Nama & NPM	Topik:	Tanggal:
1. **Abullah Hasyim Syauqi (G1F024019)** 2. **Sindi Putri utami (G1F024053)** 3. **Zaira Ayu Wandira (G1F024055)**	**Operator java**	**5 september 2024**

Identifikasi Masalah:

1. Bila kalian kuliah selama 4 tahun.
   Susunlah kode java untuk menghitung jumlah yang harus dibayar selama kuliah untuk setiap anggota kelompok!
   (Asumsi: setiap orang memiliki jumlah yang dibayarkan berbeda karena perbedaan jalur masuk, UKT, SPP)

2. Gunakan operator ternary.
   Rancanglah kode Java untuk menghitung jumlah yang harus dibayar jika kalian bisa selesai 4 tahun atau jika selesai 5 tahun!

Analisis dan Argumentasi

Analisis:

1. Kelas Mahasiswa digunakan untuk menyimpan data yang relevan terkait biaya kuliah masing-masing mahasiswa, yaitu nama, biaya UKT, dan biaya SPP per semester.
2. Metode hitungTotalBiaya() di kelas Mahasiswa menghitung total biaya kuliah selama 4 tahun dengan mengalikan jumlah semester (8) dengan total biaya per semester (UKT + SPP).
3. Input biaya UKT dan SPP diambil dalam format dengan pemisah ribuan (misalnya 3.450.000). Program memanfaatkan metode parseRupiah untuk membersihkan dan mengkonversi input menjadi tipe double.
4. Total biaya kuliah selama 4 tahun dihitung dan ditampilkan dengan format desimal yang menyertakan pemisah ribuan menggunakan DecimalFormat

Argumentasi:

Kelebihan:

1. pemodelan Data: Menyimpan data dalam kelas terpisah adalah pendekatan yang baik karena memudahkan pengelolaan data terkait mahasiswa.
2. Enkapsulasi: Kelas Mahasiswa dengan metode perhitungan memisahkan logika bisnis dari input dan output, yang meningkatkan modularitas dan pemeliharaan kode.
3. Fleksibilitas: Metode parseRupiah memungkinkan pengguna untuk memasukkan angka dalam format yang umum digunakan di beberapa negara, seperti pemisah ribuan dengan titik.
4. Robustness: Penanganan kesalahan yang baik memastikan bahwa aplikasi tidak mengalami crash akibat input yang tidak valid.

Kekurangan:

1. Scalability: Jika fitur atau atribut tambahan diperlukan di masa depan, kelas ini mungkin perlu diperbarui. Namun, ini adalah trade-off standar dalam desain berbasis objek.
2. Keamanan Input: Meskipun validasi format dasar dilakukan, penanganan input yang lebih canggih dapat dibutuhkan untuk mencegah kesalahan input yang lebih kompleks atau berbahaya.
3. Format Input yang Terbatas: Saat ini, metode hanya menangani titik sebagai pemisah ribuan. Variasi lain, seperti koma sebagai pemisah desimal, tidak ditangani.
4. Keterbatasan Validasi: Validasi dasar dilakukan dengan pengecualian untuk kesalahan format, tetapi mungkin tidak mencakup semua skenario kesalahan atau input tidak valid.

Penyusunan Algoritma dan Kode Program

```
import java.util.Scanner;
import java.text.DecimalFormat;
import java.text.ParseException;

public class BiayaKuliah {

  // Kelas untuk menyimpan informasi mahasiswa
  static class Mahasiswa {
    String nama;
    double uktPerSemester;
    double sppPerSemester;

    Mahasiswa(String nama, double uktPerSemester, double sppPerSemester) {
      this.nama = nama;
      this.uktPerSemester = uktPerSemester;
      this.sppPerSemester = sppPerSemester;
    }

    double hitungTotalBiaya() {
      // Total biaya untuk 4 tahun (8 semester)
      return (uktPerSemester + sppPerSemester) * 8;
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    DecimalFormat df = new DecimalFormat("#,###.00"); // Format angka desimal dengan
pemisah ribuan

    // Input jumlah anggota kelompok
    System.out.print("Masukkan jumlah anggota kelompok: ");
    int jumlahAnggota = scanner.nextInt();
    scanner.nextLine(); // Membersihkan newline

    Mahasiswa[] mahasiswaArray = new Mahasiswa[jumlahAnggota];

    // Input data untuk setiap mahasiswa
    for (int i = 0; i < jumlahAnggota; i++) {
      System.out.println("Data mahasiswa ke-" + (i + 1) + ":");

      System.out.print("Nama: ");
      String nama = scanner.nextLine();
```

```
        double uktPerSemester = 0.0;
        double sppPerSemester = 0.0;

        // Input dan parsing biaya UKT per semester
        while (true) {
            System.out.print("Biaya UKT per semester: ");
            String uktInput = scanner.nextLine();
            try {
                uktPerSemester = parseRupiah(uktInput);
                break;
            } catch (NumberFormatException e) {
                System.out.println("Format tidak valid. Silakan coba lagi.");
            }
        }

        // Input dan parsing biaya SPP per semester
        while (true) {
            System.out.print("Biaya SPP per semester: ");
            String sppInput = scanner.nextLine();
            try {
                sppPerSemester = parseRupiah(sppInput);
                break;
            } catch (NumberFormatException e) {
                System.out.println("Format tidak valid. Silakan coba lagi.");
            }
        }

        mahasiswaArray[i] = new Mahasiswa(nama, uktPerSemester, sppPerSemester);
    }

    // Menampilkan total biaya untuk setiap mahasiswa
    System.out.println("\nTotal biaya kuliah selama 4 tahun untuk setiap anggota kelompok:");
    for (Mahasiswa mahasiswa : mahasiswaArray) {
        double totalBiaya = mahasiswa.hitungTotalBiaya();
        System.out.printf("Nama: %s, Total Biaya: %s\n", mahasiswa.nama,
df.format(totalBiaya));
    }

    scanner.close();
  }

  // Metode untuk parsing input dengan format pemisah ribuan
  private static double parseRupiah(String input) throws NumberFormatException {
    // Menghapus titik sebagai pemisah ribuan
    String sanitizedInput = input.replace(".", "");
    // Mengubah string yang sudah dibersihkan menjadi double
    return Double.parseDouble(sanitizedInput);
  }
}
```

Kode program 2:

```
import java.text.DecimalFormat;
import java.util.Scanner;
```

```
public class TugasKelompok {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      // Input data
      System.out.print("Masukkan UKT per tahun: ");
      double ukt = scanner.nextDouble();

      System.out.print("Masukkan SPP per tahun: ");
      double spp = scanner.nextDouble();

      System.out.print("Masukkan lama studi (4 atau 5 tahun): ");
      int lamaStudi = scanner.nextInt();

      // Hitung biaya total menggunakan operator ternary
      double totalBiaya = (lamaStudi == 4)
         ? (ukt + spp) * 4
         : (ukt + spp) * 5;

      // Format output dengan DecimalFormat
      DecimalFormat df = new DecimalFormat("#,###");
      String formattedBiaya = df.format(totalBiaya);

      // Output hasil
      System.out.println("Jumlah yang harus dibayar selama " + lamaStudi + " tahun adalah: Rp "
+ formattedBiaya);

      scanner.close();
   }
}
```

Kesimpulan

kesimpulannya adalah Secara keseluruhan, program ini memudahkan pengguna untuk menghitung total biaya kuliah berdasarkan lama studi dengan hasil yang disajikan dalam format yang mudah dibaca.

Refleksi: kita dapat memhami bahwa program ini merupakan contoh sederhana dari aplikasi yang menggabungkan konsep dasar pemrograman Java seperti input/output, operator ternary untuk logika, dan format angka. Dalam aplikasi ini, pengguna diminta untuk memasukkan biaya UKT, SPP, dan lama studi, lalu program menghitung total biaya yang harus dibayar berdasarkan input tersebut.