Hamzah Rizqullah Rahmad **G1F024067** **Anggita afriyani** **G1FO24O11** **Rizqi Nadhifah** **G1F024017**	**Tugas kelompok**	**18 september 2024**

[No. 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

(a) Analisa atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

(b) Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

(c) Rekomendasi atribut, method, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!

(d) Desain kode program Java yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

[No.1] Analisis dan Argumentasi

A. Saya mengusulkan permasalahan ini dapat diatasi dengan cara
Menambahkan metode untuk menampilkan informasi yang lebih rinci tentang siswa.
b. Alasan solusi ini karena
Dengan menambahkan metode untuk menampilkan informasi, kita dapat memastikan bahwa data yang ditampilkan sudah sesuai dengan kebutuhan dan permintaan data.
C. Perbaikan kode program dengan cara
Membuat metode baru untuk menampilkan informasi yang lebih rinci tentang siswa.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma
   Algoritma adalah langkah-langkah penyelesaian masalah.
   (a) Mulai
   (b) Deklarasi KelasMahasiswa :
   (c) Deklarasikan atribut
   (d) Buat konstruktor untuk menginisialisasi atribut.
   (e) Buat pengambil metode untuk setiap atribut.
   (f) Buat metode untuk menampilkan informasi siswa.
   (g) Deklarasi KelasMain
   (h) Buat metode dan objek dengan data yang berbeda.
   (i) Selesai

2) Kode program dan luaran

```java
class Mahasiswa {
    private String nama;
    private String npm;
    private String jurusan;
    private int tahunLulus;

    public Mahasiswa(String nama, String npm, String jurusan, int tahunLulus) {
        this.nama = nama;
        this.npm = npm;
        this.jurusan = jurusan;
        this.tahunLulus = tahunLulus;
    }

    public String getNama() {
        return nama;
    }

    public String getNPM() {
        return npm;
    }

    public String getJurusan() {
        return jurusan;
    }

    public int getTahunLulus() {
        return tahunLulus;
    }

    public void tampilkanInfo() {
        System.out.println("Nama: " + nama);
        System.out.println("NPM: " + npm);
        System.out.println("Jurusan: " + jurusan);
        System.out.println("Tahun Lulus: " + tahunLulus);
        System.out.println();
    }
}

public class Main {
    public static void main(String[] args) {
        Mahasiswa m1 = new Mahasiswa("Anggita Afriyani", "G1FO24011", "Sistem Informasi", 2028);
        Mahasiswa m2 = new Mahasiswa("Hamzah Risqullah Rahmad", "G1FO24067", "Sistem Informasi", 2028);
        Mahasiswa m3 = new Mahasiswa("Rizqi Nadhifah Setiani", "G1FO24017", "Sistem Informasi", 2028);

        m1.tampilkanInfo();
        m2.tampilkanInfo();
        m3.tampilkanInfo();
    }
```

Gambar 1 (input)

Output Generated Files

```
Nama: Anggita Afriyani
NPM: G1FO24011
Jurusan: Sistem Informasi
Tahun Lulus: 2028

Nama: Hamzah Risqullah Rahmad
NPM: G1FO24067
Jurusan: Sistem Informasi
Tahun Lulus: 2028

Nama: Rizqi Nadhifah Setiani
NPM: G1FO24017
Jurusan: Sistem Informasi
Tahun Lulus: 2028
```

Gambar 1 (output)

a) Screenshot/ Capture potongan kode dan hasil luaran

b) Analisa luaran yang dihasilkan

Analisa kode program mendefinisikan dua kelas dalam bahasa pemrograman Java, yaitu Mahasiswa dan Main. Kelas Mahasiswa memiliki beberapa atribut privat: nama, npm, jurusan, dan tahunLulus, yang merepresentasikan data mahasiswa. Konstruktornya

digunakan untuk menginisialisasi nilai dari atribut-atribut tersebut. Selain itu, ada beberapa method getter untuk mengambil nilai dari setiap atribut, serta method tampilkanInfo() yang digunakan untuk mencetak informasi mahasiswa ke konsol.

Pada kelas Main, metode main membuat tiga objek Mahasiswa: m1, m2, dan m3, yang masing-masing merepresentasikan mahasiswa dengan nama, NPM, jurusan, dan tahun lulus yang berbeda. Objek-objek ini kemudian memanggil metode tampilkanInfo() untuk menampilkan detail masing-masing mahasiswa. Keluaran dari program ini adalah informasi detail dari ketiga mahasiswa yang ditampilkan satu per satu di konsol, dengan format: "Nama", "NPM", "Jurusan", dan "Tahun Lulus". Program berjalan tanpa interaksi pengguna, hanya menampilkan hasil yang telah diinisialisasi dalam kode.

[No.1] Kesimpulan

1) Analisa

a) Kesimpulan berdasarkan permasalahan, algoritma, dan kode program menunjukkan bahwa penambahan metode baru untuk menampilkan informasi yang lebih rinci sangat penting untuk memastikan bahwa data yang ditampilkan sudah sesuai dengan kebutuhan dan permintaan data.
b) Dasar alasan pengambilan keputusan saya untuk kasus ini adalah:
Pada program itu saya menggunakan bentuk kelas public karena diperlukan aksesibilitas dari luar kelas.
Perbaikan program dengan menambahkan metode baru karena struktur Java memerlukan penggunaan enkapsulasi dan polimorfisme untuk melindungi data dan menambahkan perilaku yang berbeda.
Refleksi
Minggu ini, saya belajar banyak tentang konsep dasar pemrograman berorientasi objek (OOP) di Java, termasuk kelas, objek, pewarisan, dan polimorfisme. Saya memahami pentingnya enkapsulasi dalam menjaga integritas data dan mengatasi tantangan dalam penamaan dan implementasi konsep OOP. Pengalaman ini meningkatkan pemahaman saya tentang struktur program Java dan penerapan prinsip OOP dalam pengembangan aplikasi.

Refleksi

Selama minggu ini, kami belajar lebih dalam tentang konsep OOP di Java, khususnya pada pentingnya enkapsulasi dan pewarisan untuk membangun program yang aman dan efisien. Tantangan yang kami hadapi adalah dalam memahami polimorfisme dan penerapannya dalam kode. Pengalaman ini meningkatkan pemahaman kami tentang OOP dan cara menerapkan konsep-konsep ini dalam aplikasi nyata.