**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**INHERITANCE**

Logo

Description automatically generated with medium confidence

**Disusun Oleh:**

**Rayhan Surya Destian (105222024)**

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS SAINS DAN KOMPUTER**

**PEMROGRAMAN BERORIENTASI OBJEK**

**UNIVERSITAS PERTAMINA**

**2023/2024**

1. **Pendahuluan**

Laporan ini menjelaskan struktur dan fungsi dari program Java yang dirancang untuk menghitung nilai rata-rata mahasiswa dalam setiap kelas dan keseluruhan nilai rata-rata dari semua kelas.

1. **Variabel**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Variabel | Tipe data | Fungsi |
| 1 | nim | String | Menyimpan nomor induk mahasiswa |
| 2 | nama | String | Menyimpan nama mahasiswa |
| 3 | nilai | double | Menyimpan nilai mahasiswa |
| 4 | namaKelas | String | Menyimpan nama kelas |
| 5 | daftarMahasiswa | ArrayList<Mahasiswa> | Menyimpan daftar mahasiswa dalam kelas |
| 6 | jumlahMahasiswa | int (static) | Menyimpan jumlah total mahasiswa di semua kelas |
| 7 | daftarKelas | ArrayList<Kelas> | Menyimpan daftar semua kelas yang terdaftar dalam sistem |

1. **Constructor dan Method**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Nama Metode | Jenis Metode | Fungsi |
| 1 | Mahasiswa(nim, nama, nilai) | Constructor | Inisialisasi objek Mahasiswa dengan nilai spesifik |
| 2 | Mahasiswa(nim, nama) | Constructor | Inisialisasi objek Mahasiswa dengan nilai default |
| 3 | Kelas(namaKelas) | Constructor | Inisialisasi objek Kelas dan array daftarMahasiswa |
| 4 | tambahMahasiswa(m) | Procedural | Menambahkan objek Mahasiswa ke dalam daftarMahasiswa |
| 5 | hapusMahasiswa(m) | Procedural | Menghapus objek Mahasiswa dari daftarMahasiswa |
| 6 | hitungRataRata() | Functional | Menghitung dan mengembalikan nilai rata-rata dari kelas |
| 7 | displayRataRata(daftarKelas) | Procedural | Menampilkan rata-rata nilai untuk setiap kelas |
| 8 | displayRataRataSemuaKelas(daftarKelas) | Procedural | Menampilkan jumlah mahasiswa dan rata-rata nilai seluruh kelas |

1. **Dokumentasi dan Pembahasan Code**

Program yang dibuat terdiri dari tiga kelas utama: `Mahasiswa`, `Kelas`, dan `Main`. Setiap kelas berfungsi untuk mengorganisasi dan memanipulasi data terkait mahasiswa dan kelas mereka dengan cara yang efisien dan terstruktur.

**Mahasiswa**

Kelas `Mahasiswa` mendefinisikan objek mahasiswa dengan atribut `nim`, `nama`, dan `nilai`. Terdapat dua konstruktor, yang pertama menerima semua atribut, sedangkan yang kedua hanya menerima `nim` dan `nama`, dengan `nilai` diinisialisasi ke 0. Ini memungkinkan fleksibilitas dalam penciptaan objek Mahasiswa tergantung pada ketersediaan data nilai.

**Kelas**

Kelas `Kelas` bertanggung jawab untuk mengelola grup mahasiswa. Atributnya termasuk `namaKelas` dan `daftarMahasiswa`, sebuah ArrayList yang menyimpan objek-objek Mahasiswa. Metode `tambahMahasiswa` dan `hapusMahasiswa` memodifikasi daftar ini. Metode `hitungRataRata` mengkalkulasi rata-rata nilai mahasiswa dalam kelas, yang merupakan contoh praktis dari penggunaan koleksi dan loop.

**Main**

Kelas `Main` mengandung metode `main`, di mana objek-objek Mahasiswa dan Kelas diciptakan dan dimanipulasi. Data disusun ke dalam ArrayList `daftarKelas`, dan fungsi `displayRataRata` serta `displayRataRataSemuaKelas` digunakan untuk menampilkan informasi. Ini menunjukkan implementasi dari konsep seperti enkapsulasi dan manajemen koleksi dalam konteks Java yang lebih besar.

1. **Kesimpulan**

Program yang dikembangkan menunjukkan efektivitas konsep Pemrograman Berorientasi Objek (OOP) dalam Java untuk mengelola data mahasiswa. Dengan menggunakan enkapsulasi, program ini memastikan keamanan data dan menyederhanakan akses melalui metode yang terdefinisi dengan baik.

1. **Daftar Pustaka**

-