

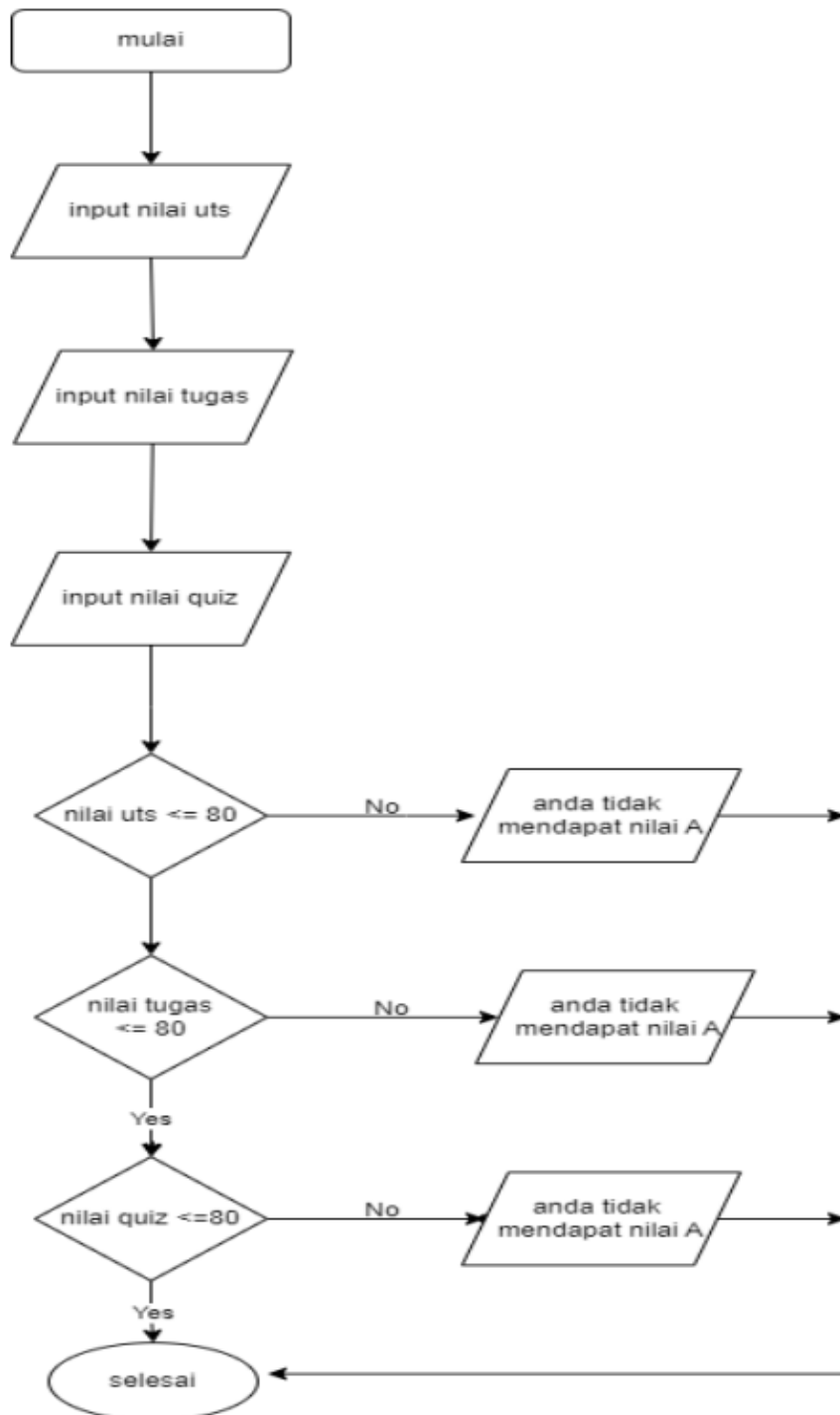
Nama & NPM	Topik:	Tanggal:
Putri G1A000001	Tipe Data	26 Agustus 2022

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel Contoh:

Tuliskan kembali soal:

- 1.1. Bilangan genap merupakan bilangan yang habis dibagi 2. Bilangan ganjil adalah bilangan yang tidak habis dibagi 2. Analisa kode program yang tepat untuk menghitung masukan pengguna termasuk bilangan genap atau bilangan ganjil (lihat Contoh 1)? (Petunjuk: $\text{hitung} = \text{nilai} \% 2$ jika $\text{hitung} = 0$ maka bilangan genap, jika $\text{hitung} = 1$ maka bilangan ganjil)
- 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. Permasalahan: apakah bilangan yang dimasukkan oleh pengguna adaah genap atau ganjil. Variabel: `hitung` : variabel untuk menyimpan hasil dari `nilai % 2` `bilangan` : variabel untuk menyimpan hasil input dari pengguna.
- 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 ($\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80$) b. IF ($\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \ \&\& \text{nilaiQ} \geq 80$) c. IF ($\text{nilaiU} \geq 80 \ \&\& \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80$) d. IF ($\text{nilaiU} \geq 80 \ \&\& \text{nilaiT} \geq 80 \ \&\& \text{nilaiQ} \geq 80$) jawaban: a. IF ($\text{nilaiU} \geq 80 \mid \mid \text{nilaiT} \geq 80 \mid \mid \text{nilaiQ} \geq 80$) Hal ini didasari karena hanya memerlukan satu kondisi yang benar.
- 1.4. Uraikan gambar diagram flowchart dari Latihan 1.2



[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan operator logika yang benar untuk memeriksa kondisi.
- 2) Alasan solusi ini karena untuk memudahkan pembacaan kode.
- 3) Perbaiki kode program dengan cara menggunakan satu kondisi IF yang menggabungkan semua kondisi menggunakan operator logika OR.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

- (a) Pengguna memasukkan bilangan.
- (b) Hitung apakah bilangan genap atau ganjil.
- (c) Pengguna memasukkan nilai UTS, Tugas, dan kuis.

(d) Check apakah nilai tersebut lebih dari sama dengan 80.

(e) Tampilkan hasil

Kode program dan luaran

```
12 }
13
14 // Fungsi untuk mengecek apakah siswa mendapat nilai A
15 public static String cekNilaiA(int nilaiU, int nilaiT, int nilaiQ) {
16     if (nilaiU >= 80 || nilaiT >= 80 || nilaiQ >= 80) {
17         return "Siswa mendapat nilai A";
18     } else {
19         return "Siswa tidak mendapat nilai A";
20     }
21 }
22
23 public static void main(String[] args) {
24     Scanner scanner = new Scanner(System.in);
25
26     // Input untuk bilangan
27     System.out.print("Masukkan bilangan: ");
28     int bilangan = scanner.nextInt();
29     String hasilBilangan = cekBilangan(bilangan);
30     System.out.println(hasilBilangan);
31
32     // Input untuk nilai UTS, Tugas, dan Quiz
33     System.out.print("Masukkan nilai UTS: ");
34     int nilaiU = scanner.nextInt();
35
36     System.out.print("Masukkan nilai Tugas: ");
37     int nilaiT = scanner.nextInt();
```

a) Hasil luaran

```
Masukkan bilangan: 9
Bilangan Ganjil
Masukkan nilai UTS: 78
Masukkan nilai Tugas: 84
Masukkan nilai Quiz: 100
Siswa mendapat nilai A
```

b) Analisa luaran yang dihasilkan

kode program akan meminta pengguna untuk memasukkan sebuah bilangan dan menampilkan apakah bilangan tersebut genap atau ganjil. Lalu, program meminta pengguna untuk memasukkan nilai UTS, Tugas, dan Quiz sehingga dapat menampilkan apakah siswa mendapat nilai A atau tidak berdasarkan kriteria yang telah ditentukan

[No.1] Kesimpulan

Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Pembuatan program yang menggunakan algoritma efektif untuk memeriksa berdasarkan nilai UTS, Tugas, dan Quiz apakah siswa mendapatkan nilai A.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Dasar pengambilan keputusan untuk kasus ini dengan membuat program berlogika sederhana sehingga kode lebih mudah dipahami. Pada program itu saya menggunakan bentuk kelas public karena memang sedang tidak ada kebutuhan untuk membuat suatu kelas privat. Perbaikan program dengan menambahkan fungsi karena suatu program memerlukan fungsi yang terstruktur. Penggunaan tipe data yang sesuai untuk nilai input serta hasil output sehingga program lebih mudah dipahami

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel Contoh:

Tuliskan kembali soal:

2.1. Cermati kode pada Contoh 3.

Hapuslah kode break; pada //baris 1, eksekusi kembali. Kemudian hapuslah kode break; pada //baris 2, eksekusi kembali.

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

(1) Kode penggunaan `break`. Jika pengguna memasukkan, outputnya : `A` dengan teks (anda sudah rajin belajar) Jika memasukkan, output: `B` dengan teks (anda perlu kurangi main game) dan jika pengguna memasukkan pilihan lain, output nya: (pilihan anda di luar A atau b).

(2) Ketika `break` pada baris 1 dihapus Jika pengguna memasukkan, outputnya akan menjadi: `A` dengan teks (anda sudah rajin belajar, anda perlu kurangi main game) Karena di baris 1 dihapus, program akan melanjutkan ke kasus berikutnya dan mencetak pesan kedua.

(3) Ketika `break` pada baris 2 dihapus Jika pengguna memasukkan, outputnya akan menjadi: `B` dengan teks (anda sudah rajin belajar, anda perlu kurangi main game) Karena di baris 2 dihapus, program akan melanjutkan ke kasus berikutnya dan mencetak pesan default.

b) Analisa kegunaan baris kode dengan break dan kata kunci default! Kegunaan break untuk menghentikan eksekusi dari switch statement setelah kasus yang cocok ditemukan.

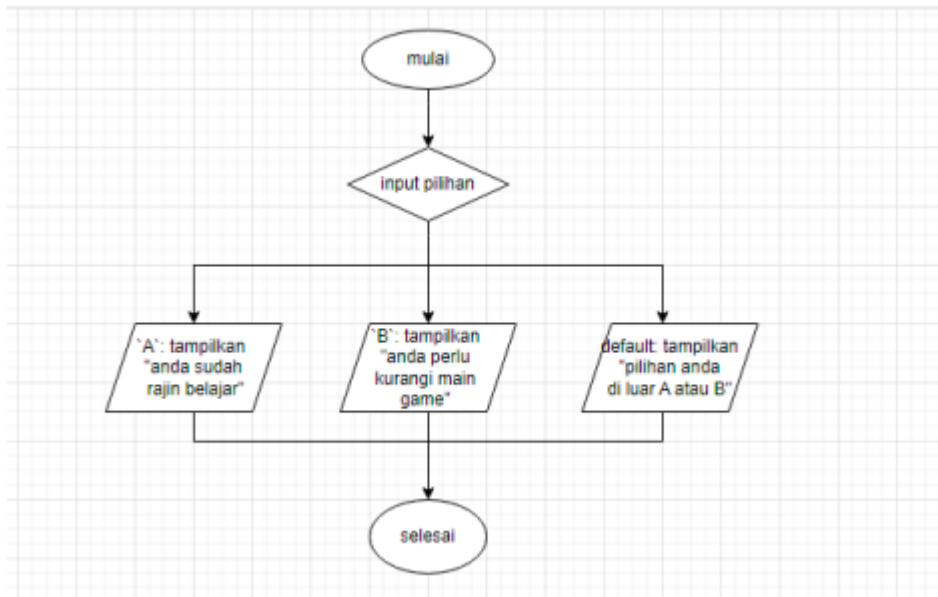
Sedangkan untuk kata kunci default menyediakan jalur alternatif jika tidak ada case yang cocok. Ini berguna untuk menangani input yang tidak valid dan memberitahu pengguna bahwa input tidak sesuai.

2.2. Cermati kode pada Contoh 4 yang menampilkan jumlah hari sesuai dengan bulannya. Namun kode tersebut baru sampai bulan ke-4. Tambahkan sampai bulan ke-12 pada baris ke-1 dan baris ke-2.

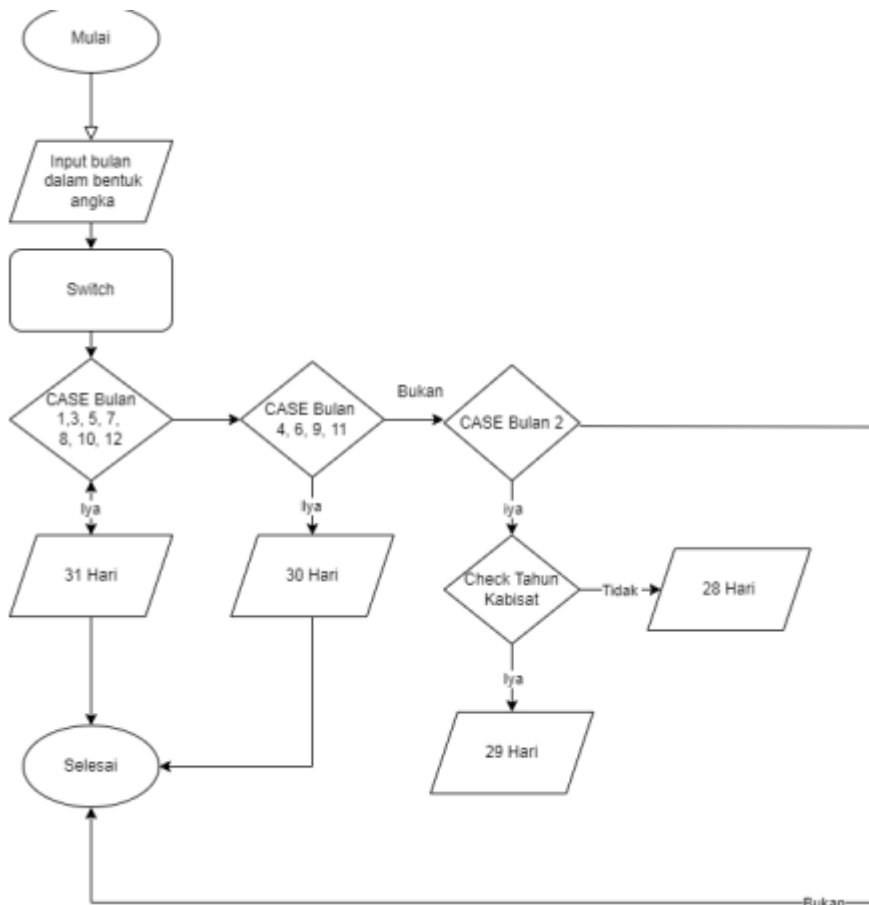
2.3. Cermati permasalahan yang dipecahkan pada Contoh 3. Apakah masalah ini bisa diubah menjadi perintah IF? Iya, bisa. Jika bisa, rincikan analisa Anda! Bandingkan masalah yang dapat diselesaikan percabangan dengan IF atau SWITCH ! Dengan menggunakan if-else untuk menangani beberapa kondisi yang berbeda. Penggunaan if-else lebih fleksibel karena menangani lebih banyak kondisi dengan mudah. Sedangkan dengan switch itu lebih efisien untuk menangani beberapa kondisi berbeda dengan nilai yang terbatas. Dan menggunakan break untuk menghentikan eksekusi setelah case yang cocok ditemukan.

2.4. Buatlah dokumentasi gambar flowchart dari Latihan 2.1. dan Latihan 2.2! Permasalahan : visualisasi alur logika dari program. variable: flowchart

Latihan 2.1



Latihan 2.2



[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan kondisi dalam struktur kontrol yang ada.
- 2) Alasan solusi ini karena menambahkan kondisi akan memperluas kemampuan program dalam menangani lebih banyak kasus.
- 3) Perbaiki kode program dengan cara menghapus atau juga menyesuaikan penggunaan pernyataan `break`, serta menambahkan kondisi dalam penghitungan jumlah hari setiap bulan sampai bulan ke-12

[No.2] Penyusunan Algoritma dan Kode Program


1) Algoritma

- Mohon input nilai dari pengguna
- Tentukan bulan berdasarkan input tersebut
- Gunakan struktur kontrol (IF/SWITCH) untuk menentukan jumlah hari.
- Tampilkan hasil output ke pengguna.

3) Kode program dan luaran

```
1 import java.util.Scanner;
2
3 public class JumlahHari {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Minta input dari pengguna untuk memilih bulan
8         System.out.print("Masukkan bulan (1-12): ");
9         int month = scanner.nextInt();
10
11         // Gunakan switch untuk menentukan jumlah hari berdasarkan bulan
12         switch (month) {
13             case 1:
14             case 3:
15             case 5:
16             case 7:
17             case 8:
18             case 10:
19             case 12:
20                 System.out.println("Jumlah hari: 31");
21                 break;
22             case 4:
23             case 6:
24             case 9:
25             case 11:
26                 System.out.println("Jumlah hari: 30");
27                 break;
28             case 2:
29                 System.out.println("Jumlah hari: 28/29");
30                 break;
31             default:
32                 System.out.println("Bulan tidak valid.");
33                 break;
34         }
35         scanner.close();
36     }
37 }
38 }
```

a) Kode program dan luaran



```
Masukkan bulan (1-12): 6
Jumlah hari: 30
```

- kode dengan pernyataan break: Program akan meminta pengguna memasukkan input kemudian akan menampilkan hasil berdasarkan kondisi yang diberikan. Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.2] Kesimpulan

1) Analisa

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

Pengidentifikasi pada program ini berhasil, program berhasil menemukan perbedaan hasil luaran, penentuan jumlah hari sesuai dengan bulan, dan memecahkan permasalahan dengan percabangan yang tepat.

- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Penggunaan percabangan IF ELSE untuk memudahkan pembacaan kode dan agar tidak terlalu kompleks. Serta penggunaan tipe data yang sesuai setiap variabel sehingga memudahkan dalam pengolahan data

Refleksi

Untuk pengalaman pelajaran pada minggu ini begitu berharga karena saya menjadi dapat memahami lebih lanjut tentang penggunaan struktur kontrol dalam pemrograman. Saya belajar bahwa flowchart begitu penting untuk memahami alur program. Tantangan terbesar adalah cara aku memahami perubahan kecil dalam kode dapat memengaruhi hasil akhir, serta memberikan wawasan lebih terhadap debugging dan pengembangan software secara umum.