

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Vivilia Regita Pramesti G1F024060	If Else	23 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)		

Jawaban:

Nama & NPM	Topik:	Tanggal:
Vivilia Regita Pramesti G1F024060	If Else	23 September 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

```
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!

Jawab:

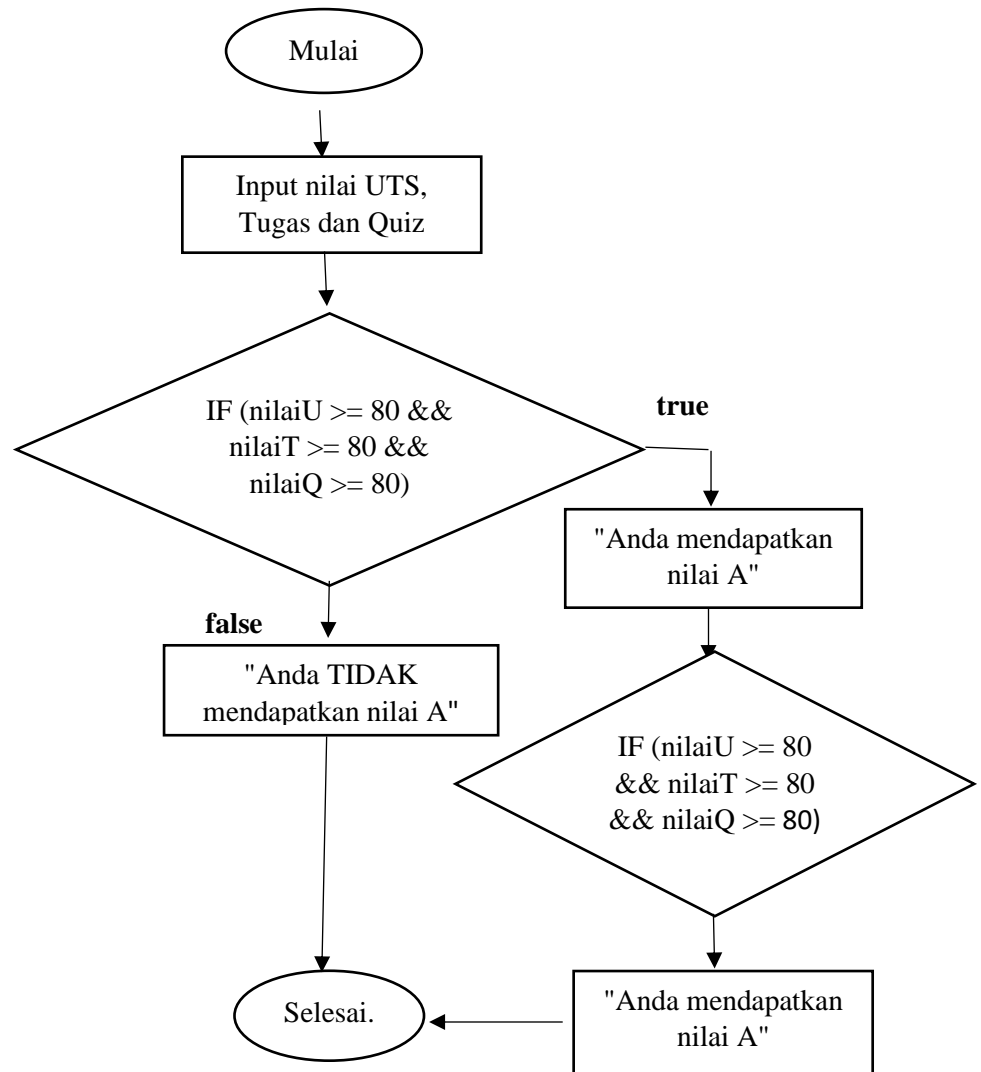
- 1.1 Contoh 1 menunjukkan implementasi queue sederhana dengan percabangan if. Namun, beberapa kesalahan dalam kode ini menyebabkan kegagalan kompilasi. Pertama, ada kesalahan dalam penggunaan variabel masuk, yang seharusnya diganti dengan input sesuai dengan nama objek Scanner yang dideklarasikan. Selain itu, variabel nilai tidak dideklarasikan sebelum digunakan, yang menyebabkan error. Terakhir, pernyataan if harus menggunakan operator perbandingan (==) daripada operator penugasan (=).

Contoh 2 menunjukkan penggunaan percabangan bersarang yang lebih kompleks. Pengguna diminta untuk memasukkan nilai tugas dan quiz dalam kode ini. Namun, variabel nilaiU tidak dideklarasikan, yang menyebabkan error saat program dijalankan. Nilai T dan Q adalah variabel yang digunakan untuk menyimpan input pengguna.

- 1.2 Sudah saya tambahkan pada program
- 1.3 Kondisi yang tepat adalah `IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)`

1.4. Uraikan gambar diagram flowchart dari Latihan 1.2!

Jawab :



[No.1] Analisis dan Argumentasi

Jawab:

Contoh 1 menunjukkan implementasi queue sederhana dengan percabangan if. Namun, beberapa kesalahan dalam kode ini menyebabkan kegagalan kompilasi. Pertama, ada kesalahan dalam penggunaan variabel masuk, yang seharusnya diganti dengan input sesuai dengan nama objek Scanner yang dideklarasikan. Selain itu, variabel nilai tidak dideklarasikan sebelum digunakan, yang menyebabkan error. Terakhir, pernyataan if harus menggunakan operator perbandingan (==) daripada operator penugasan (=).

Contoh 2 menunjukkan penggunaan percabangan bersarang yang lebih kompleks. Pengguna diminta untuk memasukkan nilai tugas dan quiz dalam kode ini. Namun, variabel nilaiU tidak dideklarasikan, yang menyebabkan error saat program dijalankan. Nilai T dan Q adalah variabel yang digunakan untuk menyimpan input pengguna.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma contoh 1

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai

2. Inisialisasi : Buat objek Scanner untuk mengambil input dari pengguna.
3. Tampilkan pesan "Masukkan Angka Anda:".
4. Ambil input dari pengguna dan simpan dalam variabel nilai
5. Cek kondisi
6. Selesai.

Algoritma contoh 2

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai.
2. Tampilkan pesan "Masukkan Angka UTS Anda:" dan ambil input nilai UTS.
3. Tampilkan pesan "Masukkan Angka Tugas Anda:" dan ambil input nilai Tugas.
4. Tampilkan pesan "Masukkan Angka Quiz Anda:" dan ambil input nilai Quiz.
5. Jika nilai Quiz, Tugas, UTS ≥ 80 , tampilkan "Anda mendapatkan nilai A". Jika tidak, tampilkan "Anda TIDAK mendapatkan nilai A".
6. Jika nilai UTS < 80 , tampilkan "Anda TIDAK mendapatkan nilai A".
7. Selesai.

2. Kode program dan luaran

- a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

Contoh 1:

The screenshot shows the Eclipse IDE with a Java project. The editor displays the following code for 'IfPercabangan.java':

```

1 package Scanner;
2
3 import java.util.Scanner; //memanggil impor package yang membaca masukan pengguna
4
5 public class IfPercabangan {
6     public static void main(String[] args) {
7         Scanner masuk = new Scanner(System.in); // membaca teks yang dimasukkan pengguna
8         System.out.print("Masukkan Angka Anda : "); //pengguna memasukkan data
9         int nilai = masuk.nextInt(); //menyimpan masukan pengguna ke tipe data
10
11         if (nilai == 1000) { //percabangan yang memeriksa kondisi
12             System.out.println("Seribu"); //baris kode yang dieksekusi bila benar
13         }
14         else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
15             System.out.println("Nilai Bukan Seribu");
16         }
17     }
18 }
19

```

Below the code editor, the console output is shown:

```

<terminated> IfPercabangan [Java Application] C:\Users\ASUS\p2\pool\plugins\org.eclipse.justj.openjdk.h
Masukkan Angka Anda : 1000
Seribu

```

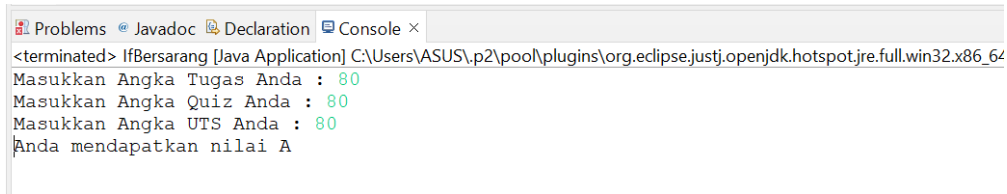
Contoh 2:

The screenshot shows the Eclipse IDE with a Java project. The editor displays the following code for 'IfBersarang.java':

```

1 package Scanner;
2
3 import java.util.Scanner;
4
5 public class IfBersarang {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in); // Menggunakan satu objek Scanner
8         System.out.print("Masukkan Angka Tugas Anda : ");
9         int nilaiT = input.nextInt();
10
11         System.out.print("Masukkan Angka Quiz Anda : ");
12         int nilaiQ = input.nextInt();
13
14         System.out.print("Masukkan Angka UTS Anda : ");
15         int nilaiU = input.nextInt();
16
17         // Memeriksa semua kondisi
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         input.close(); // Menutup scanner
25     }
26 }

```



```
<terminated> IfBersarang [Java Application] C:\Users\ASUS\AppData\Local\Temp\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64
Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 80
Masukkan Angka UTS Anda : 80
Anda mendapatkan nilai A
```

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.1] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

Seperti yang ditunjukkan dalam diskusi tentang implementasi algoritma dalam kode program, penggunaan satu objek Scanner untuk membaca input pengguna lebih efisien daripada menggunakan banyak objek. Nilai akhir siswa berdasarkan nilai tugas, quiz, dan UTS dihitung dengan berhasil oleh algoritma yang digunakan. Dengan kondisi yang tepat, program dapat menghasilkan output yang sesuai dengan standar penilaian yang diinginkan.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Prinsip efisiensi dan kemudahan pemeliharaan kode menentukan keputusan untuk menggunakan satu objek Scanner atau beberapa. Penggunaan beberapa objek Scanner dapat memperumit alur program dan meningkatkan kemungkinan terjadinya kesalahan. Selain itu, kondisi yang menyatukan semua nilai dalam satu pernyataan logika (menggunakan &&) memastikan bahwa semua aspek penilaian dipertimbangkan secara bersamaan. Ini mencerminkan prinsip keadilan dan keakuratan dalam penilaian akademis.

[No. 2] Identifikasi Masalah:

1. Uraikan permasalahan dan variable

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!

Jawab:

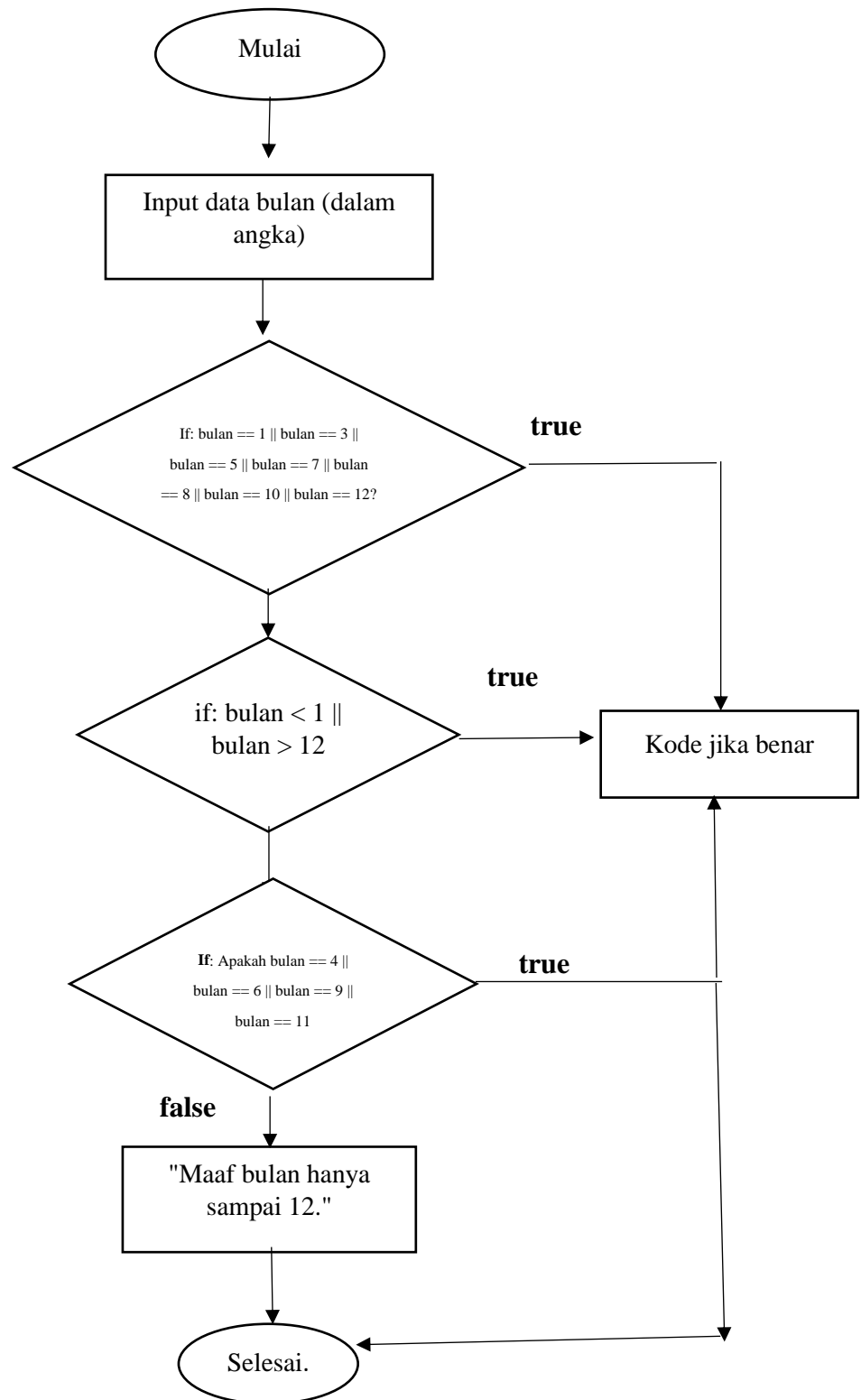
2.1 Variabel data: Nama variabel yang digunakan untuk input seharusnya masukData bukan data. Penggunaan : setelah switch(data): Seharusnya menggunakan { untuk memulai blok. Case tidak dikelilingi oleh tanda kutip: Kasus case A harusnya ditulis sebagai case'A'.

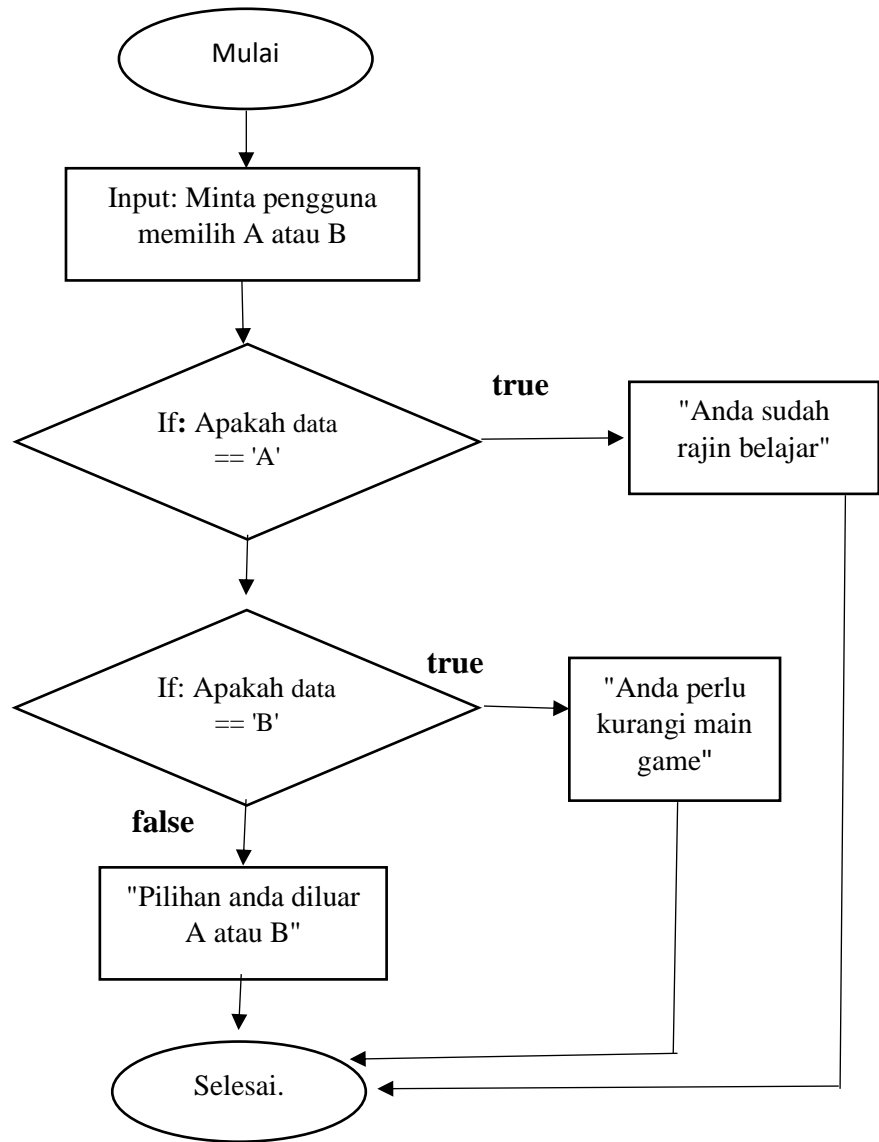
2.2 penulisan jumlahHari = 31; dan jumlahHari = 30; diulang beberapa kali. Saya sudah membuat kode program menjai lebih ringkas dan efisien.

2.3 Switch lebih mudah dibaca saat ada banyak kondisi tetap. Sangat efektif untuk nilai konstan (seperti angka atau karakter). If lebih fleksibel dan dapat menangani kondisi yang lebih kompleks, termasuk rentang atau ekspresi boolean.

2.4 Flowchart latihan 2.2 dan latihan 2.3

1





[No.2] Analisis dan Argumentasi

Jawab:

Pada contoh 3, variabel data dan sintaksis switch(data), yang seharusnya menggunakan tanda kurung kurawal {}, dideklarasikan dengan salah. Penggunaan case tanpa tanda kutip juga merupakan kesalahan. Sangat penting untuk mempertimbangkan pernyataan break karena jika tidak ada, program akan melanjutkan eksekusi ke kasus berikutnya, yang dapat menghasilkan output yang tidak diinginkan. Namun, Contoh 4 menggunakan switch untuk menghitung jumlah hari dalam bulan dengan efektif. Namun, dengan menggunakan array untuk menyimpan jumlah hari, efisiensi dapat ditingkatkan dan duplikasi kode dapat dikurangi.

[No.2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma contoh 3
 1. Mulai
 2. Input minta pengguna untuk memilih A atau B.
 3. Switch Cek nilai data
 4. Selesai
- 2) Algoritma contoh 4

1. Mulai
2. Input Minta pengguna untuk memasukkan nomor bulan (1-12)
3. Inisialisasi set variabel jumlahHari ke 0.
4. Switch cek nilai bulan
5. Tampilkan output jumlah hari
6. Selesai

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

Contoh 3:

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

```
<terminated> SwitchBersarangDua [Java Application] C:\Users\ASUS\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre-17.0.2\bin\java.exe
Pilih A atau B : A
Anda sudah rajin belajar
```

Contoh 4:

```
1 package Scanner;
2
3 import java.util.Scanner;
4
5 public class SwitchEmpatBersarang {
6     public static void main(String[] args) {
7         byte bulan;
8         int tahun = 2022;
9         int jumlahHari;
10
11        System.out.print("Masukkan data bulan (dalam angka): ");
12        Scanner masukData = new Scanner(System.in);
13        bulan = masukData.nextByte();
14
15        // Array untuk jumlah hari dalam setiap bulan
16        int[] hariPerBulan = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
17
18        // Memeriksa bulan
19        if (bulan < 1 || bulan > 12) {
20            System.out.println("Maaf bulan hanya sampai 12.");
21            return; // Keluar jika bulan tidak valid
22        }
23
24        // Mengambil jumlah hari dari array
25        jumlahHari = hariPerBulan[bulan];
26
27        // Memeriksa tahun kabisat
28        if (bulan == 2 && (tahun % 4 == 0 && (tahun % 100 != 0 || tahun % 400 == 0))) {
29            jumlahHari = 29; // Tahun kabisat
30        }
31
32        System.out.println("Jumlah hari = " + jumlahHari);
33        masukData.close(); // Menutup Scanner
34    }
35 }
```

```
<terminated> SwitchEmpatBersarang [Java Application] C:\Users\ASUS\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full\jre-17.0.2\bin\java.exe
Masukkan data bulan (dalam angka): 11
Jumlah hari = 30
```

- b) Analisa luaran yang dihasilkan

Contoh:

Luaran sudah sesuai dengan program yang disusun.

[No.2] Kesimpulan

Analisa

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

Jawab:

Dari analisis yang dilakukan, permasalahan dalam kode program terkait penggunaan struktur kontrol switch dan penanganan input pengguna dapat diselesaikan dengan baik. Algoritma yang disusun untuk kedua contoh menunjukkan langkah-langkah yang jelas dan logis, mulai dari pengambilan input hingga pengolahan hasil. Kode program yang telah diperbaiki memenuhi semua kriteria yang diinginkan, dengan penggunaan switch untuk memeriksa pilihan pengguna dan memberikan output yang sesuai. Dengan penanganan yang tepat, kesalahan sintaksis sebelumnya dapat diatasi, sehingga program berfungsi sesuai harapan.

- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Efisiensi dan keterbacaan kode adalah dasar dari keputusan ini. Struktur switch dipilih karena tidak perlu menggunakan if-else berulang untuk menangani banyak kondisi. Selain itu, pengembangan metode input dan output yang lebih baik memastikan interaksi pengguna yang lancar dan hasil yang jelas. Oleh karena itu, keputusan ini dibuat untuk meningkatkan kualitas program dan mengurangi kesalahan saat eksekusi.

Refleksi:

Minggu ini, saya lebih memahami struktur percabangan pemrograman seperti if-else, if bercabang, if bersarang, dan switch bersarang. Pengalaman ini meningkatkan kemampuan saya untuk mengelola alur eksekusi program dalam berbagai kondisi dan memberikan pemahaman yang lebih baik tentang kapan masing-masing struktur harus digunakan. Saya merasa lebih percaya diri dan siap untuk menerapkan logika yang telah saya pelajari dalam proyek-proyek mendatang berkat pengalaman ini, meskipun saya menghadapi kesulitan dalam menjaga kode terbaca dan memilih struktur yang tepat. Saya sadar bahwa menguasai konsep ini sangat penting untuk membuat program yang berhasil.