Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MUHAMMAD ARIQ HAFIDL FATANAH G1F024014** | **IF & SWITCH** | **23 September 2024** |

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

1. Uraikan permasalahan dan variabel

Tuliskan kembali soal:

**Contoh 1:** Salin dan tempel kode program berikut ke Eclipse.

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");  
        }    }  
}

**Luaran Contoh 1:**  
Exception in thread "main" java.lang.Error: Unresolved compilation problems:       
    nilai cannot be resolved to a variable  
    masuk cannot be resolved  
    nilai cannot be resolved to a variable

    at PercabanganIf.main(PercabanganIf.java:7)

**Contoh 2:** Salin dan tempel kode program berikut ke Eclipse.

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");  
    }  
    }  
}

**Luaran Contoh 2:**  
Masukkan Angka Tugas Anda : 70  
Masukkan Angka Quiz Anda : 70  
Masukkan Angka UTS Anda : 70  
Anda TIDAK mendapatkan nilai A

**Latihan 1:**

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!

Pada kode program contoh satu masih ada kesalahan yang perlu diperbaiki dan begitu juga dengan kode program contoh 2

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

1. Saya mengusulkan permasalahan pada contoh 1 adalah dengan mengganti deklarasi tipe data yang semula byte menjadi int agar nilai dari variable dapat sesuai.

Begitupun juga untuk contoh 2, perbaikan kode program dengan mengganti tipe data yang lebih sesuai yaitu int dan mengubah nilaiU menjadi nilaiQ

1. Alasan solusi ini karena menyesuaikan nilai dari variable yang ada dimana byte kurang tepat untuk menyimpan rentang nilai 32 ke atas sehingga menggantinnya dengan tipe data int agar dapat tersimpan dengan baik
2. Perbaikan kode program dengan cara mengganti nextByte() dengan nextInt():

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

1. Algoritma contoh 1
2. Mulai Program
3. Import Paket Scanner

Memanggil paket yang diperlukan untuk membaca input dari pengguna.

1. Inisialisasi Scanner

Buat objek Scanner untuk mengambil input dari pengguna.

1. Tampilkan Pesan Input

Tampilkan pesan "Masukkan Angka Anda: ".

1. Baca Input Pengguna

Simpan input pengguna ke dalam variabel nilai sebagai tipe data int.

1. Cek Kondisi

Jika nilai sama dengan 1000, lakukan langkah berikut:

Tampilkan pesan "Seribu".

Jika tidak, lakukan langkah berikut:

Tampilkan pesan "Nilai Bukan Seribu".

1. Akhiri Program.
2. Algoritma contoh 2
3. Mulai
4. Impor Kelas Scanner
5. Deklarasikan Variabel:

nilaiT untuk menyimpan nilai tugas.

nilaiQ untuk menyimpan nilai kuis.

1. Buat Instance Scanner untuk membaca input dari pengguna.
2. Tampilkan Pesan: "Masukkan Angka Tugas Anda:"
3. Baca Input dan simpan ke nilaiT.
4. Tampilkan Pesan: "Masukkan Angka Quiz Anda:"
5. Baca Input dan simpan ke nilaiQ.
6. Jika nilaiQ >= 80 maka:
7. Jika nilaiT >= 80 maka:

Tampilkan: "Anda mendapatkan nilai A"

Selain itu:

Tampilkan: "Anda TIDAK mendapatkan nilai A"

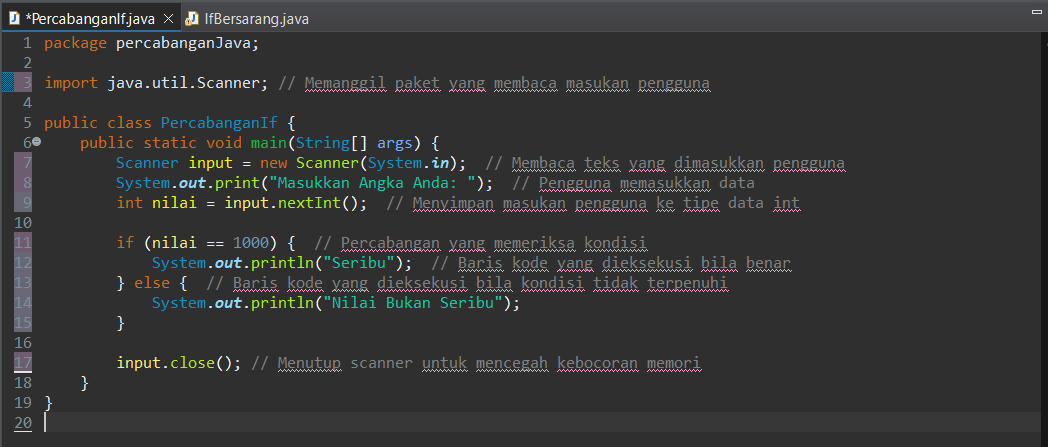
Selain itu:

1. Tampilkan: "Anda TIDAK mendapatkan nilai A"
2. Selesai
3. Kode program dan luaran

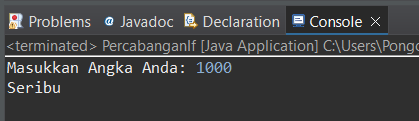
**CONTOH 1**

1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

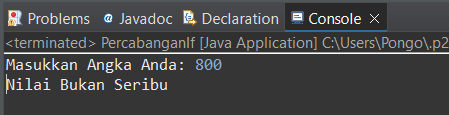


**Gambar 1. Contoh 1**

1. Analisa luaran yang dihasilkan   
   

**Gambar 2. Output Jika 1000**

Luaran sudah sesuai dengan program yang disusun. Ketika pengguna mengisi nilai 1000 maka yang ditampilkan adalah “Seribu”. Alasannya karena Ketika pengguna menginput 1000, percabangan yang memeriksa langsung menampilkan “Seribu” sesuai dengan int(nilai==1000)

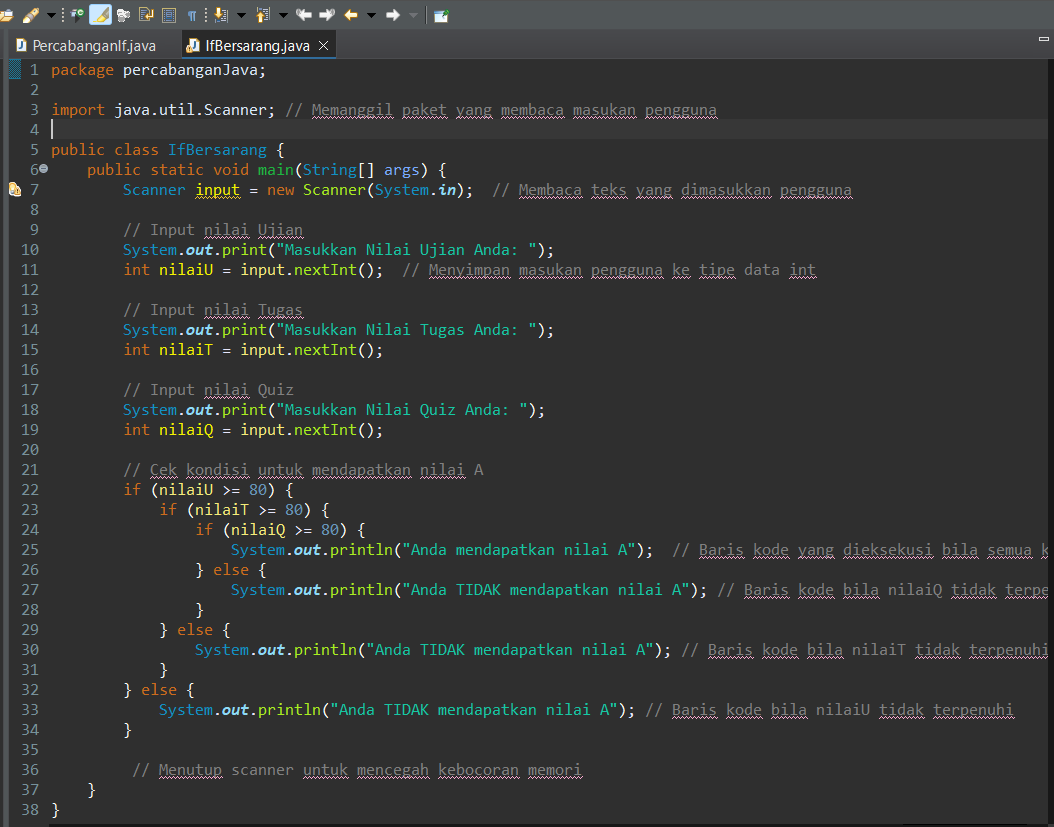


**Gambar 3. Output Jika bukan 1000**Luaran sudah sesuai dengan program yang disusun. Ketika pengguna mengisi nilai 1000 maka yang ditampilkan adalah “Nilai Bukan Seribu”. Ketika pengguna tidak memasukan nilai 1000 maka percabangan akan membaca kode ke else dan menampilkan output “Nilai Bukan Seribu”.

**CONTOH 2**

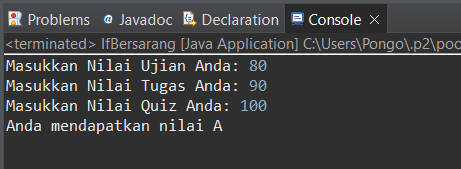
1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

****

**Gambar 4. Kode program nilai U,T,Q**

b). Analisa luaran yang dihasilkan

****

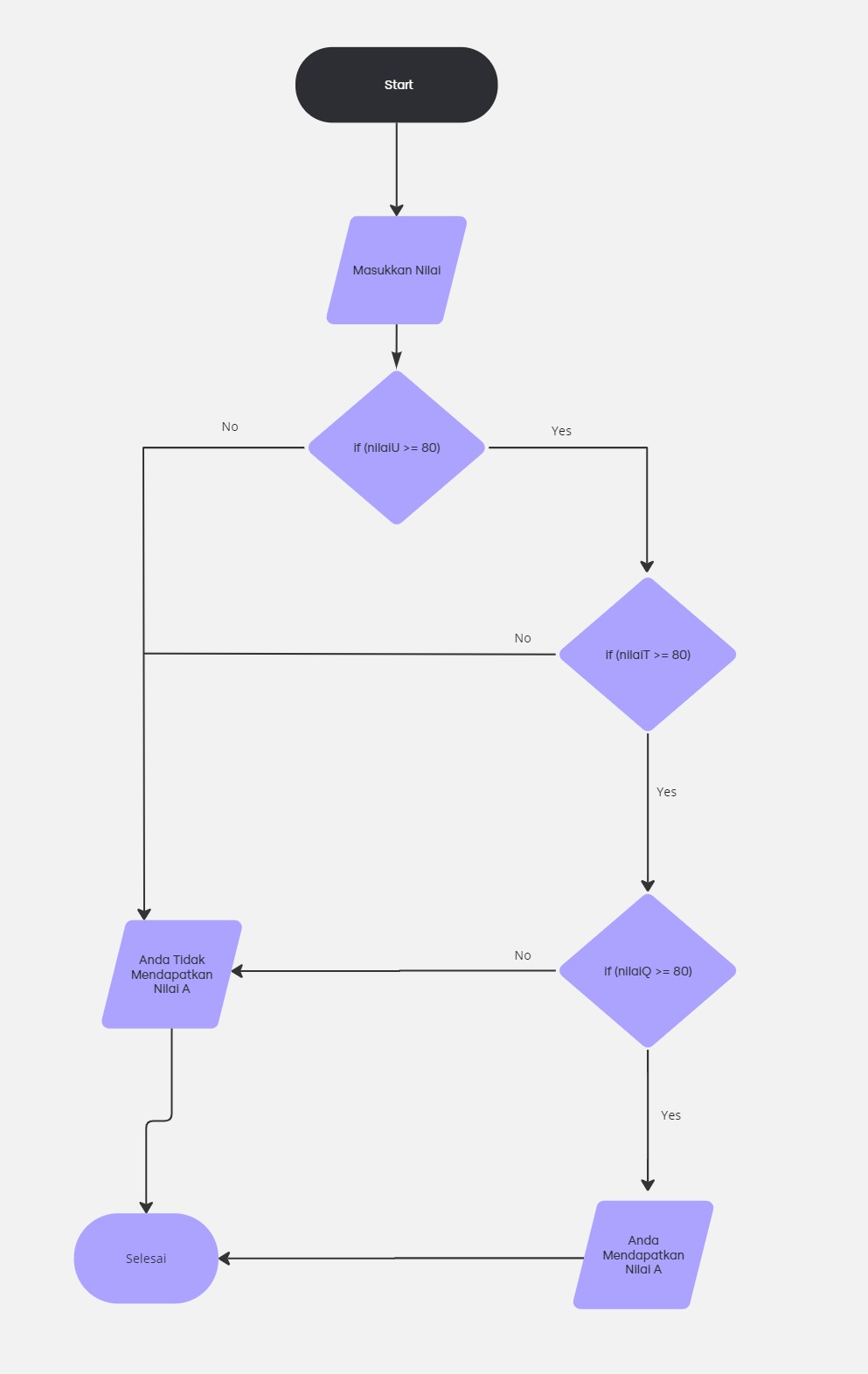
**Gambar 5. Output nilai U,T,Q**

**Jawab 1.3**

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)

Dari analisis di atas, pilihan a (IF (nilaiU >= 80 || nilaiT >= 80 || nilaiQ >= 80)) adalah kondisi yang paling tepat untuk menggantikan ketiga kondisi tersebut. Ini mencakup semua kemungkinan di mana setidaknya satu dari nilai tersebut memenuhi syarat.

**1.4.    Uraikan gambar diagram flowchart dari Latihan 1.2!**

****

**Gambar 6 Flowchart**

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Pada program itu saya menganalisa bahwa masih ada keselahan dari kode program. Kesimpulan dari permasalahan dalam contoh kode program adalah terdapat kesalahan pada deklarasi variabel dan logika percabangan yang perlu diperbaiki agar program berfungsi dengan baik. Di Contoh 1, variabel `nilai` dan objek `Scanner` salah penamaan, sedangkan di Contoh 2, variabel `nilaiU` tidak dideklarasikan. Penting untuk mendeklarasikan semua variabel sebelum digunakan dan memastikan logika percabangan menangani semua kondisi. Pengambilan keputusan didasarkan pada kebutuhan untuk menghindari error kompilasi dan memastikan program dapat memberikan output yang tepat, sehingga meningkatkan kualitas dan keandalan kode.

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

1. Uraikan permasalahan dan variabel

Tuliskan kembali soal:

**Contoh 3:**Salin dan tempel kode program berikut ke Eclipse.

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;  
}    }

**Luaran Contoh 3:**  
Exception in thread "main" java.lang.Error: Unresolved compilation problems:   
    Cannot invoke next() on the primitive type char  
    Syntax error on token ":", { expected  
    Syntax error, insert ":: IdentifierOrNew" to complete ReferenceExpression  
    Syntax error, insert ":" to complete SwitchLabel  
    A cannot be resolved to a variable  
    Syntax error on token "default", } expected

    at SwitchBersarang.main(SwitchBersarang.java:8)

**Contoh 4:**Salin dan tempel kode program berikut ke Eclipse.

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);  
}    }

**Luaran Contoh 4:**

Masukkan data bulan (dalam angka): 7  
Jumlah hari = 31

**Latihan 2:**  
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!

**[No.2] Analisis dan Argumentasi (Contoh 3)**

2.1 Saya mengusulkan permasalahan ini dapat diatasi dengan menganalisis kode sehingga dapat menemukan kesalahan dari kode tersebut, dari analysis tersebut saya menemukan permasalan pada Contoh 3 yaitu :

1. syntax pada kode char data = data.next().charAt(0); diganti menjadi char data = masukData.next().charAt(0);
2. case A menjadi case 'A':
3. switch(data): menjadi switch(data) {
4. terakhir tanda } kurang 1

Kemudian jika kode break; dihapus maka output yang dikeluarkan akan berantakan jadi dapat kita simpulkan bahwa kode break; berfungsi sebagai sekat pembatas antara kode program awal dan seterusnya.

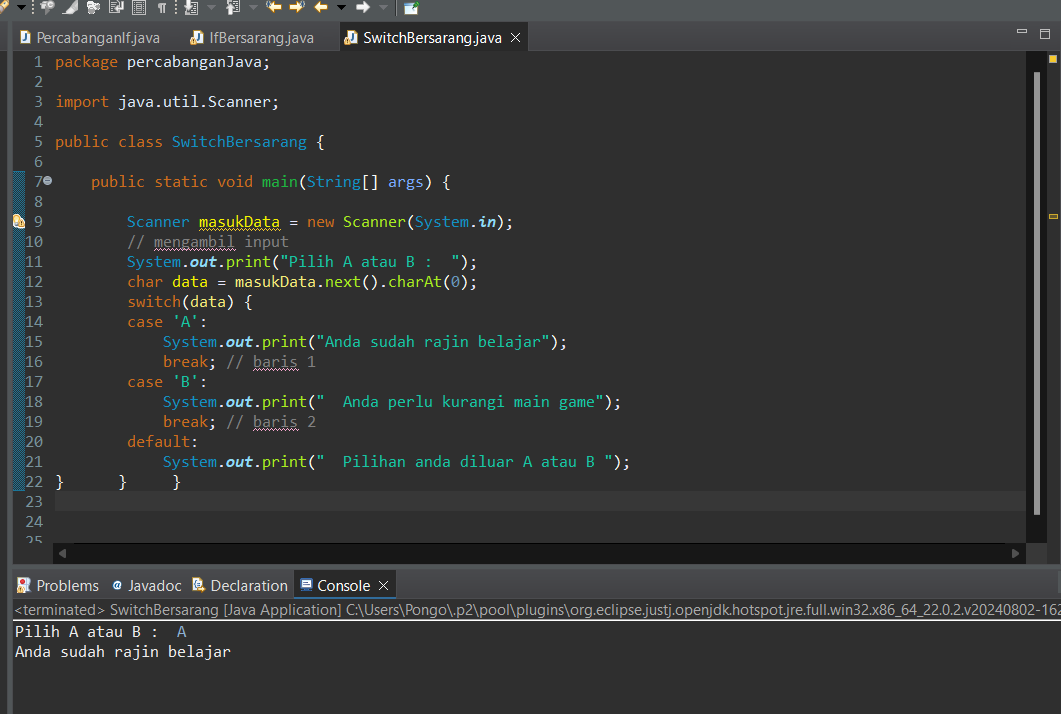
1. Alasan solusi ini karena agar program dapat berjalan dengan lancar.
2. Perbaikan kode program dengan cara :

permasalan dapat diatasi dengan :

1. syntax pada kode char data = data.next().charAt(0); diganti menjadi char data = masukData.next().charAt(0);
2. case A menjadi case 'A':
3. switch(data): menjadi switch(data) {
4. terakhir tanda } kurang 1

**[No.2] Penyusunan Algoritma dan Kode Program (Contoh 3)**

1. Algoritma
2. Mulai
3. Import package
4. Import class
5. Buat static void
6. Kerjakan kodingan
7. Selesai
8. Kode program dan luaran
9. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

1. Analisa luaran yang dihasilkan   
   Analisis keluaran dari program SwitchBersarang menunjukkan bahwa program meminta pengguna untuk memilih antara opsi 'A' atau 'B'. Jika pengguna memasukkan 'A', program akan menampilkan pesan "Anda sudah rajin belajar"; jika memilih 'B', akan muncul pesan "Anda perlu kurangi main game". Apabila input tidak sesuai (bukan 'A' atau 'B'), program akan memberikan pesan bahwa pilihan berada di luar opsi yang tersedia. Program ini menggunakan struktur switch-case untuk menangani input, sehingga menghasilkan output yang sesuai berdasarkan pilihan pengguna dengan penanganan yang jelas untuk opsi yang tidak valid.

**[No.2] Analisis dan Argumentasi (Contoh 4)**

2.2 Menurut saya kode tersebut sudah efesien, kode ini sudah cukup baik dalam hal efisiensi dan jelas dalam logika, sehingga mudah dipahami dan dipelihara.

2.3 Menurut saya bisa menjadi perintah IF, rekomendasi kodenya:

if (data == 'A') {

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

} else if (data == 'B') {

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

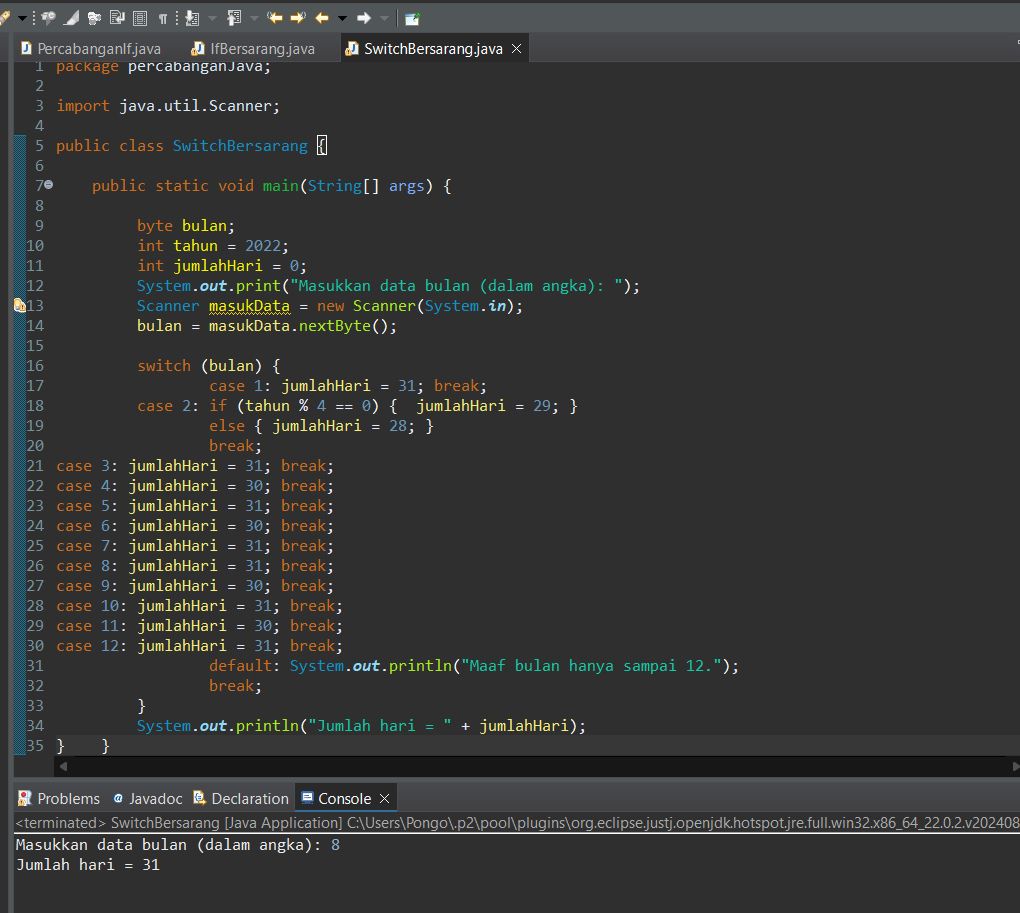
} else {

System.out.print("Pilihan anda diluar A atau B");

}

**[No.2] Penyusunan Algoritma dan Kode Program (Contoh 4)**

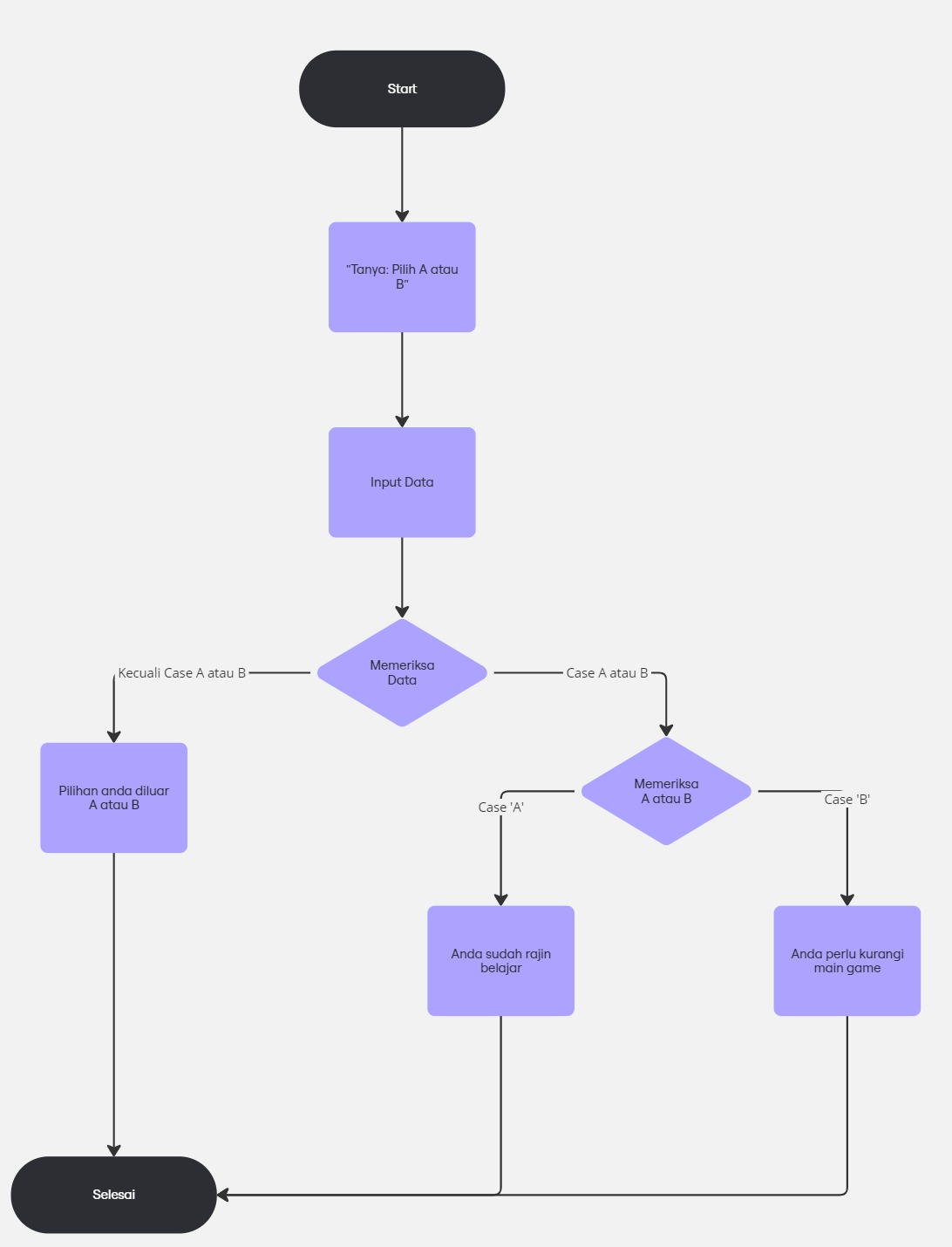
1. Algoritma
2. Mulai
3. Import package
4. Import class
5. Buat static void
6. Kerjakan kodingan
7. Selesai
8. Kode program dan luaran
9. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot  


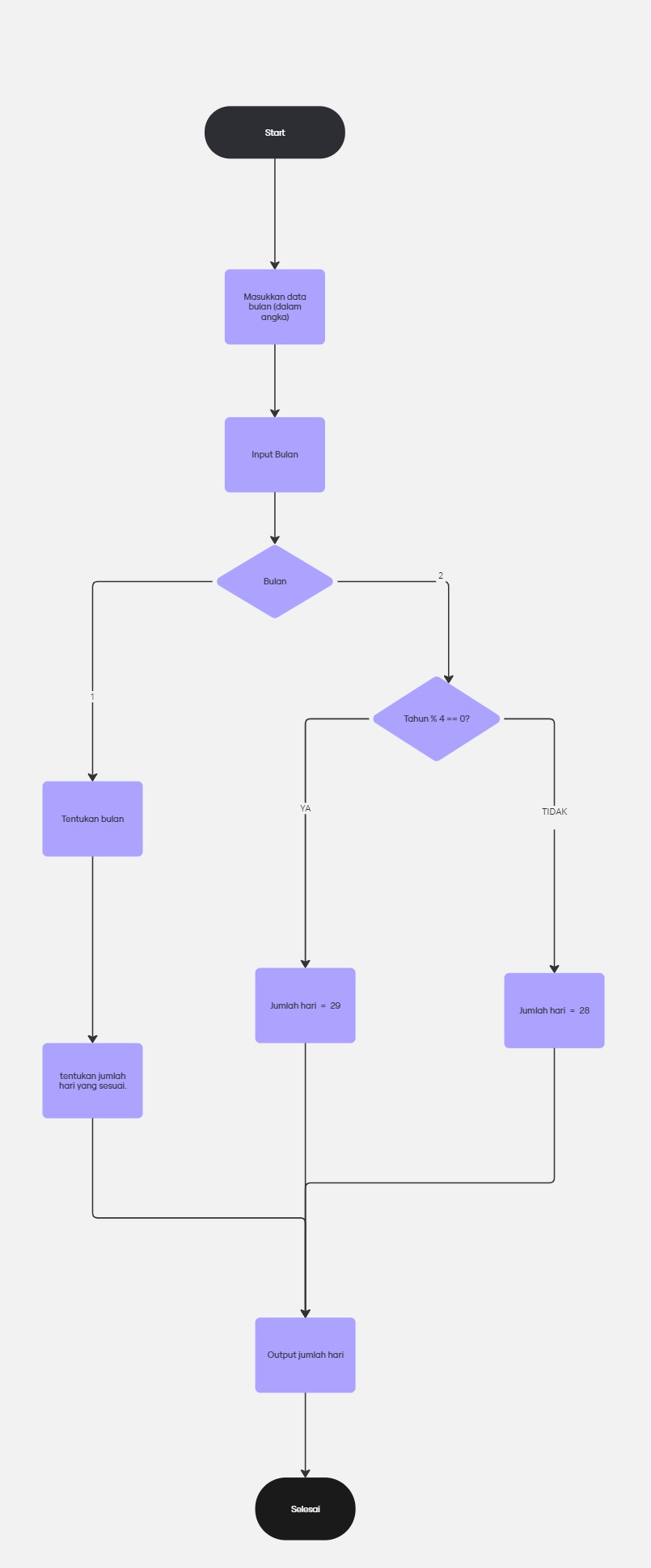
1. Analisa luaran yang dihasilkan   
   Program ini menghasilkan keluaran yang menunjukkan jumlah hari dalam bulan yang dimasukkan oleh pengguna berdasarkan input numerik. Jika pengguna memasukkan angka yang valid antara 1 hingga 12, program akan mengkalkulasi dan mencetak jumlah hari sesuai dengan bulan tersebut, termasuk penanganan tahun kabisat untuk bulan Februari. Jika input tidak valid, program memberikan pesan kesalahan yang menjelaskan bahwa bulan hanya sampai 12. Dengan demikian, keluaran program memberikan informasi yang akurat dan jelas, sesuai dengan input yang diberikan, serta menangani kondisi tidak valid dengan baik.

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

**1). Flowchart 2.2**

****

**2) Flowchart 2.3**



**[No.2] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Kesimpulan dari analisis permasalahan, algoritma, dan kode program pertama (menggunakan `switch`) dan kedua (menggunakan `if`) adalah bahwa keduanya efektif dalam menentukan jumlah hari dalam bulan berdasarkan input pengguna, dengan algoritma yang serupa dan penanganan tahun kabisat pada bulan Februari. Kode pertama lebih terstruktur dan mudah dibaca, sedangkan kode kedua lebih fleksibel untuk kondisi kompleks. Dasar pengambilan keputusan dalam memilih antara `switch` dan `if` bergantung pada konteks: `switch` lebih cocok untuk banyak pilihan tetap, sementara `if` lebih baik untuk logika yang lebih variatif. Pemilihan ini sejalan dengan kebutuhan spesifik program dan keseimbangan antara keterbacaan dan fleksibilitas.

**Refleksi**

Hari ini, saya belajar tentang struktur percabangan `switch` dan `if` dalam pemrograman. Saya mengeksplorasi cara keduanya digunakan untuk menentukan jumlah hari dalam bulan berdasarkan input pengguna. Saya memahami bahwa `switch` lebih terstruktur dan mudah dibaca, sementara `if` lebih fleksibel untuk kondisi kompleks. Pembelajaran ini membuat saya menyadari pentingnya memilih alat yang tepat untuk meningkatkan keterbacaan dan efisiensi kode, dan saya merasa lebih percaya diri untuk menerapkannya di proyek mendatang.