

Nama & NPM	Topik:	Tanggal:
Dewangga Nugroho Anwar G1F024045	IF	19 September 2024

[No. 1] Identifikasi Masalah Contoh 1:

1. Analisa kode program yang tepat untuk menghitung masukan pengguna termasuk bilangan genap atau bilangan ganjil.

Jawab:

Analisa kode program yang tepat adalah dengan memodifikasi atau menambahkan operator pembandingan `==` pada struktur percabangan `if` nilai 10 untuk membuat Keputusan. Saya menambahkan operator modulus (`nilai % 2 == 0`) untuk mengetahui bilangan yang di masukan oleh pengguna adalah bilangan ganjil atau genap, jika nilai yang di masukan jika bilangan tersebut genap maka output akan menjadi 'Bilangan Genap', jika nilai yang di masukan adalah bilangan ganjil maka output akan menjadi 'Bilangan Ganjil'.

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan operator modulus pada line baru dengan kode sebagai berikut:

```
if (nilai % 2 == 0) {
    System.out.println("Bilangan Genap");
} else {
    System.out.println("Bilangan Ganjil");
}
```

- 2) Alasan solusi ini karena pada soal meminta kita untuk membuat program yang dapat mengecek apakah bilangan yang di input oleh user adalah bilangan ganjil atau genap karena modulus (%) memberikan sisa hasil bagi dari suatu pembagian.
- 3) Perbaiki kode program dengan cara. Kode program tidak memiliki letak kesalahan namun membutuhkan modifikasi tambahan karena menurut soal

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Deklarasi class utama

- (c) Pemeriksaan nilai
 - (d) Pemeriksaan bilangan ganjil atau genap
 - (e) Selesai
- 2) Kode program dan luaran

<pre> 1- import java.util.Scanner; // memanggil impor package yang membaca masukan pengguna 2 3- public class PercabanganIf { 4- public static void main(String[] args) { 5 Scanner masuk = new Scanner(System.in); // membaca teks yang dimasukkan pengguna 6 System.out.print("Masukkan Angka Anda : "); // pengguna memasukkan data 7 int nilai = masuk.nextByte(); // menyimpan masukan pengguna ke tipe data byte 8 9 // Memeriksa apakah angka yang dimasukkan adalah sepuluh 10 if (nilai == 10) { // percabangan yang memeriksa kondisi 11 System.out.println("Sepuluh"); // baris kode yang dieksekusi bila benar 12 } else { 13 System.out.println("Nilai Bukan Sepuluh"); 14 } 15 16 // Memeriksa apakah bilangan genap atau ganjil 17 if (nilai % 2 == 0) { // pengecekan apakah bilangan habis dibagi 2 18 System.out.println("Bilangan Genap"); 19 } else { 20 System.out.println("Bilangan Ganjil"); 21 } 22 } 23 } </pre>	<pre> java -cp /tmp/GJjCOhJpmI/PercabanganIf Masukkan Angka Anda : 10 Sepuluh Bilangan Genap === Code Execution Successful === </pre>
--	--

Gambar Contoh (1) 1.1

- a) Analisa luaran yang dihasilkan

Jawab.

Luaran sudah sesuai dengan yang di minta dan perintah saol, pada gambar 1.1 menampilkan input dan output yang mana output nya adalah bilangan genap karena nilai yang di masukan oleh saya adalah bilangan genap 10.

[No.1] Kesimpulan

1) Kreasi

- a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?

Jawab.

Pengetahuan baru yang saya dapat setelah mempelajari program contoh 1 bahwa, jika kita ingin menghitung ganjil atau genap dari nilai yang kita inputkan secara manual kita dapat menambahkan percabangan if baru dengan menggunakan operator modulus karena operator modulus dapat mengecek apakah bilangan yang di input oleh user adalah bilangan ganjil atau genap.

[No. 1] Identifikasi Masalah Contoh 2

- 1) Apakah ketiga kondisi IF pada Contoh 1.2. dapat diringkas menjadi satu kondisi?

Jawab.

Ketiga kondisi if pada program tersebut tentu dapat di ringkas menjadi 1 baris dengan menggunakan operator && dengan menggunakan operator && kita dapat memeriksa sekaligus apakah semua kondisi sudah terpenuhi dan lebih efisien.

[No.1] Analisis dan Argumentasi

- 1) Cermati contoh 2, analisa kondisi pada IF bersarang!

Jawab.

Hasil dari Analisa saya terhadap kondisi if bersarang pada program contoh 2 adalah. Terdapat 3 kondisi if bersarang pada program yang pertama memiliki tugas untuk memeriksa nilai tugas T yang di masukan pengguna lebih dari atau sama dengan 80, jika benar program akan melanjutkan ke kondisi ke dua. Pada kondisi ke dua setelah kondisi pertama selesai memeriksa nilai T maka program akan memeriksa nilai quiz lebih atau sama dengan 80. Terakhir jika kondisi pertama dan kedua benar, maka program akan mengecek apakah nilai U lebih dari atau sama dengan 80.

- 2) Apakah ketiga kondisi IF pada Contoh 1.2. dapat diringkas menjadi satu kondisi?

Jawab.

Meringkas ketiga kondisi IF tentu dapat di lakukan dengan menggunakan operator &&.

Dengan menggunakan operator && kita dapat memeriksa sekaligus apakah semua kondisi sudah terpenuhi dan lebih efisien.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma.
 - a) Mulai
 - b) Deklarasikan variable
 - c) Deklarasi input dari pengguna
 - d) Deklarasi if
 - e) Selesai

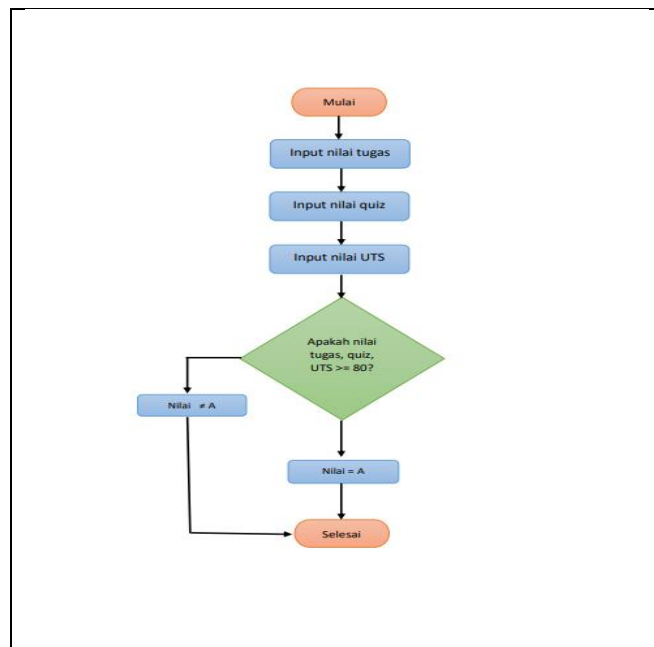
2) Kode program dan luaran

```
1- import java.util.Scanner;
2-
3- public class IFBersarang {
4-     public static void main(String[] args) {
5-         Scanner varI = new Scanner(System.in);
6-         System.out.print("Masukkan Angka Tugas Anda : ");
7-         int nilaiI = varI.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 Quiz Anda : ");
15-        int nilaiU = varU.nextByte();
16-
17-        if (nilaiU >= 80 && nilaiI >= 80 && nilaiQ >= 80) {
18-            System.out.println("Anda mendapatkan nilai A");
19-        }
20-        else{
21-            System.out.println("Anda TIDAK mendapatkan nilai A");
22-        }
23-    }
24- }
```

```
java -cp /tmp/kyhe511o40:/IFBersarang
Masukkan Angka Tugas Anda : 85
Masukkan Angka Quiz Anda : 85
Masukkan Angka Quiz Anda : 85
Anda mendapatkan nilai A
=== Code Execution Successful ===
```

Gambar Contoh (2) 2.1

3) Uraikan gambar diagram flowchart dari Latihan 1.2!



Gambar Flowchart 2.2

Uraian Gambar Flowchart dari program Contoh 2 di atas representasi visual dari pemrograman, dari bentuk oval (Mulai) pada bagian paling atas adalah titik awal mulai nya program. Di bawah nya ada kotak biru pertama dengan isi (Input nilai tugas) merupakan program yang meminta pengguna untuk memasukkan nilai tugas. Pada kotak biru kedua (Input nilai quiz), program meminta pengguna untuk menginput jumlah nilai quiz. Kotak biru ke tiga (Input nilai UTS), program meminta pengguna untuk memasukan nilai quiz. Bentuk ketupat warna hijau adalah tahap pengambilan Keputusan untuk memeriksa apakah nilai lebih dari atau kurang dari 80. Jika hasil nya lebih besar dari 80 maka akan langsung di eksekusi dan selesai dengan output "Anda mendapatkan nilai A". Namun jika pada salah satu dari nilai Tugas, Quiz, dan UTS nilai nya kurang maka akan menghasilkan output "Anda tidak mendapatkan nilai A".

[No. 2] Identifikasi Masalah Contoh 3:

1) Cermati kode pada Contoh 3.

- a) Analisis perbedaan hasil luaran ketika kode menggunakan break, Ketika kode break baris 1 dihapus, dan ketika kode break baris 2 dihapus!.

Jawab.

Hasil luaran Ketika menggunakan kedua mode break saat pengguna melakukan input case A atau case B, hasilnya adalah teratur dan sesuai dengan yang diinginkan. Pada saat pengguna memasukkan case A dan break 1 di hapus output yang akan keluar langsung dari ke dua case, karena fungsi break adalah sebagai penghenti eksekusi lebih lanjut dari switch. Jika kedua break di hapus pada saat pengguna menginput nilai case A akan keluar ouput dari ke tiga case pada program karena tidak adanya break yang mana fungsi break adalah sebagai pembatas atau penghenti eksekusi lebih lanjut. Namun jika pengguna menginput nila case B output case A tidak akan ikut tereksekusi karena case A berada di atas case B dan Default, atau lebih tepat nya di sebut fall-through.

- b) Analisa kegunaan baris kode dengan break dan kata kunci default!.

Jawab.

Fungsi baris kode break pada program tersebut adalah sebagai pengehenti eksekusi lebih lanjut dari switch. Sedangkan untuk fungsi Default adalah sebagai penangkap jika nilai yang di masukan oleh pengguna tidak ada yang sama dalam case.

2) Kode program dan luaran.

<pre>1 import java.util.Scanner; 2 3 public class SwitchBersarang { 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 switch(data) { 10 case 'A': 11 System.out.print("Anda sudah rajin belajar"); 12 13 case 'B': 14 System.out.print(" Anda perlu kurangi main game"); 15 16 default: 17 System.out.print(" Pilihan anda diluar A atau B "); 18 } 19 } 20 }</pre>	<pre>java -cp /tmp/11vi5N3cJA/SwitchBersarang Pilih A atau B : A Anda sudah rajin belajar Anda perlu kurangi main game Pilihan anda diluar A atau B === Code Execution Successful ===</pre>
--	---

Gambar Contoh (3) 3.1

[No. 2] Identifikasi Masalah Contoh 4:

- 1) Apakah masalah ini bisa diubah menjadi perintah IF? Jika bisa, rincikan analisa Anda!

Jawab.

Masalah (*switch*) tersebut dapat kita ubah ke bentuk if karena kedua nyua sama-sama memiliki fungsi untuk memeriksa nilai untuk suatu variable dan menjalankan kode tertentu berdasar kondisi.

- 2) Bandingkan masalah yang dapat diselesaikan percabangan dengan IF atau SWITCH!.

Jawab.

Perbandingan penyelesaian masalah antara percabangan IF atau SWITCH tergantung pada kondisi dan kebutuhan. Pada percabangan IF keunggulan nya terletak pada fleksibilitas yang bisa menyelesaikan sebuah kondisi sulit atau kompleks seperti rentang nilai dan penggunaan ekspresi Boolean dengan kekurangan efisien jika dalam suatu kondisi ada data yang harus di uji dalam jumlah yang banyak. Sedangkan untuk SWITCH keunggulan nya yaitu lebih terstruktur format yang rapi dan mudah di baca, kekurangan SWITCH antara lain yaitu hanya bisa memeriksa nilai konstan seperti Interger, Char, dan String.

[No.2] Penyusunan Algoritma dan Kode Program.

1) Algoritma

- a) Mulai
- b) Deklarasi variable
- c) Deklarasi input bulan
- d) Deklarasi output
- e) Selesai

2) Kode Program.



```
1 import java.util.Scanner;
2
3 public class SwitchBersarang {
4
5     public static void main(String[] args) {
6         byte bulan;
7         int tahun = 2022;
8         int jumlahHari = 0;
9         System.out.print("Masukkan data bulan (dalam angka): ");
10        Scanner masukData = new Scanner(System.in);
11        bulan = masukData.nextByte();
12
13        switch (bulan) {
14            case 1: case 3: case 5: case 7: case 8: case 10: case 12: // baris 1
15                jumlahHari = 31;
16                break;
17            case 4: case 6: case 9: case 11: // baris 2
18                jumlahHari = 30;
19                break;
20            case 2:
21                // Cek apakah tahun kabisat
22                if (tahun % 4 == 0 && (tahun % 100 != 0 || tahun % 400 == 0)) {
23                    jumlahHari = 29; // Tahun kabisat
24                } else {
25                    jumlahHari = 28; // Tahun biasa
26                }
27                break;
28            default:
29                System.out.println("Maaf, bulan hanya sampai 12.");
30                return; // Mengakhiri program jika input tidak valid
31        }
32        System.out.println("Jumlah hari = " + jumlahHari);
33    }
34 }
35 }
```

```
java -cp ./tmp/7cx271m00w/SwitchBersarang
Masukkan data bulan (dalam angka): 7
Jumlah hari = 31
=== Code Execution Successful ===
```

Gambar Contoh (4) 4.1

```

1- import java.util.Scanner;
2
3- public class IfElseBersarang {
4
5-     public static void main(String[] args) {
6-         byte bulan;
7-         int tahun = 2022;
8-         int jumlahHari = 0;
9-         System.out.print("Masukkan data bulan (dalam angka): ");
10-        Scanner masukData = new Scanner(System.in);
11-        bulan = masukData.nextByte();
12
13-        // Menggunakan perintah if-else untuk memeriksa bulan
14-        if (bulan == 1 || bulan == 3 || bulan == 5 || bulan == 7 || bulan == 8 || bulan == 10
15-            || bulan == 12) {
16-            jumlahHari = 31;
17-        } else if (bulan == 4 || bulan == 6 || bulan == 9 || bulan == 11) {
18-            jumlahHari = 30;
19-        } else if (bulan == 2) {
20-            // Cek apakah tahun kabisat
21-            if (tahun % 4 == 0 && (tahun % 100 != 0 || tahun % 400 == 0)) {
22-                jumlahHari = 29; // Tahun kabisat
23-            } else {
24-                jumlahHari = 28; // Tahun biasa
25-            }
26-        } else {
27-            System.out.println("Maaf, bulan hanya sampai 12.");
28-            return; // Mengakhiri program jika input tidak valid
29-        }
30
31-        System.out.println("Jumlah hari = " + jumlahHari);
32-    }
33- }

```

```

java -cp /tmp/9H8Q8N0If0/IfElseBersarang
Masukkan data bulan (dalam angka): 5
Jumlah hari = 31

=== Code Execution Successful ===

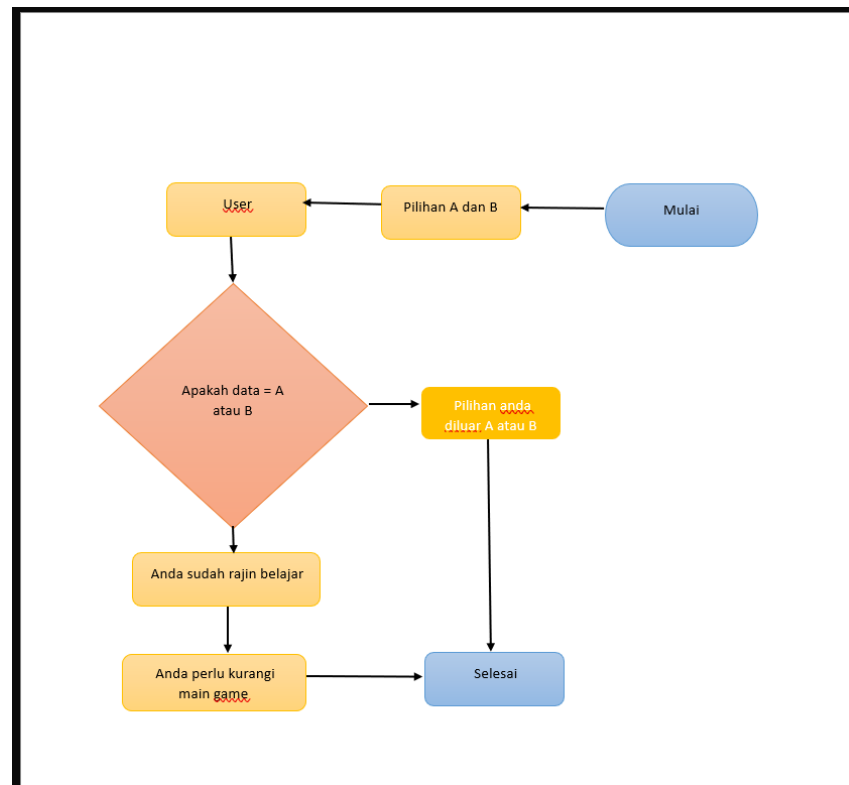
```

Gambar Contoh (4) 4.2

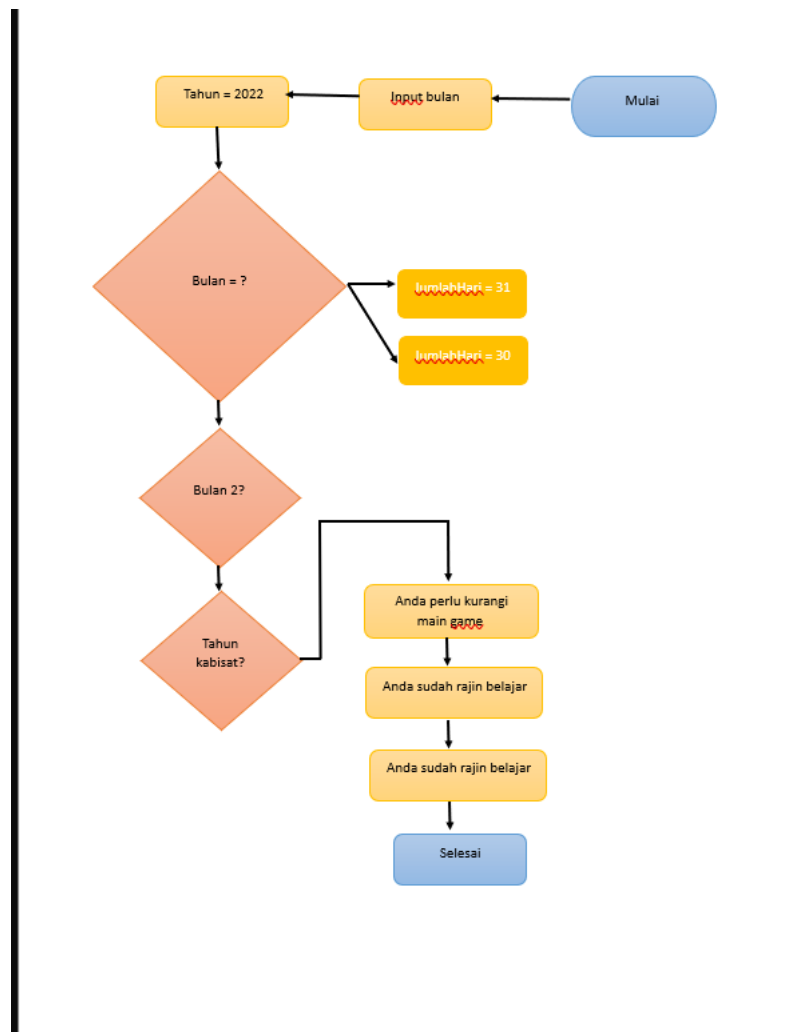
Pada foto 4.1 di atas merupakan kode program perulangan kalender yang menggunakan SWITCH. Sedangkan untuk foto 4.2 adalah input dan output kode program perulangan yang menggunakan perulangan IF.

[No.2] Buatlah dokumentasi gambar flowchart dari Latihan 2.1. dan Latihan 2.2!

1) Flowchart



Gambar 4.3



Gambar 4.4

[No.2] Kesimpulan

1) Evaluasi

Kesimpulan yang dapat saya ambil dari Latihan no 2 contoh 3 dan 4 tersebut bahwa pada contoh 3 jika kita ingin membuat sebuah program dengan fungsi yang sama seperti itu, penggunaan break sangat di sarankan karena jika tidak kode program akan langsung mengeksekusi semua case di saat yang bersamaan saat user melakukan input nilai A atau B. Sedangkan untuk contoh 4 kita dapat mengubah perulangan SWITCH menggunakan IF namun tentu di setiap statement memiliki kelebihan maupun kekurangan sendiri seperti pada statement SWITCH memiliki kelebihan dari segi kerapian dan efisiensi di banding dengan statement IF. Pada statement IF memiliki kelebihan pada fleksibilitas nya.

