

Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Renni Ren Rofa.Pgb G1F021002	Percabangan: IF dan SWITCH	10 Oktober 2024

[No.1] Identifikasi Masalah:

- 1) Uraikan permasalahan :
 - 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: hitung = nilai % 2 jika hitung = 0 maka bilangan genap, jika 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.
 - 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!

Variabel yang relevan dalam masalah ini adalah:

- nilaiU: Nilai UTS
- nilaiT: Nilai Tugas
- nilaiQ: Nilai Quiz

- 2) Rincikan sumber informasi yang relevan :
Webpage: Dokumentasi resmi Java di <https://docs.oracle.com/javase/tutorial/> digunakan untuk memahami lebih dalam tentang implementasi struktur kontrol dalam bahasa Java.
- 3) Uraikan rancangan solusi yang diusulkan :
Solusi yang diusulkan adalah menggunakan struktur keputusan bersarang (nested if) untuk memeriksa apakah semua nilai (UTS, Tugas, dan Quiz) lebih besar atau sama dengan 80. Jika semua memenuhi syarat, maka siswa mendapat nilai A.
- 4) Analisis susunan solusi, parameter solusi :
Parameter solusi: Solusi ini bergantung pada kondisi yang telah ditetapkan, yaitu nilai harus >= 80. Jika salah satu nilai tidak memenuhi kriteria, maka hasilnya akan berupa pesan yang menandakan nilai tidak cukup untuk mendapat A.

[No.1] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan:
Solusi yang diusulkan adalah mengecek nilai UTS, Tugas, dan Quiz dengan menggunakan percabangan bersarang. Jika semua nilai tersebut lebih dari atau sama dengan 80, maka output berupa pesan "Anda mendapatkan nilai A". Jika ada nilai yang kurang dari 80, program akan memberikan pesan spesifik terkait nilai yang kurang.
- 2) Analisis solusi, kaitkan dengan permasalahan.

Solusi ini efektif untuk menyelesaikan masalah pengecekan nilai menggunakan percabangan bersarang karena dapat memeriksa beberapa kondisi sekaligus dan memberikan hasil spesifik tergantung pada kondisi yang dipenuhi. Program ini memastikan bahwa jika semua nilai memenuhi kriteria, siswa akan mendapatkan hasil yang diinginkan, yaitu nilai A.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma :
 1. Ambil input nilai UTS, Tugas, dan Quiz dari pengguna.
 2. Periksa apakah nilai UTS ≥ 80 .
 3. Jika ya, periksa apakah nilai Tugas ≥ 80 .
 4. Jika ya, periksa apakah nilai Quiz ≥ 80 .
 5. Jika semua ya, cetak "Anda mendapatkan nilai A".
 6. Jika salah satu kondisi tidak terpenuhi, cetak pesan sesuai nilai yang kurang.

2) Tuliskan kode program dan luaran :

1.1 Program untuk Mengecek Bilangan Genap atau Ganjil:

```
import java.util.Scanner;

public class CekGenapGanjil {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan sebuah bilangan: ");
        int bilangan = input.nextInt();

        if (bilangan % 2 == 0) {
            System.out.println(bilangan + " adalah bilangan genap");
        } else {
            System.out.println(bilangan + " adalah bilangan ganjil");
        }
    }
}
```

Luaran :

Masukkan sebuah bilangan: 20
20 adalah bilangan genap

1.2 Analisis IF Bersarang dan Penambahan Kondisi Nilai Quiz

```
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();

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

        if (nilaiU  $\geq$  80) {
            if (nilaiT  $\geq$  80) {
```

```

        if (nilaiQ >= 80) {
            System.out.println("Anda mendapatkan nilai A");
        } else {
            System.out.println("Nilai Quiz Anda kurang dari 80");
        }
    } else {
        System.out.println("Nilai Tugas Anda kurang dari 80");
    }
} else {
    System.out.println("Nilai UTS Anda kurang dari 80");
}
}
}

```

Luaran :

Masukkan Angka Tugas Anda : 90

Masukkan Angka Quiz Anda : 100

Masukkan Angka UTS Anda : 95

Anda mendapatkan nilai A

1.3 Menyederhanakan Kondisi IF

Pertanyaan ini meminta kita untuk menyederhanakan tiga kondisi IF menjadi satu. Di sini kita perlu menggunakan operator logika && (dan) atau || (atau). Yang kita butuhkan adalah semua nilai (UTS, Tugas, dan Quiz) harus >= 80 agar siswa mendapat nilai A. Maka pilihan yang tepat adalah:

d. IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)

Penjelasan:

- Operator && digunakan untuk memastikan semua kondisi terpenuhi.
- Jika semua kondisi (nilaiU, nilaiT, nilaiQ) memenuhi syarat, barulah hasilnya benar.

a) Uraikan luaran yang dihasilkan :

Luaran yang dihasilkan bergantung pada input pengguna. Jika semua nilai >= 80, akan keluar pesan "Anda mendapatkan nilai A". Jika salah satu kurang dari 80, pesan yang relevan dengan nilai yang kurang akan ditampilkan.

b) Screenshot/ Capture potongan kode dan hasil luaran:

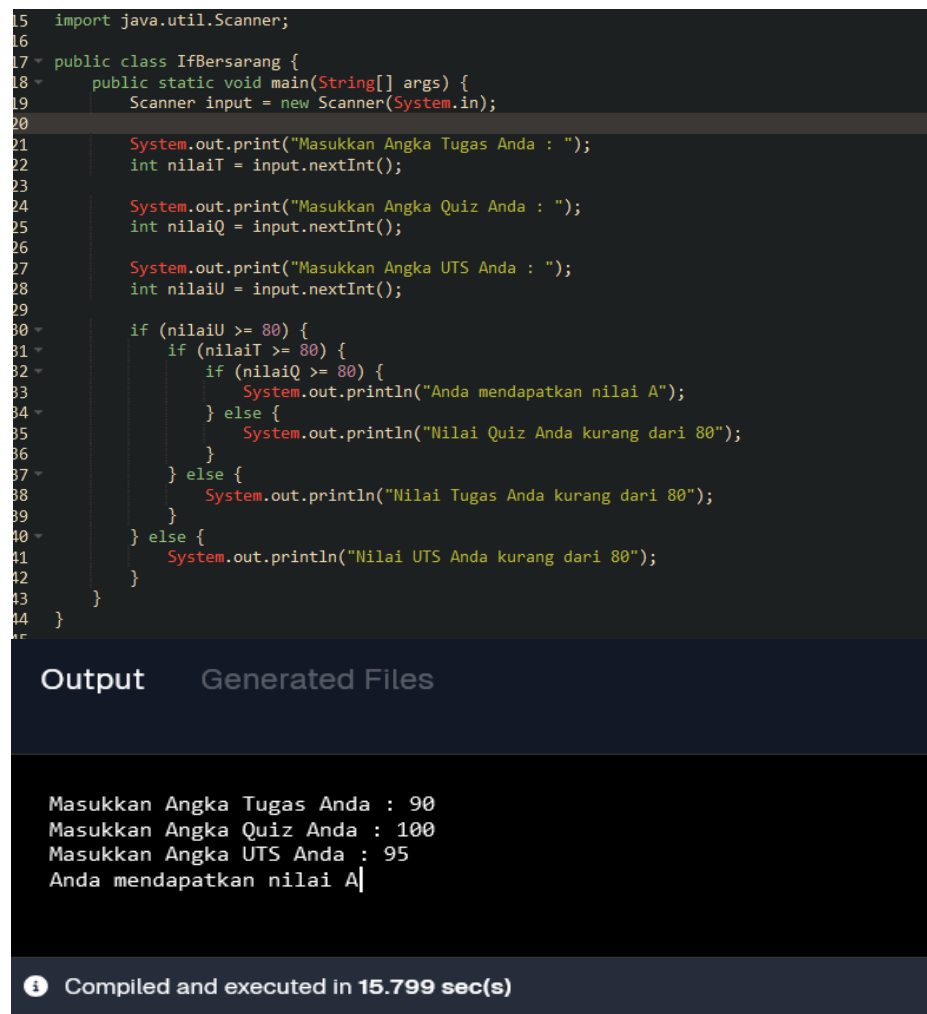
```

14
15 import java.util.Scanner;
16
17 public class CekGenapGanjil {
18     public static void main(String[] args) {
19         Scanner input = new Scanner(System.in);
20         System.out.print("Masukkan sebuah bilangan: ");
21         int bilangan = input.nextInt();
22
23         if (bilangan % 2 == 0) {
24             System.out.println(bilangan + " adalah bilangan genap");
25         } else {
26             System.out.println(bilangan + " adalah bilangan ganjil");
27         }
28     }
29 }

```



Gambar 1.1 Kode dan luaran Mengecek Bilangan Genap atau Ganjil



Gambar 1.2 Kode dan luaran IF Bersarang dan Penambahan Kondisi Nilai Quiz

[No.1] Kesimpulan

1) Analisa

- a) Berdasarkan permasalahan yang diberikan, algoritma menggunakan percabangan bersarang efektif untuk menentukan apakah nilai UTS, Tugas, dan Quiz memenuhi kriteria untuk mendapatkan nilai A. Kode program berjalan dengan baik untuk melakukan pengecekan kondisi ini.

Latihan 2

[No.2] Identifikasi Masalah:

1) Uraikan permasalahan dan variable:

- 2.1 Cermati kode pada Contoh 3. Hapuslah kode break; pada //baris 1, eksekusi kembali. Kemudian hapuslah kode break; pada //baris 2, eksekusi kembali. Analisis perbedaan hasil luaran ketika kode menggunakan break, ketika kode break baris 1 dihapus, dan ketika kode break baris 2 dihapus! Analisa kegunaan baris kode dengan break dan kata kunci default!
- 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? Jika bisa, rincikan analisa Anda! Bandingkan masalah yang dapat diselesaikan percabangan dengan IF atau SWITCH !
- 2.4 Buatlah dokumentasi gambar flowchart dari Latihan 2.1. dan Latihan 2.2!

2) Rincikan sumber informasi yang relevan :

Webpage: Dokumentasi Java dari Oracle(<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>) yang membahas pernyataan switch dan penggunaannya dalam pengambilan keputusan.

3) Uraikan rancangan solusi yang diusulkan:

- Pada Latihan 2.1, solusinya adalah dengan menghapus baris kode break; pada switch-case dan kemudian mengamati perbedaannya. Diharapkan bahwa setelah break; dihapus, alur eksekusi akan terus berjalan ke case berikutnya tanpa terhenti.
- Pada Latihan 2.2, solusi yang diusulkan adalah dengan menambahkan case bulan dari 5 hingga 12 dan memastikan perhitungan jumlah hari yang benar, termasuk perhitungan khusus untuk bulan Februari dalam tahun kabisat.

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara ,untuk :
 - Latihan 2.1 mengusulkan untuk mengeksekusi program tanpa break; pada case-case tertentu, yang memungkinkan kita melihat bagaimana program terus mengeksekusi tanpa berhenti pada case yang sesuai. Hal ini memberikan pemahaman mendalam tentang cara kerja switch-case dalam Java.
 - Latihan 2.2 mengusulkan penambahan case untuk setiap bulan, termasuk penanganan khusus untuk bulan Februari dalam tahun kabisat. Dengan demikian, program dapat menentukan jumlah hari dalam setiap bulan dengan benar.

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma:

- **Latihan 2.1:**
 1. Input data (A/B).
 2. Switch-case: A untuk pesan rajin belajar, B untuk kurang main game.
 3. Hapus break; dan amati eksekusi yang terjadi setelah setiap case.
- **Latihan 2.2:**
 1. Input bulan dalam angka (1–12).
 2. Tentukan jumlah hari untuk bulan tersebut, dengan pengecualian untuk Februari (periksa tahun kabisat).
 3. Output jumlah hari dalam bulan yang diinput.

2) Kode program dan luaran

2.1 Analisis Kode Contoh 3 (Break dan Default)

- **Penghapusan break; pertama**

```
import java.util.Scanner;
```

```
public class SwitchBersarang {  
    public static void main(String[] args) {  
        Scanner masukData = new Scanner(System.in);  
        System.out.print("Pilih A atau B : ");  
        char data = masukData.next().charAt(0);  
        switch(data) {  
            case 'A':  
                System.out.print("Anda sudah rajin belajar");  
                //hapus untuk pengujian  
            case 'B':  
                System.out.print("Anda perlu kurangi main game");  
                break; //hapus untuk pengujian  
            default:  
                System.out.print("Pilihan anda diluar A atau B");  
        }  
    }  
}
```

Luaran :

Pilih A atau B : A

Anda sudah rajin belajarAnda perlu kurangi main game

- **Penghapusan break; kedua**

```
import java.util.Scanner;
```

```
public class SwitchBersarang {  
    public static void main(String[] args) {  
        Scanner masukData = new Scanner(System.in);  
        System.out.print("Pilih A atau B : ");  
        char data = masukData.next().charAt(0);  
        switch(data) {  
            case 'A':  
                System.out.print("Anda sudah rajin belajar");  
                //hapus untuk pengujian  
            case 'B':  
                System.out.print("Anda perlu kurangi main game");  
                //hapus untuk pengujian  
            default:  
                System.out.print("Pilihan anda diluar A atau B");  
        }  
    }  
}
```

Luaran :

Pilih A atau B : A

Anda sudah rajin belajarAnda perlu kurangi main gamePilihan anda diluar A atau B

Penjelasan :

Saat kode break; pada baris 1 dihapus: Jika pengguna memasukkan 'A', program akan tetap menjalankan perintah pada case 'A', tetapi karena tidak ada

break, program akan melanjutkan eksekusi ke case 'B' dan mencetak output dari case 'B'. Sedangkan Saat kode break; pada baris 2 dihapus: Jika pengguna memasukkan 'B', program akan mencetak output dari case 'B' dan kemudian juga mencetak output dari default karena tidak ada break

2.2 Modifikasi Kode Contoh 4 (Sampai Bulan ke-12)

```
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: case 3: case 5: case 7: case 8: case 10: case 12:
                jumlahHari = 31;
                break;
            case 4: case 6: case 9: case 11:
                jumlahHari = 30;
                break;
            case 2:
                if (tahun % 4 == 0)
                    jumlahHari = 29;
                else
                    jumlahHari = 28;
                break;
            default:
                System.out.println("Maaf bulan hanya sampai 12.");
                break;
        }
        System.out.println("Jumlah hari = " + jumlahHari);
    }
}
```

Luaran :

Masukkan data bulan (dalam angka): 12

Jumlah hari = 31

2.3 Analisis Penggunaan IF dan SWITCH pada Contoh 3

Dapatkah masalah pada Contoh 3 diubah menjadi perintah IF? Ya, kode tersebut dapat diubah menjadi struktur IF. Kode menggunakan IF akan terlihat seperti ini:

```
if (data == 'A') {
    System.out.print("Anda sudah rajin belajar");
} else if (data == 'B') {
    System.out.print("Anda perlu kurang main game");
} else {
    System.out.print("Pilihan anda diluar A atau B");
}
```

Perbandingan IF dan SWITCH:

- **Switch:** lebih efisien jika ada beberapa pilihan kondisi tetap seperti 'A' atau 'B'. Switch juga lebih mudah dibaca untuk kasus-kasus sederhana seperti ini.
- **If-else:** lebih fleksibel untuk kondisi yang lebih kompleks dan bisa mengatasi lebih banyak tipe perbandingan seperti range nilai atau kondisi logis yang lebih rumit.

a) Screenshot/ Capture potongan kode dan hasil luaran

```
import java.util.Scanner;

public class SwitchBersarang {
    public static void main(String[] args) {
        Scanner masukData = new Scanner(System.in);
        System.out.print("Pilih A atau B : ");
        char data = masukData.next().charAt(0);
        switch(data) {
            case 'A':
                System.out.print("Anda sudah rajin belajar");
                //hapus untuk pengujian
            case 'B':
                System.out.print("Anda perlu kurangi main game");
                break; //hapus untuk pengujian
            default:
                System.out.print("Pilihan anda diluar A atau B");
        }
    }
}
```

Pilih A atau B : A
Anda sudah rajin belajarAnda perlu kurangi main game|

Compiled and executed in 23.321 sec(s)

Gambar 2.1 tanpa break; baris 1

```
import java.util.Scanner;

public class SwitchBersarang {
    public static void main(String[] args) {
        Scanner masukData = new Scanner(System.in);
        System.out.print("Pilih A atau B : ");
        char data = masukData.next().charAt(0);
        switch(data) {
            case 'A':
                System.out.print("Anda sudah rajin belajar");
                //hapus untuk pengujian
            case 'B':
                System.out.print("Anda perlu kurangi main game");
                //hapus untuk pengujian
            default:
                System.out.print("Pilihan anda diluar A atau B");
        }
    }
}
```



```
Pilih A atau B : A
Anda sudah rajin belajarAnda perlu kurangi main gamePilihan anda diluar A atau B

Compiled and executed in 88.894 sec(s)
```

Gambar 2.2 tanpa break; baris 1 dan 2

```
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: case 3: case 5: case 7: case 8: case 10: case 12:
                jumlahHari = 31;
                break;
            case 4: case 6: case 9: case 11:
                jumlahHari = 30;
                break;
            case 2:
                if (tahun % 4 == 0)
                    jumlahHari = 29;
                else
                    jumlahHari = 28;
                break;
            default:
                System.out.println("Maaf bulan hanya sampai 12.");
                break;
        }
        System.out.println("Jumlah hari = " + jumlahHari);
    }
}
```

```
Masukkan data bulan (dalam angka): 12
Jumlah hari = 31

Compiled and executed in 49.858 sec(s)
```

Gambar 2.3 telah ditambahkan case untuk bulan 5 hingga 12

[No.2] Kesimpulan

1) Evaluasi

- Konsekuensi dari skenario pemrograman: Pada Latihan 2.1, tanpa break;, alur kontrol akan berjalan melewati case yang dituju. Pada Latihan 2.2, logika yang tepat untuk bulan Februari dalam tahun kabisat memastikan program menghasilkan perhitungan yang akurat.
- Evaluasi input, proses, dan luaran: Input bulan dan tahun diterima dengan benar, proses logika switch-case berjalan sesuai dengan desain, dan luaran berupa jumlah hari sesuai bulan ditampilkan dengan benar.