

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Karina Hadiyah Ramadona G1F024040	IF	24 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 masuk = new Scanner(System.in); // membaca teks yang dimasukkan pengguna
 System.out.print("Masukkan Angka Anda : "); //pengguna memasukkan data
 int nilai = masuk.nextByte(); //menyimpan masukan pengguna ke tipe data

if (nilai == 10) { //percabangan yang memeriksa kondisi
 System.out.println("Sepuluh"); //baris kode yang dieksekusi bila benar
 }
else { //baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
 System.out.println("Nilai Bukan Sepuluh");
 } }
}

Luaran Contoh 1:

Masukkan Angka Anda : 8

Nilai Bukan Sepuluh

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. 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.

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!

2) Rincikan sumber informasi yang relevan (buku / webpage)

<https://www.youtube.com/watch?v=G0dfdAFa9iM>

[No.1] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

1.1 Saya mengusulkan permasalahan ini dapat diatasi dengan cara memeriksa apakah nilai yang dimasukkan oleh pengguna habis dibagi 2 atau tidak. Hal ini bisa dilakukan dengan menggunakan operator modulus (%). Jika hasilnya 0, maka bilangan tersebut genap. Jika hasilnya bukan 0 (khususnya 1), maka bilangan tersebut ganjil.

Solusi ini dipilih karena konsep dasar bilangan genap dan ganjil bergantung pada hasil pembagian oleh angka 2. Penggunaan operator modulus (%) sangat cocok untuk mengecek sisa pembagian suatu angka, dan jika sisa pembagiannya 0, maka bilangan tersebut genap, jika tidak, maka ganjil. Ini merupakan pendekatan yang sederhana dan efisien.

1.2 Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan kondisi if bersarang yang baru untuk memeriksa nilai Quiz setelah pengecekan nilai UTS dan nilai Tugas.

Alasan solusi ini karena untuk menentukan apakah siswa mendapatkan nilai A, diperlukan pengecekan terhadap tiga nilai (UTS, Tugas, dan Quiz). Jika ketiganya memenuhi syarat (lebih besar atau sama dengan 80), siswa akan mendapatkan nilai A.

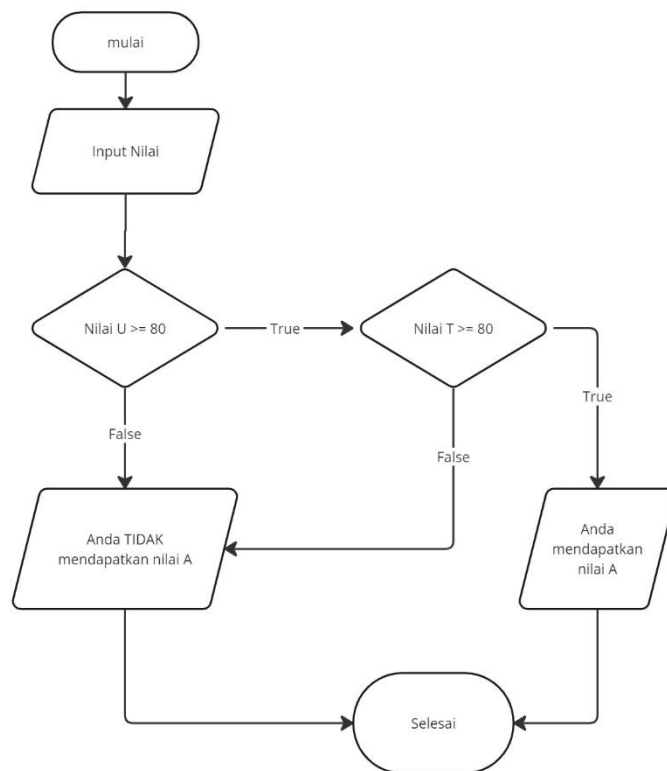
Perbaiki kode program dengan cara menambahkan satu kondisi IF bersarang baru untuk nilai Quiz. Jika nilai UTS, Tugas, dan Quiz semua lebih besar atau sama dengan 80, maka siswa akan mendapatkan nilai A.

1.3 Saya mengusulkan permasalahan ini dapat diatasi dengan cara meringkas ketiga kondisi IF yang memeriksa nilai UTS, Tugas, dan Quiz menjadi satu kondisi saja. Ini akan membuat kode lebih ringkas dan mudah dipahami.

Alasan solusi ini karena kondisi if bersarang bisa diringkas menjadi satu if statement yang langsung mengecek ketiga nilai dalam satu perbandingan. Untuk menentukan apakah

siswa mendapatkan nilai A, diperlukan ketiga nilai (UTS, Tugas, dan Quiz) harus lebih besar atau sama dengan 80.

Perbaiki kode program dengan cara menggunakan opsi d: IF (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80). Ini adalah kondisi yang paling tepat karena menggunakan operator logika AND (&&) untuk memastikan bahwa semua kondisi harus terpenuhi. Artinya, siswa hanya mendapatkan nilai A jika semua nilai UTS, Tugas, dan Quiz lebih besar atau sama dengan 80. Opsi lain menggunakan operator OR (| |) yang akan memberikan nilai A jika salah satu kondisi terpenuhi, yang tidak sesuai dengan aturan yang diinginkan.



1.4

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 1. Mulai.
 2. Masukkan nilai UTS, Tugas, dan Quiz dari pengguna:
 3. Periksa kondisi apakah semua nilai (UTS, Tugas, dan Quiz) lebih besar atau sama dengan 80:
 4. Selesai.
- 2) Tuliskan kode program dan luaran
 - 2.1.1)

```

1
2 // latihan 1 1.1
3 import java.util.Scanner; //memanggil impor package yang membaca masukan pengguna
4
5 public class PercabanganIf {
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.nextByte(); // menyimpan masukan pengguna ke tipe data
10
11         if (nilai % 2 == 0) { // percabangan yang memeriksa kondisi
12             System.out.println("Nilai Bilangan Genap"); // baris kode yang dieksekusi bila benar
13         } else { // baris kode yang dieksekusi bila kondisi tidak terpenuhi dan salah
14             System.out.println("Nilai Bukan Bilangan Genap");
15         }
16     }
17 }

```

Luaran

```

Masukkan Angka Anda : 12
Nilai Bilangan Genap

```

```

C:\koprog.java\latihan4>

```

```

1 import java.util.Scanner; // Import kelas Scanner untuk input dari pengguna
2
3 public class IfBersarang {
4     public static void main(String[] args) {
5         // Membuat Scanner untuk input nilai tugas
6         Scanner varT = new Scanner(System.in);
7         System.out.print("Masukkan Angka Tugas Anda : ");
8         int nilaiT = varT.nextByte(); // Mengambil input nilai tugas
9
10        // Membuat Scanner untuk input nilai quiz
11        Scanner varQ = new Scanner(System.in);
12        System.out.print("Masukkan Angka Quiz Anda : ");
13        int nilaiQ = varQ.nextByte(); // Mengambil input nilai quiz
14
15        // Membuat Scanner untuk input nilai ujian
16        Scanner varU = new Scanner(System.in);
17        System.out.print("Masukkan Angka Ujian Anda : "); // Perbaiki pesan ini
18        int nilaiU = varU.nextByte(); // Mengambil input nilai ujian
19
20        // Memeriksa apakah nilai ujian >= 80
21        if (nilaiU >= 80) {
22            // Memeriksa apakah nilai tugas >= 80
23            if (nilaiT >= 80) {
24                // Memeriksa apakah nilai quiz >= 80
25                if (nilaiQ >= 80) {
26                    System.out.println("Anda mendapatkan nilai A"); // Jika semua syarat terpenuhi
27                }
28            }
29        } else {
30            System.out.println("Anda TIDAK mendapatkan nilai A"); // Jika nilai ujian < 80
31        }
32    }
33 }
34

```

2.1.2)

Luaran

```

Masukkan Angka Tugas Anda : 80
Masukkan Angka Quiz Anda : 80
Masukkan Angka Quiz Anda : 80
Anda mendapatkan nilai A

```

```

C:\koprog.java\latihan4>

```

2.1.3)

```
1 import java.util.Scanner; // Mengimpor kelas Scanner untuk mengambil input dari pengguna
2
3 public class IfBersarang {
4     public static void main(String[] args) {
5         // Membuat objek Scanner untuk input nilai tugas
6         Scanner varT = new Scanner(System.in);
7         System.out.print("Masukkan Angka Tugas Anda : ");
8         int nilaiT = varT.nextByte(); // Mengambil input nilai tugas
9
10        // Membuat objek Scanner untuk input nilai quiz
11        Scanner varQ = new Scanner(System.in);
12        System.out.print("Masukkan Angka Quiz Anda : ");
13        int nilaiQ = varQ.nextByte(); // Mengambil input nilai quiz
14
15        // Membuat objek Scanner untuk input nilai ujian
16        Scanner varU = new Scanner(System.in);
17        System.out.print("Masukkan Angka Ujian Anda : "); // Perbaiki pesan ini
18        int nilaiU = varU.nextByte(); // Mengambil input nilai ujian
19
20        // Memeriksa apakah semua nilai >= 80
21        if (nilaiU >= 80 && nilaiT >= 80 && nilaiQ >= 80)
22            System.out.println("Anda mendapatkan nilai A"); // Jika semua syarat terpenuhi
23        else {
24            System.out.println("Anda TIDAK mendapatkan nilai A"); // Jika tidak memenuhi syarat
25        }
26    }
27 }
28
```

Luaran

```
Masukkan Angka Tugas Anda : 70
Masukkan Angka Quiz Anda : 70
Masukkan Angka Ujian Anda : 70
Anda TIDAK mendapatkan nilai A

c:\koprog.java\latihan4>
```

[No.1] Kesimpulan

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

Pada permasalahan ini, kita dihadapkan pada kebutuhan untuk memeriksa apakah nilai UTS, Tugas, dan Quiz memenuhi syarat tertentu untuk mendapatkan nilai A. Permasalahan ini diselesaikan dengan meringkas tiga kondisi IF bersarang menjadi satu kondisi IF sederhana yang memeriksa apakah semua nilai lebih besar atau sama dengan 80.

Algoritma yang digunakan mengikuti langkah-langkah logis untuk menerima input dari pengguna, memeriksa ketiga nilai, dan menentukan hasilnya. Kode program kemudian diimplementasikan berdasarkan algoritma tersebut, dengan menggunakan satu pernyataan IF yang memeriksa kondisi ketiga nilai sekaligus. Program menjadi lebih ringkas dan efisien, serta mudah dipahami.
 - b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Perbaiki program dengan menambahkan satu kondisi IF untuk memeriksa nilai UTS, Tugas, dan Quiz sekaligus karena struktur Java mengharuskan kita menggunakan operator logika yang sesuai (AND (&&)) untuk memeriksa semua syarat dalam satu pernyataan. Hal ini mempermudah pengecekan kondisi secara simultan tanpa perlu menggunakan IF bersarang yang memperumit program. Pada program awal, digunakan IF bersarang yang memeriksa nilai UTS, Tugas, dan Quiz secara bertahap. Namun, karena struktur logika memerlukan semua syarat terpenuhi, kita dapat menyederhanakan kondisi tersebut menjadi satu pernyataan IF yang menggunakan operator AND (&&).

Nama & NPM	Topik:	Tanggal:
Karina Hadiyah Ramadona G1F024040	SWITCH	24 September 2024

[No.2] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

```
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.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 ");
        } } }
```

Luaran Contoh 3:

Pilih A atau B : A
Anda sudah rajin belajar

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: case 3: //baris 1
                jumlahHari = 31;
                break;
            case 4: //baris 2
                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 Contoh 4:

Masukkan data bulan (dalam angka): 7

Jumlah hari = 31

Latihan 2:

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 (buku / webpage)

<https://www.youtube.com/watch?v=G0dfdAFa9iM>

[No.2] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

2.1 1) Hapus kode break; pada baris 1 (case 'A')

Hasil Luaran: Saat kita memilih 'A', hasil yang dicetak tidak hanya mencetak "Anda sudah rajin belajar" tetapi juga akan mencetak pesan dari case 'B' ("Anda perlu kurangi main game"). Hal ini terjadi karena tanpa break;, eksekusi tidak berhenti setelah case 'A' dijalankan, dan lanjut ke case 'B'.

2) Hapus kode break; pada baris 2 (case 'B')

Hasil Luaran: Saat kita memilih 'B', program akan mencetak pesan "Anda perlu kurangi main game", dan kemudian dilanjutkan ke blok default, sehingga juga akan mencetak "Pilihan anda diluar A atau B". Ini disebabkan karena tanpa break;, program tidak berhenti setelah case 'B' dan melanjutkan ke default.

Analisis Kegunaan break; dan default:

Kegunaan break:

Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan perintah **break**; untuk menghentikan eksekusi setelah satu case selesai dijalankan. Tanpa **break**;, program akan melanjutkan eksekusi ke case berikutnya, menyebabkan hasil yang tidak diinginkan.

Alasan solusi ini karena perintah **break**; digunakan untuk mencegah **fall-through**, yaitu situasi di mana setelah sebuah case dijalankan, case berikutnya ikut dijalankan meskipun kondisinya tidak sesuai.

Perbaiki kode program dengan cara menambahkan **break**; pada setiap case dalam pernyataan **switch** untuk memastikan bahwa hanya satu case yang dijalankan dan program tidak melanjutkan eksekusi ke case yang lain.

Kata kunci default digunakan untuk menangani input yang tidak sesuai dengan salah satu nilai yang diperiksa dalam case-case sebelumnya. Ketika tidak ada case yang cocok, blok default akan dijalankan sebagai penanganan terakhir.

2.2 Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan case yang mewakili bulan 5 hingga bulan 12 dalam blok **switch** sehingga program dapat memproses semua bulan dalam setahun.

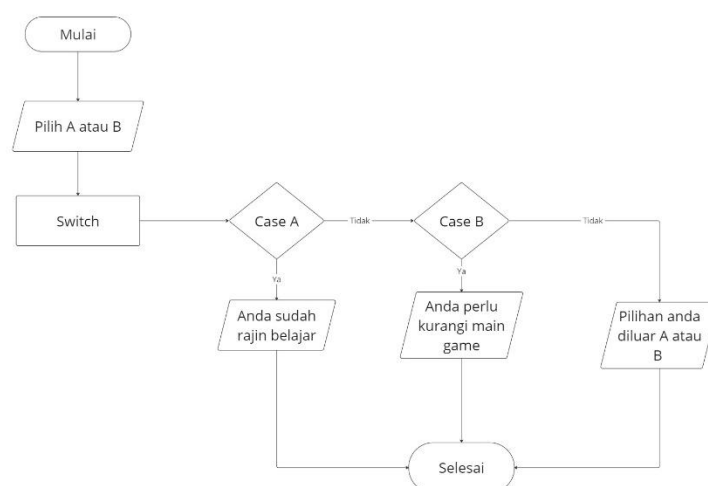
Alasan solusi ini karena pada kasus aslinya, program hanya memproses sampai bulan ke-4, sehingga perlu ditambahkan kondisi untuk bulan-bulan lainnya agar program dapat menampilkan jumlah hari untuk setiap bulan dari Januari hingga Desember.

Perbaiki kode program dengan cara menambahkan case-case untuk bulan 5 hingga bulan 12 dan mengatur jumlah hari sesuai dengan jumlah hari di setiap bulan. Selain itu, tambahkan logika untuk menangani bulan Februari (pada tahun kabisat) agar program berfungsi dengan benar untuk semua bulan.

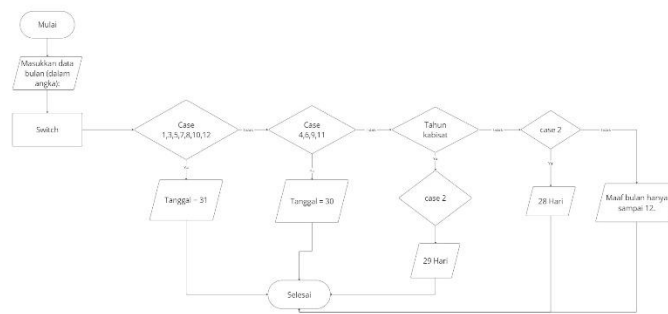
2.3 karena struktur **switch case** dapat diubah menjadi perintah **if-else** dalam kasus ini.

Struktur **switch case** pada dasarnya memeriksa satu nilai dan mengeksekusi blok kode tertentu berdasarkan nilai tersebut, yang juga dapat dilakukan oleh perintah **if-else**.

2.4 Flowchart 2.1



Flowchart 2.2



[No.2] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 1. Mulai.
 2. Minta pengguna untuk memasukkan karakter ('A' atau 'B').
 3. Baca input karakter dari pengguna.
 4. Selesai..
- 2) Tuliskan kode program dan luaran

```

1 import java.util.Scanner; // Mengimpor kelas Scanner untuk mengambil input dari pengguna
2
3 public class SwitchBersarang {
4     public static void main(String[] args) {
5         byte bulan; // Variabel untuk menyimpan bulan (1-12)
6         int tahun = 2022; // Tahun tetap, dapat diperluas untuk menerima input pengguna
7         int jumlahHari = 0; // Variabel untuk menyimpan jumlah hari dalam bulan tersebut
8
9         // Mengambil input bulan dari pengguna
10        System.out.print("Masukkan data bulan (dalam angka): ");
11        Scanner masukData = new Scanner(System.in);
12        bulan = masukData.nextByte(); // Membaca input bulan
13
14        // Menggunakan switch untuk menentukan jumlah hari berdasarkan bulan
15        switch (bulan) {
16            case 1: // Januari
17            case 3: // Maret
18            case 5: // Mei
19            case 7: // Juli
20            case 8: // Agustus
21            case 10: // Oktober
22            case 12: // Desember
23                jumlahHari = 31; // Bulan dengan 31 hari
24                break;
25            case 4: // April
26            case 6: // Juni
27            case 9: // September
28            case 11: // November
29                jumlahHari = 30; // Bulan dengan 30 hari
30                break;
31            case 2: // Februari
32                // Periksa apakah tahun Kabisat
33                if (tahun % 4 == 0) // Tahun kabisat jika habis dibagi 4
34                    jumlahHari = 29; // Februari di tahun kabisat
35                else
36                    jumlahHari = 28; // Februari di tahun non-kabisat
37                break;
38            default:
39                System.out.println("Maaf bulan hanya sampai 12."); // Pesan kesalahan jika bulan tidak valid
40                break;
41        }
42
43        // Menampilkan jumlah hari dalam bulan yang dimasukkan
44        System.out.println("Jumlah hari = " + jumlahHari);
45    }
46 }
47

```

2.2

Luaran

```

Masukkan data bulan (dalam angka): 10
Jumlah hari = 31

c:\koprog.java\latihan4>
  
```

2.3

```
1 import java.util.Scanner; // Mengimpor kelas Scanner untuk mengambil input dari pengguna
2
3 public class SwitchBersarang {
4     public static void main(String[] args) {
5         Scanner masukData = new Scanner(System.in); // Membuat objek Scanner untuk input
6
7         // Mengambil input dari pengguna
8         System.out.print("Pilih A atau B : ");
9         char data = masukData.next().charAt(0); // Membaca input karakter pertama dari pengguna
10
11        // Menggunakan if statements untuk mengevaluasi pilihan
12        if (data == 'A') { // Memeriksa apakah input adalah 'A'
13            System.out.print("Anda sudah rajin belajar"); // Output jika pilihan A
14        } else if (data == 'B') { // Memeriksa apakah input adalah 'B'
15            System.out.print("Anda perlu kurangi main game"); // Output jika pilihan B
16        } else {
17            System.out.print("Pilihan anda diluar A atau B"); // Output jika input tidak valid
18        }
19
20        masukData.close(); // Menutup Scanner untuk menghindari kebocoran sumber daya
21    }
22 }
23
```

Luaran

```
Pilih A atau B : A
Anda sudah rajin belajar
C:\koprog.java\latihan4>
```

[No.2] Kesimpulan

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

Berdasarkan permasalahan yang dihadapi, program awal menggunakan struktur switch case untuk memeriksa input pengguna ('A' atau 'B') dan memberikan respons yang sesuai. Namun, untuk menyederhanakan proses dan mengatasi masalah dengan lebih jelas, struktur if-else diusulkan sebagai alternatif.

Dalam algoritma yang telah disusun, langkah-langkah yang jelas menggambarkan bagaimana program meminta input, memeriksa kondisi menggunakan if-else, dan memberikan hasil berdasarkan input tersebut. Kode program yang diperbarui dengan menggunakan if-else berhasil memenuhi tujuan yang sama dengan switch case namun dengan cara yang lebih sederhana dan mudah dipahami.
 - b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Perbaiki program dengan menambahkan struktur kontrol if-else karena struktur Java mengharuskan penggunaan kontrol alur yang jelas dan sederhana. Dengan menggunakan if-else, logika pemrograman menjadi lebih eksplisit dan memudahkan pemahaman bagi pengembang dan pembaca kode. Struktur ini juga membantu menghindari kebingungan yang dapat muncul dari penggunaan switch case, terutama ketika jumlah pilihan terbatas.