

LAPORAN PRAKTIKUM WEEK 7
PEMROGRAMAN BERORIENTASI OBJEK
“SISTEM MANAJEMEN NILAI MAHASISWA”



Oleh
Benony Gabriel
105222002

PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN ILMU KOMPUTER
UNIVERSITAS PERTAMINA
2024

1. Pendahuluan

1.1 Studi Kasus : Sistem Manajemen Nilai Mahasiswa

Buatlah sebuah program untuk menghitung nilai rata-rata mahasiswa dari setiap kelas dan seluruh nilai rata-rata mahasiswa dari semua kelas dengan ketentuan sebagai berikut:

1.1.1 Class Mahasiswa:

- Terdapat 3 field yaitu nim, nama, dan nilai;
- Terdapat konsep enkapsulasi pada class ini;
- User dapat membuat objek dari kelas ini dengan memasukkan 2 jenis parameter, yang pertama nim, nama, dan nilai sekaligus. Atau bisa hanya dengan nim dan nama dengan isi default field nilai dengan 0.

1.1.2 Class Kelas:

- Terdapat 4 field pada class Kelas, yaitu namaKelas, daftarMahasiswa yang bertipe arrayList dari class Mahasiswa, kemudian jumlahMahasiswa yang bertipe static, dan mhs yang bertipe class Mahasiswa;
- Pada class Kelas terdapat constructor yang memiliki jumlah parameter 1, yaitu namaKelas. Pada constructor ini terdapat inisiasi object dari arrayList daftarMahasiswa
- Terdapat fungsi tambah mahasiswa ke dalam kelas
- Terdapat fungsi hapus mahasiswa dalam kelas
- Terdapat fungsi hitung rata-rata untuk menghitung rata-rata nilai mahasiswa dalam kelas tersebut
- Menerapkan konsep enkapsulasi pada field namaKelas, daftarMahasiswa, dan jumlahMahasiswa

1.1.3 Class Main

- Buatlah beberapa object mahasiswa (minimal 10)
- Buatlah beberapa object kelas (minimal 3)
- Bagi object mahasiswa yang telah kamu buat ke dalam masing-masing kelas (bebas pembagiannya)
- Inisiasi arrayList dengan nama daftarKelas untuk mengumpulkan kelas-kelas tersebut ke dalam satu wadah
- Masukkan semua object kelas yang telah dibuat ke dalam variabel daftarKelas
- Terdapat fungsi hitungRataRataSemuaKelas untuk menghitung rata-rata nilai mahasiswa dari semua kelas
- Terdapat fungsi displatRataRata untuk menampilkan nama kelas dengan nilai rata-ratanya

- Terdapat fungsi `displayRataRataSemuaKelas` untuk menampilkan jumlah mahasiswa dari seluruh kelas dan rata-rata nilai mahasiswa dari seluruh kelas.

2. Variabel dan Tipe Data

Berikut ini adalah deskripsi singkat mengenai variabel-variabel yang dibuat di setiap kelas:

2.1 Class Mahasiswa

Berikut ini adalah deskripsi singkat mengenai variabel-variabel yang dibuat di dalam kelas Mahasiswa.

Nama Variabel	Tipe Data	Keterangan
nim	String	Untuk menyimpan nomor induk mahasiswa
nama	String	Untuk menyimpan nama mahasiswa
nilai	Double	Untuk menyimpan nilai Mahasiswa

Dari ketiga variabel tersebut, semuanya memiliki akses modifier `private`, yang artinya variabel tersebut hanya dapat diakses oleh kelas itu sendiri dan jika mengaksesnya dari kelas lain maka menggunakan setter dan getter.

2.2 Class Kelas

Didalam class Kelas terdapat tiga buah variabel yaitu `namaKelas`, `daftarMahasiswa` dan `jumlahMahasiswa`. Berikut ini adalah deskripsi singkat mengenai variabel-variabel tersebut.

Nama Variabel	Tipe Data	Keterangan
namaKelas	String	Untuk menyimpan nama dari setiap kelas
daftarMahasiswa	<code>ArrayList<Mahasiswa></code>	Untuk menyimpan daftar Mahasiswa dari setiap kelas
jumlahMahasiswa	Static int	Untuk menyimpan total Mahasiswa

Variabel `daftarMahasiswa` bertipe `ArrayList` bertujuan untuk penyimpanan daftar Mahasiswa dari setiap kelas. `ArrayList` ini bertipe `Mahasiswa` yang merupakan class `Mahasiswa`.

2.3 Class Main

Kelas ini adalah tempat untuk mengeksekusi program (main program) yang di dalamnya terdapat objek-objek yang telah dibuat dari class Mahasiswa dan class Kelas.

3. Constructor dan Method

Dalam kelas Java, constructor dan method bekerja bersama-sama untuk membuat objek berfungsi sesuai dengan kebutuhan aplikasi. Constructor digunakan untuk inisialisasi awal objek, sementara method digunakan untuk mendefinisikan perilaku objek dan melakukan operasi pada data.

Berikut ini akan dijelaskan setiap constructor dari masing-masing kelas:

3.1 Class Mahasiswa

Nama Method	Jenis	Keterangan
Mahasiswa(String nim, String nama, Double nilai)	Constructor	Untuk menginisialisasi objek dari Mahasiswa
Mahasiswa(String nim, String nama)	Constructor	Untuk menginisialisasi objek dari Mahasiswa
getNIM	Function	Sebagai getter untuk atribut nim
getNama	Function	Sebagai getter untuk attribute nama
getNilai	Function	Sebagai getter untuk attribute nilai
setNilai	Procedur	Sebagai setter untuk aatribute Nilai

3.2 Class Kelas

Nama Method	Jenis	Keterangan
Kelas(String namaKelas)	Constructor	Untuk menginisialisasi objek dari Kelas
tambahMahasiswa(Mahasiswa mahasiswa)	Prosedur	Untuk menambahkan mahasiswa ke kelas matakuliah

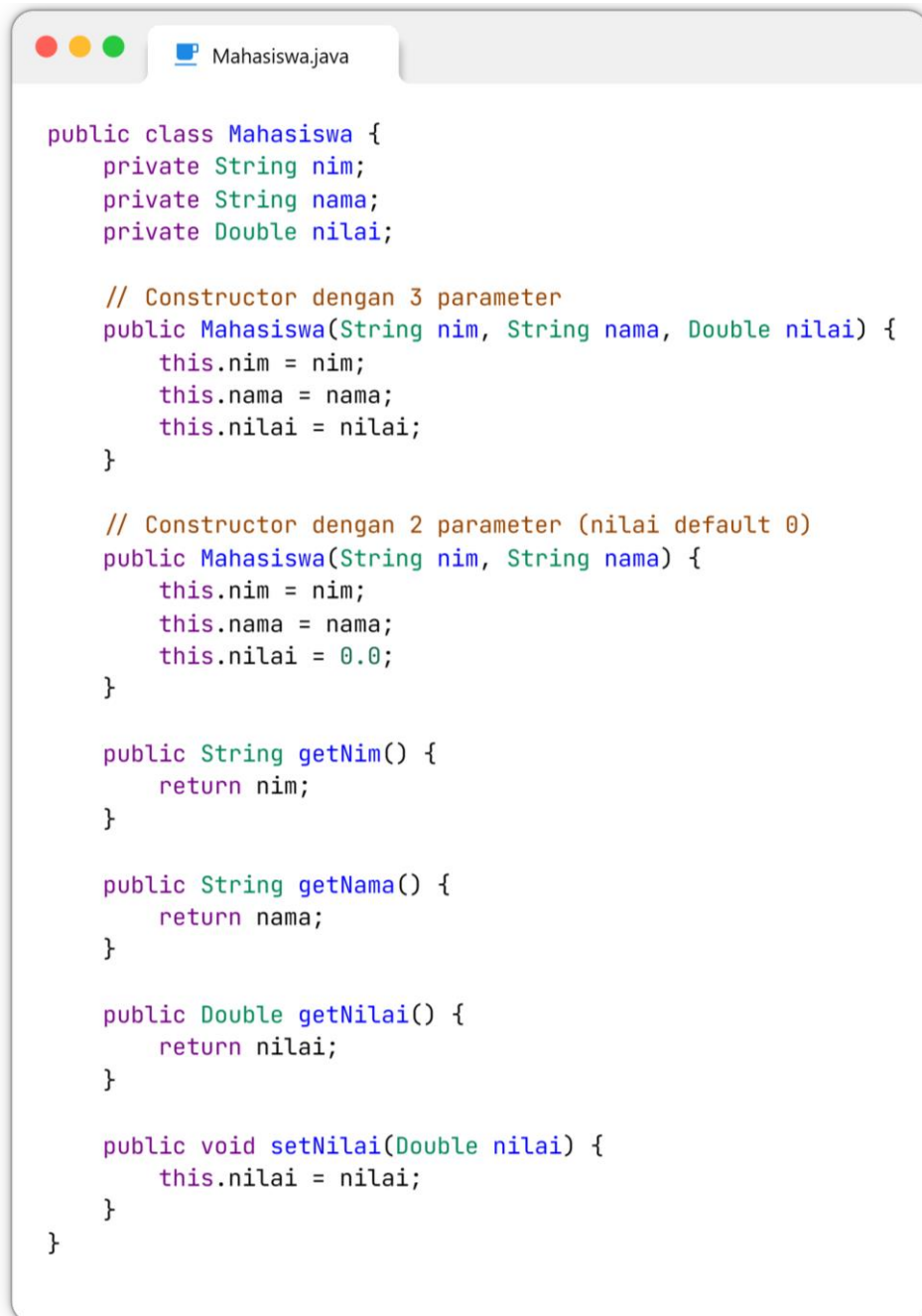
hapusMahasiswa(String nim)	Prosedur	Fungsi untuk menghapus mahasiswa dari kelas
hitungRataRata()	Function	Fungsi hitung rata-rata nilai mahasiswa dalam kelas
getNamaKelas()	Function	Getter untuk atribut namaKelas
getDaftarMahasiswa()	Function	Getter untuk attribute daftarMahasiswa
getJumlahMahasiswa()	Function	Untuk mendapatkan jumlah mahasiswa

3.3 Class Main

Nama Method	Jenis	Keterangan
hitungRataRataSemuaKelas(ArrayList<Kelas> daftarKelas)	Prosedure	Untuk menghitung nilai rata-rata dari semua kelas
displayRataRataSemuaKelas(ArrayList<Kelas> daftarKelas)	Prosedur	Untuk menampilkan rata-rata semua kelas

4. Dokumentasi dan Pembahasan Code

4.1 Class Mahasiswa



```
public class Mahasiswa {
    private String nim;
    private String nama;
    private Double nilai;

    // Constructor dengan 3 parameter
    public Mahasiswa(String nim, String nama, Double nilai) {
        this.nim = nim;
        this.nama = nama;
        this.nilai = nilai;
    }

    // Constructor dengan 2 parameter (nilai default 0)
    public Mahasiswa(String nim, String nama) {
        this.nim = nim;
        this.nama = nama;
        this.nilai = 0.0;
    }

    public String getNim() {
        return nim;
    }

    public String getNama() {
        return nama;
    }

    public Double getNilai() {
        return nilai;
    }

    public void setNilai(Double nilai) {
        this.nilai = nilai;
    }
}
```

Kode di atas merupakan implementasi dari kelas `Mahasiswa`. Berikut adalah penjelasan dari kode tersebut:

- Field (atribut):
 - `private String nim`: Menyimpan NIM (Nomor Induk Mahasiswa) dari mahasiswa.

- `private String nama`: Menyimpan nama dari mahasiswa.
 - `private Double nilai`: Menyimpan nilai dari mahasiswa.
- Constructor:
 - `public Mahasiswa(String nim, String nama, Double nilai)`: Constructor ini menerima tiga parameter yaitu NIM, nama, dan nilai mahasiswa. Ketika objek `Mahasiswa` dibuat dengan menggunakan constructor ini, nilai-nilai parameter tersebut akan digunakan untuk menginisialisasi nilai dari atribut `nim`, `nama`, dan `nilai`.
 - `public Mahasiswa(String nim, String nama)`: Constructor ini menerima dua parameter yaitu NIM dan nama mahasiswa. Constructor ini digunakan ketika mahasiswa tidak memiliki nilai yang ditentukan secara eksplisit. Pada constructor ini, nilai atribut `nilai` diinisialisasi dengan nilai default yaitu 0.0.
 - Getter dan Setter:
 - `getNim()`: Method ini mengembalikan nilai atribut `nim`.
 - `getNama()`: Method ini mengembalikan nilai atribut `nama`.
 - `getNilai()`: Method ini mengembalikan nilai atribut `nilai`.
 - `setNilai(Double nilai)`: Method ini digunakan untuk mengatur nilai atribut `nilai` dengan nilai baru yang diberikan sebagai parameter.

Dengan demikian, kelas `Mahasiswa` memiliki kemampuan untuk menyimpan informasi NIM, nama, dan nilai mahasiswa, serta memberikan akses untuk mengambil dan mengatur nilai-nilai tersebut.

4.2 Class Kelas

```
import java.util.ArrayList;

public class Kelas {
    private String namaKelas;
    private ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>();
    private static int jumlahMahasiswa = 0;

    // Constructor dengan 1 parameter
    public Kelas(String namaKelas) {
        this.namaKelas = namaKelas;
    }

    // Fungsi tambah mahasiswa ke dalam kelas
    public void tambahMahasiswa(Mahasiswa mahasiswa) {
        daftarMahasiswa.add(mahasiswa);
        jumlahMahasiswa++;
    }

    // Fungsi hapus mahasiswa dalam kelas (berdasarkan nim)
    public void hapusMahasiswa(String nim) {
        for (int i = 0; i < daftarMahasiswa.size(); i++) {
            if (daftarMahasiswa.get(i).getNim().equals(nim)) {
                daftarMahasiswa.remove(i);
                jumlahMahasiswa--;
                break;
            }
        }
    }

    // Fungsi hitung rata-rata nilai mahasiswa dalam kelas
    public double hitungRataRata() {
        if (daftarMahasiswa.isEmpty()) return 0;
        int total = 0;
        for (Mahasiswa mhs : daftarMahasiswa) {
            total += mhs.getNilai();
        }
        return (double) total / daftarMahasiswa.size();
    }

    public String getNamaKelas() {
        return namaKelas;
    }

    public ArrayList<Mahasiswa> getDaftarMahasiswa() {
        return daftarMahasiswa;
    }

    public static int getJumlahMahasiswa() {
        return jumlahMahasiswa;
    }
}
```


Kode di atas merupakan implementasi dari kelas `Kelas`, yang digunakan untuk merepresentasikan sebuah kelas yang terdiri dari sejumlah mahasiswa. Berikut adalah penjelasan dari kode tersebut:

- Field (atribut):
 - `private String namaKelas`: Menyimpan nama dari kelas.
 - `private ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>();`: Menyimpan daftar mahasiswa yang terdaftar dalam kelas tersebut menggunakan struktur data ArrayList.
 - `private static int jumlahMahasiswa = 0;`: Menyimpan jumlah total mahasiswa dalam kelas secara keseluruhan. Atribut ini bersifat static sehingga nilainya akan bersifat global dan terbagi di antara semua objek `Kelas`.
- Constructor:
 - `public Kelas(String namaKelas)`: Constructor ini menerima satu parameter yaitu namaKelas dan digunakan untuk menginisialisasi atribut `namaKelas`.
- Method tambahMahasiswa:
 - `public void tambahMahasiswa(Mahasiswa mahasiswa)`: Method ini digunakan untuk menambahkan objek mahasiswa ke dalam daftarMahasiswa. Saat objek mahasiswa ditambahkan, jumlahMahasiswa akan bertambah satu.
- Method hapusMahasiswa:
 - `public void hapusMahasiswa(String nim)`: Method ini digunakan untuk menghapus objek mahasiswa dari daftarMahasiswa berdasarkan NIM (Nomor Induk Mahasiswa). Method ini akan mencari mahasiswa dengan NIM yang sesuai dan menghapusnya dari daftarMahasiswa. Saat objek mahasiswa dihapus, jumlahMahasiswa akan berkurang satu.
- Method hitungRataRata:
 - `public double hitungRataRata()`: Method ini digunakan untuk menghitung rata-rata nilai mahasiswa dalam kelas. Method ini akan menjumlahkan nilai-nilai mahasiswa dalam daftarMahasiswa dan kemudian membaginya dengan jumlah mahasiswa dalam kelas. Jika daftarMahasiswa kosong, method ini akan mengembalikan nilai 0.
- Getter:
 - `public String getNamaKelas()`: Method ini mengembalikan namaKelas.
 - `public ArrayList<Mahasiswa> getDaftarMahasiswa()`: Method ini mengembalikan daftarMahasiswa.

- `public static int getJumlahMahasiswa()`: Method ini mengembalikan jumlahMahasiswa.

Dengan demikian, kelas `Kelas` memiliki kemampuan untuk menambahkan, menghapus, dan menghitung rata-rata nilai mahasiswa dalam kelas.

4.3 Class Main

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {

        // Buat objek mahasiswa
        Mahasiswa mhs1 = new Mahasiswa("105222001", "John Doe", 85.5);
        Mahasiswa mhs2 = new Mahasiswa("105222002", "Erick", 90.0);
        Mahasiswa mhs3 = new Mahasiswa("105222003", "Alice", 75.8);
        Mahasiswa mhs4 = new Mahasiswa("105222004", "Bob", 80.5);
        Mahasiswa mhs5 = new Mahasiswa("105222005", "Charlie", 95.0);
        Mahasiswa mhs6 = new Mahasiswa("105222006", "David", 85.8);
        Mahasiswa mhs7 = new Mahasiswa("105222007", "Emma", 90.5);
        Mahasiswa mhs8 = new Mahasiswa("105222008", "Frank", 75.0);
        Mahasiswa mhs9 = new Mahasiswa("105222009", "Grace", 80.8);
        Mahasiswa mhs10 = new Mahasiswa("10522210", "Hannah", 95.5);

        // Buat objek kelas
        Kelas kelasPBO = new Kelas("Kelas PBO");
        Kelas kelasMetnum = new Kelas("Kelas Metnum");
        Kelas kelasRPL = new Kelas("Kelas RPL");

        // Bagi objek mahasiswa ke dalam masing-masing kelas
        kelasPBO.tambahMahasiswa(mhs1);
        kelasPBO.tambahMahasiswa(mhs2);
        kelasPBO.tambahMahasiswa(mhs3);
        kelasMetnum.tambahMahasiswa(mhs4);
        kelasMetnum.tambahMahasiswa(mhs5);
        kelasMetnum.tambahMahasiswa(mhs6);
        kelasRPL.tambahMahasiswa(mhs7);
        kelasRPL.tambahMahasiswa(mhs8);
        kelasRPL.tambahMahasiswa(mhs9);
        kelasRPL.tambahMahasiswa(mhs10);

        // Inisiasi arrayList dengan nama daftarKelas
        ArrayList<Kelas> daftarKelas = new ArrayList<>();

        // Masukkan semua objek kelas ke dalam variabel daftarKelas
        daftarKelas.add(kelasPBO);
        daftarKelas.add(kelasMetnum);
        daftarKelas.add(kelasRPL);

        // Panggil fungsi hitungRataRataSemuaKelas
        hitungRataRataSemuaKelas(daftarKelas);

        // Panggil fungsi displayRataRataSemuaKelas
        displayRataRataSemuaKelas(daftarKelas);
    }

    // Fungsi hitung rata-rata nilai mahasiswa dari semua kelas
    public static void hitungRataRataSemuaKelas(ArrayList<Kelas> daftarKelas) {
        double totalRataRata = 0;
        for (Kelas kelas : daftarKelas) {
            double rataRataKelas = kelas.hitungRataRata();
            System.out.println("Rata-rata nilai di " + kelas.getNamaKelas() + ": " + rataRataKelas);
            totalRataRata += rataRataKelas;
        }
        System.out.println("Rata-rata nilai semua kelas: " + (totalRataRata / daftarKelas.size()));
    }

    // Fungsi display rata-rata nilai dari semua kelas
    public static void displayRataRataSemuaKelas(ArrayList<Kelas> daftarKelas) {
        int totalMahasiswa = 0;
        for (Kelas kelas : daftarKelas) {
            System.out.println("Jumlah mahasiswa di " + kelas.getNamaKelas() + ": " + kelas.getDaftarMahasiswa().size());
            totalMahasiswa += kelas.getDaftarMahasiswa().size();
        }
        System.out.println("Jumlah mahasiswa dari semua kelas: " + totalMahasiswa);
    }
}
```

Kode di atas merupakan implementasi dari sebuah program untuk mengelola nilai mahasiswa dalam beberapa kelas. Berikut penjelasan singkat dari kode tersebut:

- Main Class ('Main'):
 - Dalam metode 'main', beberapa objek 'Mahasiswa' dan objek 'Kelas' dibuat.
 - Objek mahasiswa kemudian ditambahkan ke dalam kelas yang sesuai menggunakan metode 'tambahMahasiswa'.
 - Daftar kelas disimpan dalam ArrayList 'daftarKelas'.
 - Fungsi 'hitungRataRataSemuaKelas' dipanggil untuk menghitung rata-rata nilai dari semua kelas.
 - Fungsi 'displayRataRataSemuaKelas' dipanggil untuk menampilkan jumlah mahasiswa dari semua kelas.

- Method 'hitungRataRataSemuaKelas':
 - Method ini menghitung rata-rata nilai mahasiswa dari semua kelas.
 - Untuk setiap kelas dalam 'daftarKelas', rata-rata nilai kelas dihitung menggunakan method 'hitungRataRata' dari kelas 'Kelas'.
 - Selama menghitung rata-rata untuk setiap kelas, nilai rata-rata tiap kelas dan total nilai rata-rata semua kelas juga ditampilkan.

- Method 'displayRataRataSemuaKelas':
 - Method ini menampilkan jumlah mahasiswa dari semua kelas.
 - Untuk setiap kelas dalam 'daftarKelas', jumlah mahasiswa dihitung menggunakan method 'size()' dari ArrayList 'daftarMahasiswa' pada kelas 'Kelas'.
 - Selama menghitung jumlah mahasiswa untuk setiap kelas, jumlah mahasiswa tiap kelas dan total jumlah mahasiswa dari semua kelas juga ditampilkan.

Kode ini secara keseluruhan menggambarkan bagaimana kita dapat membuat objek mahasiswa dan kelas, menambahkan mahasiswa ke dalam kelas, menghitung rata-rata nilai dari setiap kelas, dan menampilkan informasi tentang jumlah mahasiswa dari semua kelas.

5. Kesimpulan

Program ini memiliki struktur yang terorganisir dengan baik, menggunakan enkapsulasi untuk melindungi atribut dan memungkinkan akses terkontrol, serta menggunakan konstruktor untuk inisialisasi objek dengan parameter default atau kustom. Penggunaan ArrayList mempermudah penyimpanan dan manajemen objek, sedangkan penggunaan variabel statis membantu dalam melacak informasi global. Program tersebut juga mengikuti prinsip pemisahan fungsi dengan memisahkan tugas-tugas seperti menghitung rata-rata nilai dan menampilkan informasi jumlah mahasiswa ke dalam fungsi-fungsi terpisah. Secara keseluruhan, program tersebut memberikan contoh bagaimana prinsip-prinsip dasar OOP dapat diterapkan untuk menciptakan program yang terstruktur, mudah dimengerti, dan mudah dikelola.