Tugas 1

Nama & NPM	**Topik:**	**Tanggal:**
Abdullah Hasyim Syauqi G1F024019	**Unit 1 IF**	**16-09-2024**

Identifikasi Masalah:

1.1 Evaluasi penyebab kesalahan dan perbaiki kode tersebut!
Contoh 1

- Tambahkan byte disamping nilai
- Ganti masuk menjadi input
- Mengganti (=) menjadi (==) untuk perbandingan di kondisi if.

Contoh 2

- Ganti nilaiU menjadi nilaiQ

1.2 Ya, ketiga kondisi if dalam contoh dapat diringkas menjadi satu kondisi menggunakan operator boolean &&. Ini memungkinkan Anda untuk memeriksa semua syarat sekaligus, sehingga kode menjadi lebih ringkas dan efisien.

1.3 Konstruksikan kode program untuk menghasilkan luaran berdasarkan informasi berikut!
Persyaratan kelulusan bagi mahasiswa Informatika dan Sistem Informasi Universitas Bengkulu yaitu:

a) Jumlah sks yang lulus minimum = 144 sks. Jika kurang dari 144 maka perlu mengulang mengambil mata kuliah hingga memenuhi
b) Nilai tes toefl minimum = 430. Jika skor belum mencapai maka ulangi tes toefl maksimal 3x atau hingga skor tercapai
c) Ujian skripsi = Lulus, jika belum lulus maka perlu ujian ulang.
d) Menulis artikel ilmiah dari skripsi yang dibuat dalam status = submitted. Jika bukti penerimaan submitted tidak dilampirkan maka persyaratan belum terpenuhi.

Jika keempat syarat ini terpenuhi maka Mahasiswa layak lulus dengan gelar S.Kom. (Sarjana Komputer).

1.4 Desain gambar diagram flowchart dari Latihan 1.3!

Penyusunan Algoritma dan Kode Program

1.1
Contoh 1 :

```
import java.util.Scanner;

public class PercabanganIf {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.print("Masukkan Angka Anda : ");
      byte nilai = input.nextByte();

      if (nilai == 1000) {
         System.out.println("Seribu");
      } else {
         System.out.println("Nilai Bukan Seribu");
      }
   }
}
```

Contoh 2 :

```
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();

      if (nilaiQ >= 80) {
         if (nilaiT >= 80) {
            System.out.println("Anda mendapatkan nilai A");
         }
      } else {
         System.out.println("Anda TIDAK mendapatkan nilai A");
      }
   } }
```

1.2

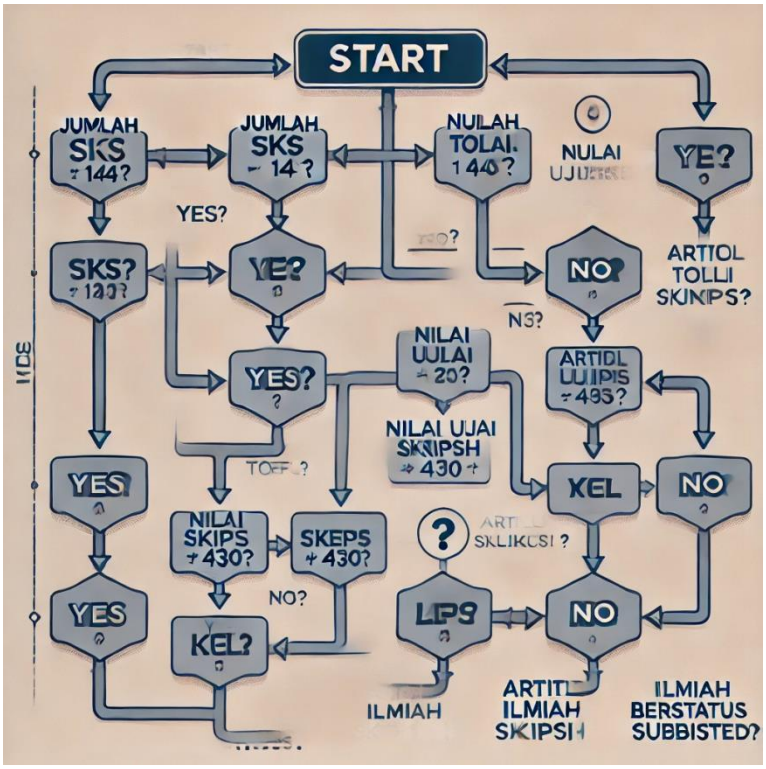
```
import java.util.Scanner;

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 nilaiUTS = input.nextInt();

      // Ringkasan kondisi IF menggunakan operator &&
      if (nilaiT >= 80 && nilaiQ >= 80 && nilaiUTS >= 80) {
         System.out.println("Anda mendapatkan nilai A");
      } else {
         System.out.println("Anda TIDAK mendapatkan nilai A");
      }
   }
}
```

1.3

```
import java.util.Scanner;

public class Hasyim {
   public static void main(String[] args) {
      Scanner input = new Scanner(System.in);

      // Masukkan jumlah sks yang telah lulus
```

```
        System.out.print("Masukkan jumlah SKS yang telah lulus: ");
        int sks = input.nextInt();

        // Masukkan nilai tes TOEFL
        System.out.print("Masukkan nilai tes TOEFL: ");
        int toefl = input.nextInt();

        // Cek apakah mahasiswa sudah mengikuti ujian skripsi
        System.out.print("Apakah Anda sudah lulus ujian skripsi? (ya/tidak): ");
        String ujianSkripsi = input.next();

        // Cek apakah artikel ilmiah sudah dalam status 'submitted'
        System.out.print("Apakah artikel ilmiah Anda sudah berstatus 'submitted'? (ya/tidak): ");
        String artikelIlmiah = input.next();

        // Memeriksa kelulusan berdasarkan persyaratan
        if (sks >= 144) {
            if (toefl >= 430) {
                if (ujianSkripsi.equalsIgnoreCase("ya")) {
                    if (artikelIlmiah.equalsIgnoreCase("ya")) {
                        System.out.println("Selamat! Anda layak lulus dengan gelar S.Kom.");
                    } else {
                        System.out.println("Persyaratan belum terpenuhi: Artikel ilmiah belum berstatus
'submitted'.");
                    }
                } else {
                    System.out.println("Persyaratan belum terpenuhi: Anda belum lulus ujian skripsi.");
                }
            } else {
                System.out.println("Persyaratan belum terpenuhi: Nilai TOEFL kurang dari 430.");
            }
        } else {
            System.out.println("Persyaratan belum terpenuhi: SKS kurang dari 144.");
        }
    }
}
```

1.4 Desain gambar diagram flowchart dari Latihan 1.3!



Saya meminta bantuan artificial intiligent untuk membantu membuat flowchart dari kode yang saya buat.

Kesimpulan

Dalam tugas ini, kita telah belajar berbagai konsep pemrograman, mulai dari penggunaan **struktur kendali if-else** sederhana hingga implementasi **if bersarang** dan penggunaan **operator boolean** untuk menyederhanakan logika program. Selain itu, kita juga mempelajari cara memeriksa beberapa persyaratan kelulusan mahasiswa dengan menggunakan pemrograman Java, serta merancang flowchart untuk menggambarkan alur kelulusan tersebut.

Refleksi

Soal diatas ini menunjukkan bagaimana logika pemrograman dapat diterapkan untuk memecahkan masalah nyata, seperti pengecekan syarat kelulusan mahasiswa. Kita belajar pentingnya menyusun logika yang efisien menggunakan struktur kendali dan operator boolean untuk menyederhanakan proses evaluasi kondisi kompleks. Selain itu, merancang flowchart formal membantu visualisasi langkah-langkah yang jelas, yang sangat berguna dalam memahami alur keputusan dalam sebuah program. Hal ini menggarisbawahi pentingnya berpikir sistematis dalam pemrograman dan pengembangan solusi.

Tugas 2

Nama & NPM	Topik:	Tanggal:
Abdullah Hasyim Syauqi G1F024019	**Unit 2 SWITCH**	**16-09-2024**

Identifikasi Masalah:

2.1 Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

- Ganti data dengan masukData saat mengambil input, karena kita sudah mendeklarasikan Scanner dengan nama masukData
- Di dalam switch, kita harus menggunakan {} sebagai penanda blok.
- Pastikan karakter dalam case (A dan B) ditulis dalam tanda petik tunggl ('A', 'B').
- Setelah default, tambahkan : untuk mengikuti sintaks yang benar

2.2

a) Evaluasi apakah penulisan kode tersebut sudah efisien?
Penulisan kode tersebut berfungsi dengan baik, namun dapat disederhanakan dan dioptimalkan lebih lanjut untuk menghindari redundansi dan meningkatkan efisiensi

b) Apakah ada penulisan informasi yang diulangi?
Ya, terdapat beberapa informasi yang diulangi
- Pengulangan Jumlah Hari untuk Bulan
- Pengulangan Validasi Tahun Kabisat

c) Jika ada, susun kembali penulisan kode yang tepat! (ada di penyusunan algoritma dibawah)

d) Simpulkan perbedaan antara kode Contoh 4 dengan kode yang kalian susun!
- **Struktur**: Kode lama menggunakan switch-case dengan pengulangan nilai jumlah hari, sedangkan kode baru menggunakan array untuk menyimpan jumlah hari, membuatnya lebih ringkas.
- **Tahun Kabisat:** Kode lama hanyamemeriksa tahun kabisat dengan kondisi sederhana, sementara kode baru menggunakan aturan kabisat yang lebih akurat (kelipatan 4, 100, dan 400)
- **Validasi**: Kode baru menambahkan validasi input bulan (1-12), sedangkan kode lama tidak memiliki validasi yang jelas
- **Efisiensi**: Kode baru lebih efisien, mudah dibaca, dan lebih mudah dipelihara karena mengurangi pengulangan dan memisahkan logika kabisat dari switch-case

2.3

a) Apakah masalah ini bisa diubah menjadi perintah IF?
Ya, kode tersebut dapat diubah menjadi perintah if-else dengan mudah.

b) Jika bisa, susun kembali kode Contoh 3 dari perintah SWITCH menjadi IF!(ada di penyusunan algoritma dibawah)

c) Simpulkan perbandingan masalah dan kode program yang dapat diselesaikan percabangan dengan IF atau SWITCH !
- if-else:
  Cocok untuk kondisi kompleks dengan perbandingan logika atau rentang nilai. Lebih fleksibel, tapi bisa sulit dibaca jika ada banyak kondisi.
- switch-case:
  Ideal untuk memeriksa satu variabel dengan banyak nilai tetap (seperti angka atau karakter).Lebih efisien dan mudah dibaca dalam kasus sederhana, tapi kurang fleksibel untuk kondisi kompleks.

2.4 Desain gambar flowchart dari Latihan 2.1. dan Latihan 2.3! !(ada di penyusunan algoritma dibawah)

Penyusunan Algoritma dan Kode Program

2.1

a. Hapuslah kode break; pada //baris 1, lalu eksekusi kembali.

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

Gambar 2.1.a luaran Hapuslah kode break; pada //baris 1, lalu eksekusi kembali.

b. Kemudian hapuslah kode break; pada //baris 2, lalu eksekusi kembali.

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

Gambar 2.1.b luaran hapuslah kode break; pada //baris 2, lalu eksekusi kembali.

Kesipulannya, Tanpa break program akan terus mengeksekusi semua case yang berada setelah case yang cocok, hingga menemukan break; atau akhir blok switch. Ini dapat menyebabkan hasil yang tidak diinginkan dan sulit dipahami.

2.2

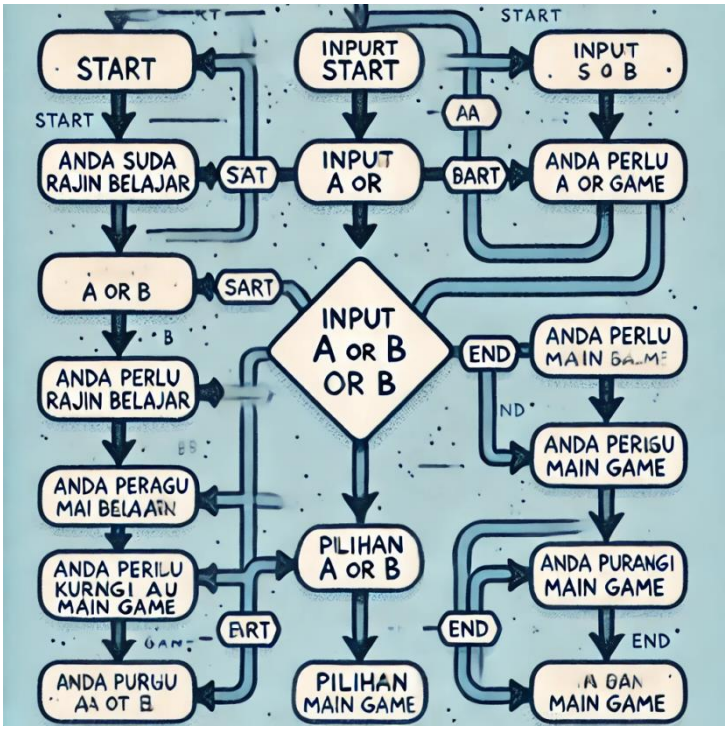
```
import java.util.Scanner;

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();

      // Array untuk jumlah hari pada setiap bulan
      int[] hariDalamBulan = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

      if (bulan < 1 || bulan > 12) {
         System.out.println("Maaf bulan hanya sampai 12.");
      } else {
         // Cek tahun kabisat hanya untuk bulan Februari
         if (bulan == 2 && (tahun % 4 == 0 && (tahun % 100 != 0 || tahun % 400 == 0))) {
            jumlahHari = 29; // Tahun kabisat
         } else {
            jumlahHari = hariDalamBulan[bulan - 1];
         }
         System.out.println("Jumlah hari = " + jumlahHari);
      }
   }
}
```

2.3

```
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);

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

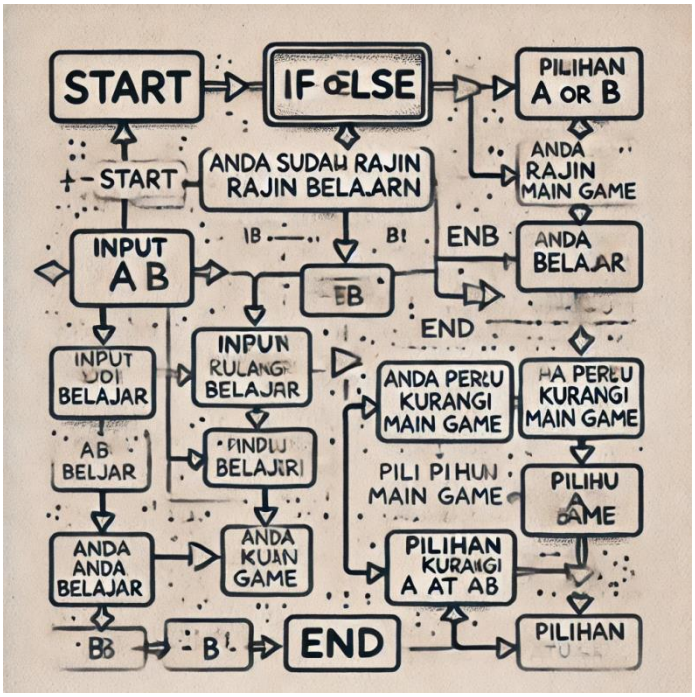
2.4 Desain gambar flowchart dari Latihan 2.1. dan Latihan 2.3

a) Gambar flowchart 2.1



Saya meminta bantuan artificial intiligent untuk membantu membuat flowchart dari kode yang saya buat.

b) Gambar flowchart 2.3



Saya meminta bantuan artificial intiligent untuk membantu membuat flowchart dari kode yang saya buat.

Kesimpulan

Kesimpulannya adalah Dalam kode diatas kita dapat membandingkan dua struktur kontrol utama dalam pemrograman Java, yaitu switch-case dan if-else. Keduanya memiliki peran penting dalam mengatur alur program berdasarkan input atau kondisi tertentu.

Refleksi

Dari kode diatas, kita dapat melihat bahwa pemilihan struktur kontrol bergantung pada kompleksitas masalah yang dihadapi. Pendekatan yang lebih efisien dan terstruktur dapat menghasilkan kode yang lebih mudah dipelihara dan dibaca, seperti yang terlihat dari penggunaan array untuk mengoptimalkan logika perhitungan jumlah hari dalam sebulan. Selain itu, desain flowchart membantu dalam memvisualisasikan alur logika dan memberikan panduan dalam memahami kode secara keseluruhan.