

Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Rayhan Prabowo G1F024022	If dan Switch	26 September 2024
[Nomor Soal] Identifikasi Masalah:		
<ol style="list-style-type: none">1) Uraikan permasalahan dan variabel2) Rincikan sumber informasi yang relevan (buku / webpage)3) Uraikan rancangan solusi yang diusulkan (jika ada).4) Analisis susunan solusi, parameter solusi (jika ada).		
[Nomor Soal] Analisis dan Argumentasi		
<ol style="list-style-type: none">1) Uraikan rancangan solusi yang diusulkan.2) Analisis solusi, kaitkan dengan permasalahan.		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<ol style="list-style-type: none">1) Rancang desain solusi atau algoritma2) Tuliskan kode program dan luaran<ol style="list-style-type: none">a) Beri komentar pada kodeb) Uraikan luaran yang dihasilkanc) Screenshot/ Capture potongan kode dan hasil luaran		
[Nomor Soal] Kesimpulan		
<ol style="list-style-type: none">1) Analisa<ol style="list-style-type: none">a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?2) Evaluasi<ol style="list-style-type: none">a) Apa konsekuensi dari skenario pemrograman ini?b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)3) Kreasi<ol style="list-style-type: none">a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)		

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

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!

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan kode program pada **Contoh 1** dapat diatasi dengan :
 - a. Mengubah nilai = masuk.nextByte(); ke int nilai = input.nextInt();

- b. Mengubah nama variabel untuk objek dari 'masuk' ke 'input'
 - c. Mengubah nextByte() ke nextInt() karena bilangan yang dideklarasikan yaitu 1000
 - d. Menambahkan input.close(); di akhir method utama untuk menutup objek Scanner dengan benar dan mencegah kebocoran sumber daya.
 - e. Menghapus komentar yang tidak perlu
 - f. Mengubah if(nilai = 1000) menjadi if (nilai == 1000), karena = adalah operator penugasan sedangkan == digunakan untuk perbandingan.
- 2) Saya mengusulkan permasalahan kode program pada **Contoh 2** dapat diatasi dengan :
- a. Mengubah nextByte() menjadi nextInt() untuk kedua input ,car aini memungkinkan nilai rentang input bisa lebih luas.
 - b. Lebih baik Menggunakan satu objek Scanner daripada membuat beberapa objek Scanner, ini bisa lebih efisien dan mencegah kebocoran sumber data.
 - c. Memperbaiki nama variabel pada pernyataan if ,awalnya if (nilaiU >= 80) menjadi if (nilaiQ >=80),karena tidak ada nilai U yang didefinisikan.
 - d. Memperbaiki pernyataan if-else untuk logika yang lebih baik : sekarang memeriksa nilai kuis dan tugas sebelum memberikan nilai A.Jika nilai kuis atau tugas dibawah 80, itu akan mencetak bahwa siswa tidak mendapatkan nilai A.
- 3) Alasan solusi ini karena pada kedua kode program diatas terdapat beberapa masalah logika dan sintaksis dalam kode asli,jadi untuk meningkatkan efisiensi dan keterbacaan program seharusnya kesalahan kode dan sintaksis diatas harus di perbaiki.Ini juga dapat memastikan bahwa program berjalan sesuai dengan yang diharapkan dan mengikuti praktik pemrograman java yang baik. _____
- 4) Perbaiki kode program dengan cara memperbaiki kesalahan sintaksis dan kode program,menambahkan apa yang kurang dari program diatas dan juga menghapus yang tidak berguna pada program tersebut.Berikut ini kode program pada kedua program diatas yang sudah diperbaiki:

Contoh 1(Setelah perbaikan)

```
import java.util.Scanner;

public class PercabanganIf {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan Angka Anda : ");
        int nilai = input.nextInt();
```

```

        if (nilai == 1000) {
            System.out.println("Seribu");
        } else {
            System.out.println("Nilai Bukan Seribu");
        }

        input.close();
    }
}

```

Contoh 2(Setelah Perbaikan)

```

import java.util.Scanner;

public class IfBersarang {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Masukkan Angka Tugas Anda : ");
        int nilaiT = input.nextInt();

        System.out.print("Masukkan Angka Quiz Anda : ");
        int nilaiQ = input.nextInt();

        if (nilaiQ >= 80) {
            if (nilaiT >= 80) {
                System.out.println("Anda mendapatkan nilai A");
            } else {
                System.out.println("Anda TIDAK mendapatkan nilai A");
            }
        } else {
            System.out.println("Anda TIDAK mendapatkan nilai A");
        }

        input.close();
    }
}

```

[No.1] Penyusunan Algoritma, Kode Program, dan diagram Flowchart

1) Algoritma

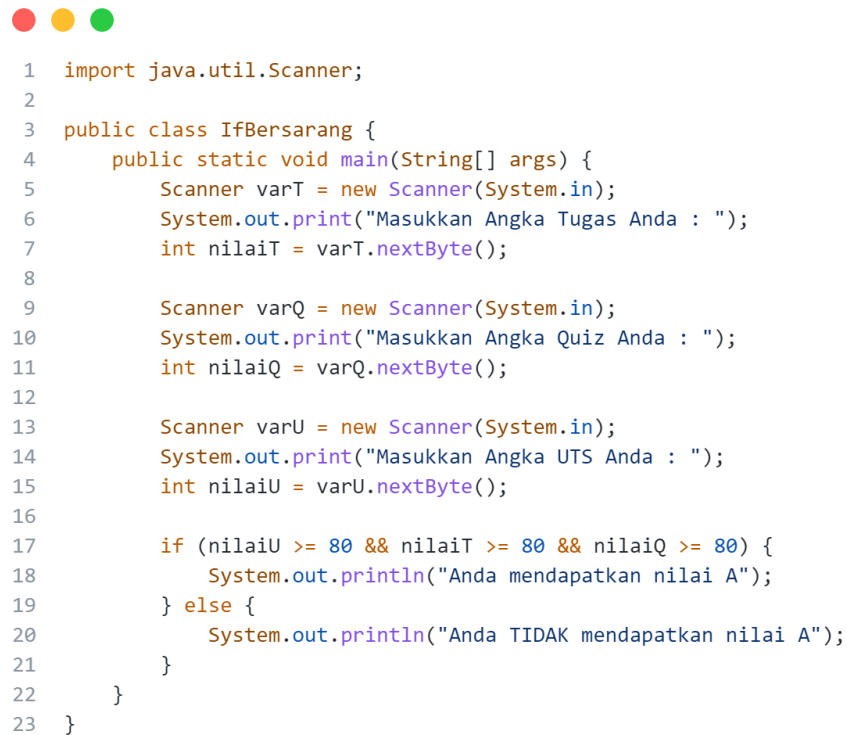
Algoritma adalah langkah-langkah penyelesaian masalah.

Algoritma program If bersarang:

- (a) Mulai.
- (b) Deklarasikan variabel nilaiT untuk menyimpan nilai tugas.
- (c) Deklarasikan variabel nilaiQ untuk menyimpan nilai quiz.
- (d) Deklarasikan variabel nilaiU untuk menyimpan nilai UTS.
- (e) Tampilkan pesan "Masukkan Angka Tugas anda :".
- (f) Baca input dari pengguna dan simpan kedalam nilaiT.
- (g) Tampilkan pesan "Masukkan Angka Quiz Anda :".
- (h) Baca input dari pengguna dan simpan kedalam nilaiQ.
- (i) Tampilkan pesan "Masukkan Angka UTS Anda :".
- (j) Baca input dari pengguna dan simpan kedalam nilaiU.
- (k) Jika nilaiU ≥ 80 dan nilaiT ≥ 80 dan nilaiQ ≥ 80 : Tampilkan "Anda mendapatkan nilai A".
- (l) Jika tidak : tampilkan "Anda TIDAK mendapatkan nilai A".
- (m) Selesai.

2) Kode program dan luaran

a) Kode Program

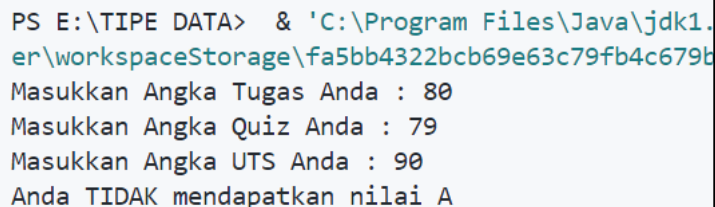


```

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         } else {
20             System.out.println("Anda TIDAK mendapatkan nilai A");
21         }
22     }
23 }

```

b) Luaran



```

PS E:\TIPE DATA> & 'C:\Program Files\Java\jdk1.
er\workspaceStorage\fa5bb4322bcb69e63c79fb4c679b
Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 79
Masukkan Angka UTS Anda : 90
Anda TIDAK mendapatkan nilai A

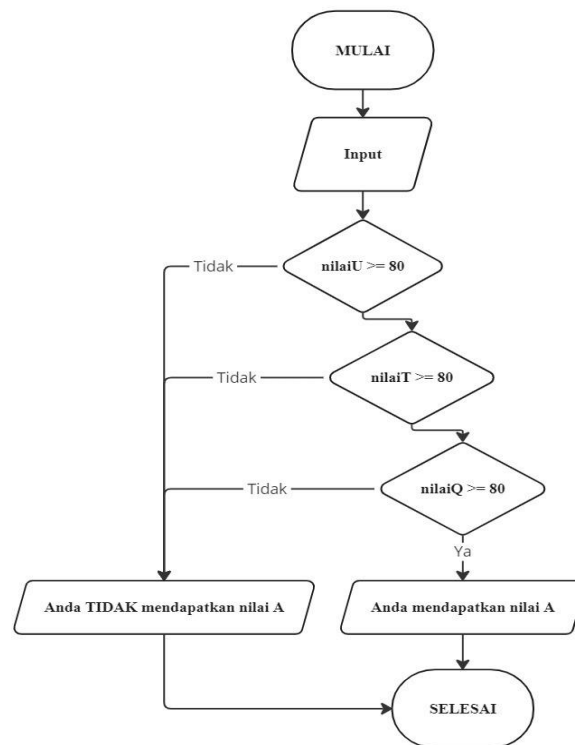
```

c) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang dijalankan. Tipe data yang digunakan pada program yaitu tipe data integer. Alasan mendapatkan luaran "Anda TIDAK mendapatkan nilai A" yaitu karena nilaiU, nilaiT, nilaiQ harus memenuhi jumlah nilai 80 semuanya, walaupun ada satu nilai yang lebih dari 80 tetapi kedua nilai sisanya tidak lebih dari 80 bahkan kurang dari 80, maka luaran yang dihasilkan akan tetap sama yaitu Anda TIDAK mendapatkan nilai A. Singkatnya kurang lebih seperti itu.

3) Diagram flowchart

Diagram flowchart untuk if bersarang adalah representasi visual dari alur logika sebuah program yang menggunakan struktur pengambilan keputusan bertingkat atau bersarang. Berikut pengimplementasian Diagram flowchart dari program yang saya buat :



[No.1] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- 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)`

- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab :

Program yang menggunakan IF bersarang adalah struktur data pemrograman yang memungkinkan pemngembang untuk membuat keputusan berdasarkan beberapa kondisi yang saling terkait. Dalam contoh program yang telah dibahas, pengguna diminta untuk memasukkan nilai tugas, Quiz, dan UTS. Program kemudian memeriksa apakah semua nilai tersebut memenuhi syarat tertentu

(lebih besar atau sama dengan 80) untuk menentukan apakah siswa mendapatkan nilai A. Penggunaan If bersarang sangat berguna dalam situasi dimana keputusan yang diambil bergantung pada beberapa faktor, dan dapat membantu dalam menyederhanakan kode dengan mengelompokkan kondisi yang terkait. Modifikasi IF bersarang dapat dilakukan untuk menambah kompleksitas logika program, seperti menambahkan lebih banyak kondisi atau mengubah cara evaluasi dilakukan. Pada program yang telah saya buat, yang dimana pada program yang saya buat ada tiga kondisi IF nya, maka saya akan memodifikasi ketiga kondisi IF tersebut menjadi satu kondisi. Berikut adalah kode ketiga Kondisi If tersebut sebelum di modifikasi:

```
if (nilaiU >= 80 ){  
    if (nilaiT >= 80) {  
        if (nilaiQ >= 80)
```

dan berikut kode kondisi If yang sudah dimodifikasi menjadi satu kondisi :

```
if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)
```

Dasar dari pengambilan keputusan modifikasi program diatas adalah dengan menggabungkan beberapa kondisi menjadi satu, kita dapat menyederhanakan kode dan meningkatkan keterbacaan program. Ini juga membantu dalam mengurangi kompleksitas logika yang terlibat dalam pengambilan keputusan, sehingga lebih mudah untuk dipahami dan dikelola.

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

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

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara :
 - a. Pada kode awal, terdapat kesalahan pada baris `char data = data.next().charAt(0);` . seharusnya kita menggunakan objek `Scanner` yang telah kita buat, yaitu `masukData`, bukan `dat`. Jadi baris yang benar adalah `char data = masukData.next().charAt(0);`.
 - b. Selanjutnya , struktur switch statement perlu diperbaiki. Pada kode awal, digunakan titik dua (:) setelah `switch(data)`, yang merupakan sintaks yang salah dalam java. Kita perlu menggantinya dengan kurung kurawal ({} untuk membuka blok switch
 - c. Pada bagian case, terdapat beberapa kesalahan sintaks, Untuk case pertama, seharusnya menggunakan tanda petik tunggal untuk karakter, jadi case A harus diubah menjadi case 'A':. Selain itu, setiap case harus diakhiri dengan titik dua (:).
 - d. Untuk memperbaiki format output `System.out.print()` menjadi `System.out.println()`. Ini akan menambahkan baris baru setelah setiap output, membuat hasil lebih mudah dibaca.
 - e. Terakhir penting untuk menutup `objekScanner` setelah kita selesai menggunakannya. Kita dapat menambahkan `masukData.close();` di akhir metode main untuk mencegah kebocoran sumber daya. _____
- 2) Alasan solusi ini karena dengan melakukan perbaikan diatas,kode akan berfungsi sebagai mestinya. Program akan menerima input 'A' atau 'B' dari pengguna, memberikan respons yang sesuai, atau menampilkan pesan default jika input tidak sesuai.kode juga akan lebih mudah terbaca dan lebih mudah dipahami.
- 3) Perbaikan kode program dengan cara mengganti atau memperbaiki kode yang salah seperti pada point 1,berikut kode yang sudah diperbaiki:

```
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.println("Anda sudah rajin belajar");
                break; // baris 1
            case 'B':
                System.out.println("Anda perlu kurangi main game");
                break; // baris 2
            default:
                System.out.println("Pilihan anda diluar A atau B");
                break;
        }

        masukData.close();
    }
}

```

4) Evaluasi efisiensi kode pada contoh 4 :

- a. Pengulangan informasi : Ada pengulangan dalam penentuan jumlah hari untuk bulan bulan tertentu. Misalnya, *jumlahHari = 31*; diulang beberapa kali untuk bulan bulan yang memiliki 31 hari
- b. Penggunaan tipe data: penggunaan byte untuk variabel bulan tidak begitu diperlukan. Int sudah cukup dan lebih umum digunakan.
- c. Struktur switch: Struktur switch bisa dioptimalkan dengan mengelompokkan case case yang memiliki hasil yang sama.

5) Rekomendasi perbaikan kode pada contoh 4 :

- a. Mengelompokkan case case dengan hasil yang sama :
 - i. Bulan dengan 30 hari : 4,6,9,11
 - ii. Bulan dengan 31 hari : 1,3,4,7,8,10,12
 - iii. Khusus untuk bulan 2(Februari)
- b. Menggunakan operator ternary untuk bulan februari untuk mempersingkat kode.
- c. Mengganti byte bulan menjadi int bulan untuk konsistensi dan menghindari potensi masalah dengan input yang lebih besar.

- d. Menambahkan `System.exit(0)` pada case default untuk menghentikan program jika input tidak valid.
- e. Menutup Scanner diakhir program untuk menghindari resource leak.

Dengan perbaikan ini, kode menjadi lebih efisien, mudah dibaca, dan mudah dipelihara. Pengulangan informasi telah diminimalkan, dan logika penentuan jumlah hari menjadi lebih terstruktur.

[No.2] Penyusunan Algoritma, Kode Program, dan Diagram Flowchart

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

Algoritma Program SwitchBersarang:

- (a) Mulai
- (b) Inisialisasi Scanner untuk menerima input dari pengguna
- (c) Tampilkan pesan "Pilih A atau B:"
- (d) Baca karakter pertama dari input pengguna
- (e) Gunakan struktur switch untuk memeriksa input:
 - a. Jika input adalah 'A' maka akan menampilkan "Anda sudah rajin belajar"
 - b. Jika input adalah 'B' maka akan menampilkan "Anda perlu kurangi main game"
 - c. Jika input bukan 'A' atau 'B' maka akan menampilkan "Pilihan anda diluar A atau B"
- (f) Tutup Scanner
- (g) Selesai.

2) Kode program dan luaran

- a) Kode Program Contoh 3 yang sudah diperbaiki

```
SwitchBersarang.java > SwitchBersarang
3  public class SwitchBersarang {
4      public static void main(String[] args) {
5          // mengambit input
6          Scanner masukData = new Scanner(System.in);
7          System.out.print(s:"Pilih A atau B: ");
8          char data = masukData.next().charAt(index:0);
9
10
11         switch(data) {
12             case 'A':
13                 System.out.println(x:"Anda sudah rajin belajar");
14                 break; // baris 1
15             case 'B':
16                 System.out.println(x:"Anda perlu kurangi main game");
17                 break; // baris 2
18             default:
19                 System.out.println(x:"Pilihan anda diluar A atau B");
20                 break;
21         }
22
23         masukData.close();
24     }
25 }
```

```
& 'C:\Program Files\Java\jdk1.8.0_341\b
Storage\fa5bb4322bcb69e63c79fb4c679be663\redhat.java\jdt_v
Pilih A atau B: B
Anda perlu kurangi main game
PS E:\TIPE DATA>
```

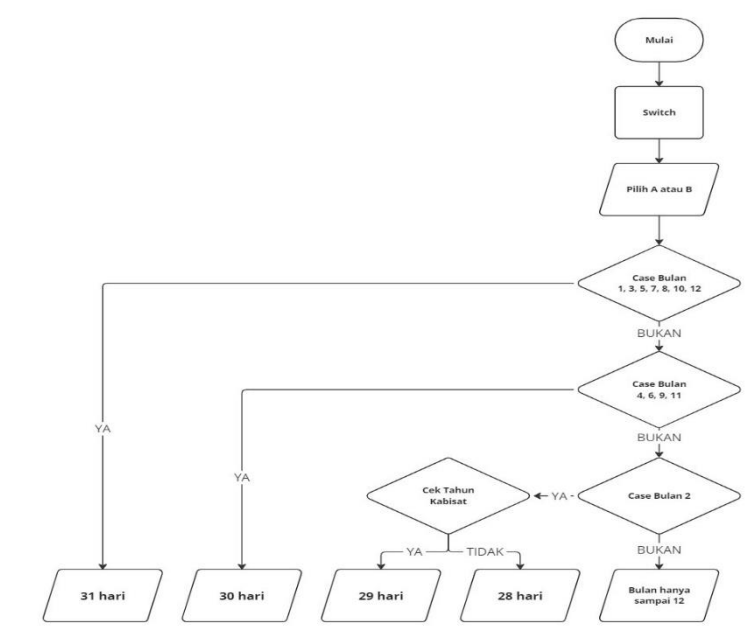
Luaran yang dihasilkan sudah sesuai dengan yang diminta pada soal. Luaran juga sudah diperbaiki sampai bisa menghasilkan luaran tanpa error.

Diagram flowchart untuk if bersarang adalah representasi visual dari alur logika sebuah program yang menggunakan struktur pengambilan keputusan bertingkat atau bersarang. Berikut pengimplementasian Diagram flowchart dari program yang saya buat :

```

graph TD
    Start([Mulai]) --> Input[/Pilih A atau B/]
    Input --> Decision1{Data == 'A'}
    Decision1 -- YA --> Print1[/Print "Anda sudah rajin belajar"/]
    Decision1 -- TIDAK --> Decision2{Data == 'B'}
    Decision2 -- YA --> Print2[/Print "Anda perlu kurangi main game"/]
    Decision2 -- TIDAK --> Print3[/Print "pilihan Anda diluar A atau B"/]
    Print1 --> End([Selesai])
    Print2 --> End
    Print3 --> End
  
```

b. Gambar flowchart Latihan 2.3



[No.1] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab

- Kesimpulan dari program diatas adalah kedua proram mendemonstrasikan penggunaan switch statement dalam java untuk menangani multiple cases, namun dengan tingkat kompleksitas dan tujuan yang berbeda. Contoh 3 menunjukkan penggunaan dasar switch namun memerlukan perbaikan sintaks , sementara contoh 4 menunjukkan aplikasi yang lebih kompleks dengan penanganan kasus yang lebih banyak dan logika tambahan. Keduanya bisa dioptimalkan lebih lanjut untuk meningkatkan efisisensi dan keterbacaan kode.
- Kesimpulan yang diberikan didasarkan pada analisis terhadap kedua program yang disajikan. Proses analisis dimulai dengan memeriksa struktur kode dari masing – masing program, yang memungkinkan identifikasi tujuan dan mekanisme kerja keduanya . Dalam contoh 3, penggunaan input karakter 'A' atau 'B' menunjukkan tujuan program untuk memberikan

respon berdasarkan pilihan sederhana. Sementara itu, contoh 4 menggunakan input numerik untuk bulan, mengindikasikan tujuannya dalam menghitung jumlah hari dalam bulan tertentu. Implementasi switch statement pada kedua program menjadi bagian utama dalam analisis. Contoh 3 menerapkan switch sederhana dengan dua pilihan utama, sementara contoh 4 menunjukkan penggunaan yang lebih kompleks dengan 12 kasus berbeda, yang mencerminkan 12 bulan dalam 1 Tahun. Perbedaan ini menjadi dasar untuk menyimpulkan tingkat kompleksitas masing-masing program.

Refleksi

Saya menyelesaikan tugas ini dengan hanya satu malam, karena tugas ini termasuk tugas yang dikategorikan tugas dengan tingkat kesulitan menengah. Saya memberikan usaha yang terbaik saat mengerjakan tugas ini, banyak hal-hal baru yang saya pelajari saat mengerjakan tugas ini. Saya bisa meningkatkan kemampuan saya saat mengerjakan tugas ini dengan cara mengganti variabel pada contoh data atau ciri-ciri yang diinstruksikan. Pada materi yang mendatang saya harap saya bisa mendapatkan materi yang lebih sulit karena jika saya mendapatkan materi yang lebih sulit maka akan lebih banyak ilmu yang saya dapatkan dari tugas tersebut.