Nama & NPM	Topik:	Tanggal:
1. **Abullah Hasyim Syauqi (G1F024019)** 2. **Sindi Putri utami (G1F024053)** 3. **Zaira Ayu Wandira (G1F024055)**	**kelas, Objek, Method**	**12 september 2024**

Identifikasi Masalah:

Apabila diketahui kelas induk adalah Mahasiswa dan Kelas anak adalah turunan dari mahasiswa maka:

- (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)!

Analisis dan Argumentasi

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

Atribut:

Atribut-atribut di kelas **Mahasiswa** umumnya menggambarkan data atau informasi dasar yang dimiliki setiap mahasiswa. Contoh atribut:

- String nama: untuk menyimpan nama mahasiswa.
- int nim: untuk menyimpan Nomor Induk Mahasiswa.
- String jurusan: untuk menyimpan jurusan yang diambil mahasiswa.
- int angkatan: untuk menyimpan tahun angkatan mahasiswa.

Method:

Method di kelas induk **Mahasiswa** bisa berupa fungsi-fungsi yang mengelola data mahasiswa. Beberapa contoh:

- void tampilkanInfo(): Menampilkan informasi mahasiswa (nama, nim, jurusan, angkatan).
- void belajar(): Method yang menggambarkan kegiatan mahasiswa untuk belajar.
- void registrasiKrs(): Untuk merepresentasikan kegiatan mahasiswa melakukan registrasi KRS.

Constructor:

Constructor pada kelas **Mahasiswa** digunakan untuk menginisialisasi nilai awal dari atributatributnya. Misalnya:

```
public Mahasiswa(String nama, int nim, String jurusan, int angkatan) {
```

```
        this.nama = nama;

    this.nim = nim;

    this.jurusan = jurusan;

    this.angkatan = angkatan; }
```

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

Kelas Induk :

- Kelas induk berfungsi sebagai template umum yang mendeskripsikan semua atribut dan perilaku dasar yang dimiliki oleh setiap objek turunan, dalam hal ini seorang mahasiswa.
- Peran utama kelas ini adalah untuk menyediakan data dan operasi dasar yang berlaku secara umum untuk semua jenis mahasiswa, baik S1, S2, maupun S3.
- Memiliki atribut-atribut dasar yang mendefinisikan identitas mahasiswa secara umum, seperti nama, nim, jurusan, dan angkatan.
- Method di kelas induk mengelola operasi dasar yang berlaku untuk semua mahasiswa, seperti tampilkanInfo() dan belajar().
- Constructor kelas induk bertugas menginisialisasi atribut dasar, seperti nama, nim, jurusan, dan angkatan.
- **Kelas Induk (Mahasiswa)** menyediakan kerangka umum yang dapat digunakan untuk berbagai jenis mahasiswa

Kelas anak :

- Kelas anak lebih spesifik dan bertujuan untuk memperluas atau memodifikasi fungsi yang disediakan oleh kelas induk.
- Menambahkan atribut yang lebih **spesifik** untuk mahasiswa S1, seperti judulTugasAkhir dan pembimbing.
- Kelas anak dapat meng-**override** method dari kelas induk. Misalnya, method tampilkanInfo() bisa diperluas untuk juga menampilkan judulTugasAkhir pada mahasiswa S1.
- Constructor kelas anak menginisialisasi atribut tambahan yang lebih spesifik untuk mahasiswa S1, seperti judulTugasAkhir dan pembimbing

(c) Rekomendasi Atribut, Method, dan Constructor Atribut Bersama:

- nama, nim, jurusan, angkatan (dari kelas induk)

Atribut Spesifik untuk Kelas Anak:

- judulTugasAkhir, pembimbing (untuk mahasiswa S1)

Method Bersama:

- void tampilkanInfo()
- void belajar()
- void registrasiKrs()

Constructor Bersama:

- Constructor di kelas induk harus dipanggil dalam constructor kelas anak menggunakan super().

Penyusunan Algoritma dan Kode Program

Desain Kode Program Java:

```
// Kelas Induk
class Mahasiswa {
    protected String nama;
    protected String nim;
    protected String jurusan;
    protected int angkatan;

    // Constructor
    public Mahasiswa(String nama, String nim, String jurusan, int angkatan) {
        this.nama = nama;
        this.nim = nim;
        this.jurusan = jurusan;
        this.angkatan = angkatan;
    }

    // Method untuk menampilkan info
    public void tampilkanInfo() {
        System.out.println("Nama: " + nama);
        System.out.println("NIM: " + nim);
        System.out.println("Jurusan: " + jurusan);
        System.out.println("Angkatan: " + angkatan);
    }

    // Method belajar
    public void belajar() {
        System.out.println(nama + " sedang belajar.");
    }

    // Method registrasi KRS
    public void registrasiKrs() {
        System.out.println(nama + " telah melakukan registrasi KRS.");
    }
}

// Kelas Anak
class MahasiswaS1 extends Mahasiswa {
```

```
    private String judulTugasAkhir; // Atribut tugas akhir
    private String pembimbing;

    // Constructor
    public MahasiswaS1(String nama, String nim, String jurusan, int angkatan) {
        super(nama, nim, jurusan, angkatan); // Memanggil constructor kelas induk
    }

    // Setter untuk judulTugasAkhir dan pembimbing
    public void setJudulTugasAkhir(String judul) {
        this.judulTugasAkhir = judul;
    }

    public void setPembimbing(String pembimbing) {
        this.pembimbing = pembimbing;
    }

    // Override method tampilkanInfo
    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo(); // Panggil method dari kelas induk
        if (judulTugasAkhir != null) {
            System.out.println("Judul Tugas Akhir: " + judulTugasAkhir);
        }
        if (pembimbing != null) {
            System.out.println("Pembimbing: " + pembimbing);
        }
    }
}

// Contoh penggunaan
public class Main {
    public static void main(String[] args) {
        // Membuat objek mahasiswa S1
        MahasiswaS1 mahasiswa1 = new MahasiswaS1("zaira", "G1F024055", "SISTEM
INFORMASI", 2024);
        mahasiswa1.setJudulTugasAkhir("Sistem Informasi");
        mahasiswa1.setPembimbing("endina putri purwandari");

        MahasiswaS1 mahasiswa2 = new MahasiswaS1("hasyim", "G1F024019" , "SISTEM
INFORMASI", 2024);
        mahasiswa2.setJudulTugasAkhir("Robo-tik");
        mahasiswa2.setPembimbing("endina putri purwandari");

        MahasiswaS1 mahasiswa3 = new MahasiswaS1("Sindi","G1F024053" , "SISTEM
INFORMASI", 2023);
        mahasiswa3.setJudulTugasAkhir("Analisis Data");
        mahasiswa3.setPembimbing("endina putri purwandari");

        // Menampilkan info untuk masing-masing mahasiswa
        System.out.println("Informasi Mahasiswa 1:");
        mahasiswa1.tampilkanInfo();
        mahasiswa1.belajar();
```

```
        mahasiswa1.registrasiKrs();
        System.out.println();

        System.out.println("Informasi Mahasiswa 2:");
        mahasiswa2.tampilkanInfo();
        mahasiswa2.belajar();
        mahasiswa2.registrasiKrs();
        System.out.println();

        System.out.println("Informasi Mahasiswa 3:");
        mahasiswa3.tampilkanInfo();
        mahasiswa3.belajar();
        mahasiswa3.registrasiKrs();
    }
}
```

kesimpulan

Kode di atas menunjukkan cara kerja pemrograman berorientasi objek dalam Java. Ada kelas induk bernama Mahasiswa yang menyimpan informasi dasar seperti nama, NIM, jurusan, dan angkatan, serta beberapa metode untuk aktivitas mahasiswa. Kelas anak MahasiswaS1 menambahkan informasi khusus, seperti judul tugas akhir dan pembimbing, dengan menggunakan metode tampilkanInfo yang sudah dimodifikasi.
Dengan penggunaan modifier protected dan setter, kode ini juga menerapkan prinsip encapsulation. Di kelas Main, beberapa objek MahasiswaS1 dibuat dan diisi data, lalu informasi dan aktivitas mereka ditampilkan. Secara keseluruhan, kode ini memberikan contoh yang jelas dan terstruktur tentang pemrograman berorientasi objek di Java.

Refleksi:kami mempelajari banyak hal dalam membuat kelas,objek,dan metode dalam pemrograman dan kami mendapatkan pengalaman belajar yang baik