**LAPORAN PRAKTIKUM WEEK 7**

**PEMROGRAMAN BERORIENTASI OBJEK**

**“SISTEM MANAJEMEN NILAI MAHASISWA”**

**A logo with a person in the center of leaves

Description automatically generated**

**Oleh**

**Benony Gabriel**

**105222002**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN ILMU KOMPUTER**

**UNIVERSITAS PERTAMINA**

**2024**

1. **Pendahuluan**
   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. 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. 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. 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.

1. **Variabel dan Tipe Data**

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

* 1. Class Mahasiswa

Berikut ini adalah deskripsi singkat mengenai variabel-variable 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.

* 1. Class Kelas

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

|  |  |  |
| --- | --- | --- |
| **Nama Variabel** | **Tipe Data** | **Keterangan** |
| namaKelas | String | Untuk menyimpan nama dari setiap kelas |
| daftarMahasiswa | ArrayList<Mahasiswa> | 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.

* 1. 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.

1. **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:

* 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 atribute nama |
| getNilai | Function | Sebagai getter untuk atribute nilai |
| setNilai | Procedur | Sebagai setter untuk aatribute Nilai |

* 1. 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 atribute daftarMahasiswa |
| getJumlahMahasiswa() | Function | Untuk mendapatkan jumlah mahasiswa |

* 1. 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 |

1. **Dokumentasi dan Pembahasan Code**
   1. Class Mahasiswa

A screenshot of a computer

Description automatically generated

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.

* 1. Class Kelas

A screenshot of a cell phone

Description automatically generated

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.

* 1. Class Main

A screen shot of a computer

Description automatically generated

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.

1. **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.