.Scanner
* Mendeklarasikan public class percabangan
* Mendeklarasikan public static void main(String[] args) didalam public class percabangan
* Mendeklarasikan new scanner(system.in) didalam public static void main(String[] args)
* Mendeklarasikan system.out.println didalam public static void main(String[] args)
* Mendeklarasikan if else didalam public static void main(String[] args)
* Program selesai

Contoh 2:

* Mulai program
* Mendeklarasikan import java.util.Scanner
* Mendeklarasikan public class ifBersarang
* Mendeklarasikan public static void main (String[] args) didalam public class ifBersarang
* Mendeklarasikan new Scanner(System.in) didalam public static void main (String[] args)
* Mendeklarasikan Sistem.out.println() didalam public static void main (String[] args)
* Mendeklarasikan variable int nilai didalm public static void main (String[] args)
* Mendaklarasikan if else didalam public static void main (String[] args)
* Program selesai

1. Kode program contoh 1:



1. Kode program contoh 2:  
   
2. Flowchard  
   

**[No.1] Kesimpulan**

Pada program itu saya menggunakan bentuk kelas public karena kelas public memungkinkan program diakses dari luar kelas dan dapat dieksekusi langsung oleh compailer. Kelas public juga merupakan standar ketika kita menulis program utama (main) dalam Java. Perbaikan program dengan menambahkan deklarasi variabel dan operator perbandingan yang benar karena struktur Java mengharuskan setiap variabel dideklarasikan terlebih dahulu sebelum digunakan, serta menggunakan operator perbandingan (==) untuk memeriksa kesamaan, bukan operator assignment (=).

**[No.2] Identifikasi Masalah:**

**Contoh 1:**

**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 = data.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 ");   
            break;  
}    }**

**contoh 2:**

**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: jumlahHari = 31; break;  
            case 2: if (tahun % 4 == 0) {  jumlahHari = 29; }  
                    else { jumlahHari = 28; }  
                    break;  
case 3: jumlahHari = 31; break;  
case 4: jumlahHari = 30; break;  
case 5: jumlahHari = 31; break;  
case 6: jumlahHari = 30; break;  
case 7: jumlahHari = 31; break;  
case 8: jumlahHari = 31; break;  
case 9: jumlahHari = 30; break;  
case 10: jumlahHari = 31; break;  
case 11: jumlahHari = 30; break;  
case 12: jumlahHari = 31; break;  
                    default: System.out.println("Maaf bulan hanya sampai 12.");  
                    break;  
            }  
            System.out.println("Jumlah hari = " + jumlahHari);  
}    }**

**[No.2] Analisis dan Argumentasi**

2.1. Cermati kode pada Contoh 3.   
       Evaluasi penyebab kesalahan dan perbaiki kode tersebut!  
       Hapuslah kode break; pada //baris 1, lalu eksekusi kembali.   
       Kemudian hapuslah kode break; pada //baris 2, lalu eksekusi kembali.  
       Simpulkan kegunaan break pada switch!

2.2. Cermati kode pada Contoh 4. Evaluasi apakah penulisan kode tersebut sudah efisien?   
       Apakah ada penulisan informasi yang diulangi?   
       Jika ada, rekomendasikan penulisan yang lebih tepat!

2.3.  Cermati permasalahan yang dipecahkan pada Contoh 3.   
        Apakah masalah ini bisa diubah menjadi perintah IF?   
        Jika bisa, rekomendasikan bentuk perintah IF dari Contoh 3!   
        Simpulkan perbandingan masalah yang dapat diselesaikan percabangan dengan IF atau SWITCH

2.4.  Desain gambar flowchart dari Latihan 2.2. dan Latihan 2.3!

Jawaban:

2.1. Penyebab Kesalahan**:**

1. Baris char data = data.next().charAt(0); salah karena variabel data belum dideklarasikan. Variabel yang benar harusnya masukData.
2. Tanda titik dua (:) pada switch(data): salah, seharusnya tanda kurung kurawal { untuk memulai blok switch.
3. case A harus menggunakan tanda kutip tunggal ('A') karena 'A' adalah karakter, bukan variabel.
4. Pada default, tidak ada tanda kurung kurawal { atau :, menyebabkan kesalahan sintaks.

Kegunaan break**:** Jika break dihapus pada suatu case, eksekusi akan melanjutkan ke case berikutnya tanpa memperhatikan nilai switch. Hal ini disebut "fall-through". break diperlukan untuk menghentikan eksekusi setelah setiap case

2.2. Efisiensi: Penulisan jumlah hari pada setiap bulan diulang-ulang. Ini bisa dioptimalkan dengan mengelompokkan bulan yang memiliki jumlah hari yang sama dalam satu case. Dengan menggabungkan beberapa case yang memiliki jumlah hari yang sama, kode menjadi lebih ringkas dan mudah dipahami.

2.3. Perbandingan IF dan SWITCH**:**

* switch lebih efisien dan mudah dibaca jika terdapat banyak pilihan tetap (seperti dalam contoh A atau B).
* if lebih fleksibel karena bisa menangani ekspresi kompleks, namun bisa kurang efisien jika terdapat banyak pilihan tetap.

2.4. flowchard ditampilkan dengan gambar

**[No.2] Algoritma dan Kode Program**

1. Algoritma

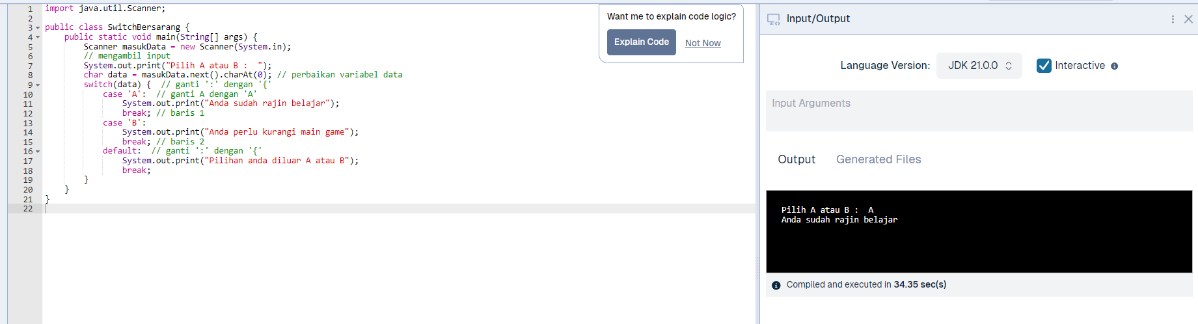
Contoh 1:

* Mulai program
* Mendeklarasikan import java.util.Scanner
* Mendeklarasikan public class switchBersarang
* Mendeklarasikan public static void main(String[] args) didalam public class switchBersarang
* Menklarasikan new Scanner (System.in) didalam public static void main(String[] args)
* Mendeklarasikan System.out.println() didalam public static void main(String[] args)
* Mendeklarasikan variable char dan switch didalam public static void main (String[] args)
* Mendeklarasikan case dan break didalam switch data
* Program selesai

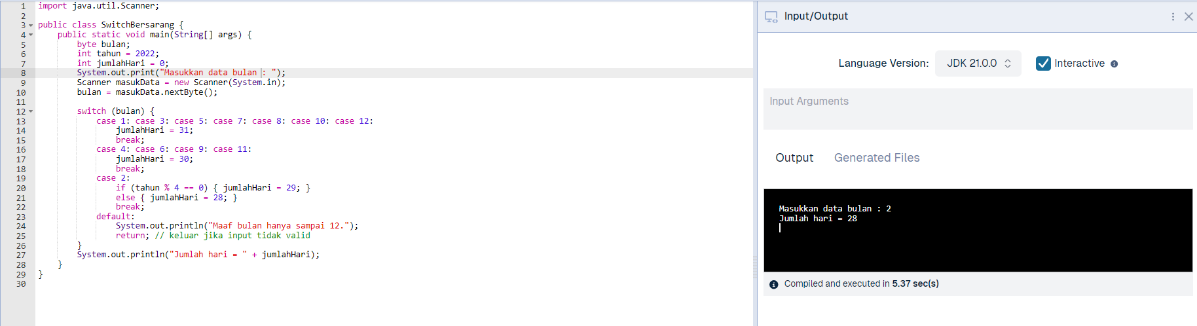
Contoh 2:

* Mulai program
* Mendeklarasikan import java.util.Scanner
* Mendeklarasikan public class switchBersarang
* Mendeklarasikan public static void main(String[] args) didalam public class switchBersarang
* Mendeklarasikan variable byte dan int didalam public static void main(String[] args)
* Menklarasikan new Scanner (System.in) didalam public static void main(String[] args)
* Mendeklarasikan System.out.println() didalam public static void main(String[] args)
* Mendeklarasikan case dan break didalam switch data
* Program selesai

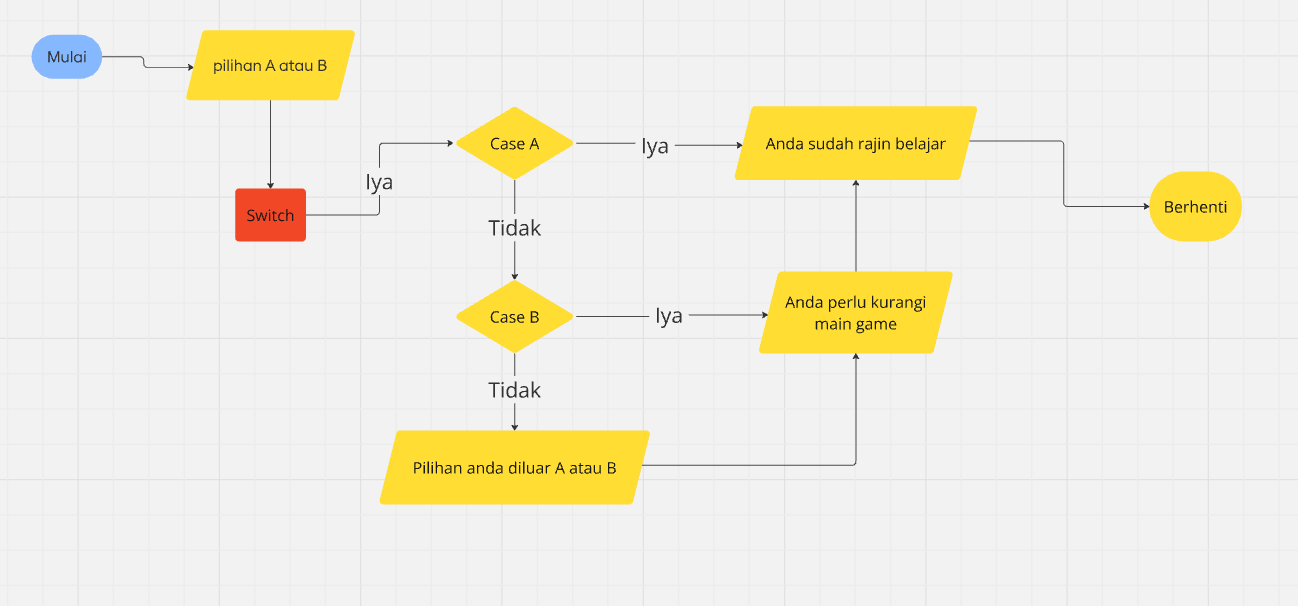
1. Kode program contoh 1:



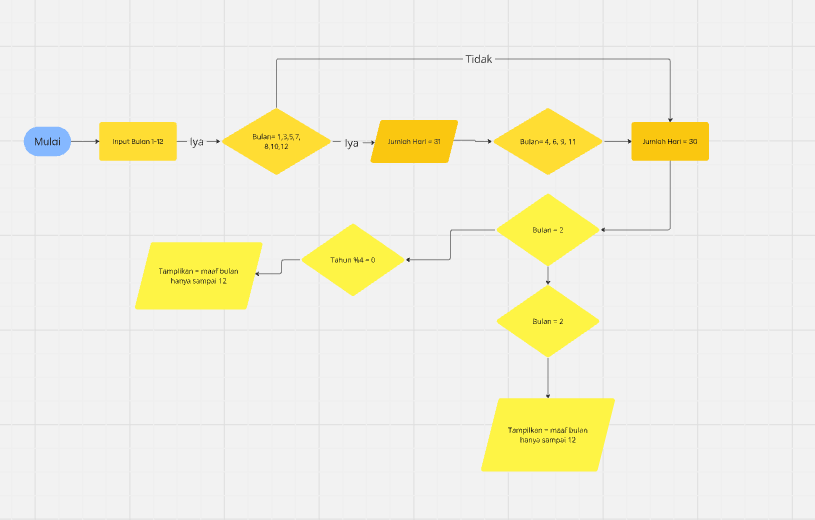
1. Kode program contoh 2:



1. Flowchard 2.1



1. Flowchard 2.2



**[No.2] Kesimpulan**

Pada program itu saya menggunakan bentuk kelas public karena kelas ini perlu diakses dari luar file atau package yang berbeda, terutama saat menjalankan program utama (main method). Perbaikan program dengan menambahkan tanda kurung kurawal {} dan penulisan karakter case dengan tanda kutip ' ' karena struktur Java mengharuskan penggunaan tanda kurung kurawal untuk memulai dan mengakhiri blok kode, serta penulisan karakter harus menggunakan tanda kutip tunggal untuk membedakan antara karakter literal dan variabel.