

Nama & NPM	Topik:	Tanggal:
<b>M. ARIQ HAFIDL FATANAH</b> <b>G1F024014</b> <b>M. HOVID ARMAN</b> <b>G1F024034</b> <b>ARIQ ASHAR SOFYAN</b> <b>G1F024052</b>	<b>KELAS, OBJEK, METHOD</b>	<b>18 SEPTEMBER 2024</b>

**[No. 1] Identifikasi Masalah:**

- 1) Uraikan permasalahan dan variabel
  1. 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)!

**[No.1] Analisis dan Argumentasi**

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan validasi data input dan meningkatkan penjelasan untuk keterbacaan kode.
- 2) Alasan solusi ini karena
  - Validasi input untuk membantu mencegah data tidak valid yang mungkin terjadi di masa depan.
  - Keterbacaan kode untuk membantu meningkatkan pemeliharaan program di masa mendatang ketika perlu memperbarui atau menambah fitur baru.
- 3) Perbaiki kode program dengan cara:
  - Membuat variabel nama, umur, dan NIM pada kelas induk, serta jurusan di kelas turunan.
  - Menggunakan input data dinamis.

**[No.1 ] Penyusunan Algoritma dan Kode Program**

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
- 1). Mulai
- 2). Deklarasikan kelas Mahasiswa dengan atribut nama, umur, dan NIM.
- 3). Deklarasi constructor untuk inisialisasi atribut kelas Mahasiswa.
- 4). Definiskan method tampilkanInfo() untuk menampilkan data mahasiswa.
- 5). Deklarasi kelas turunan MahasiswaBaru dari kelas Mahasiswa dengan atribut tambahan jurusan.
- 6). Override method tampilkanInfo() di kelas MahasiswaBaru untuk menampilkan informasi tambahan.
- 7). Inisialisasi objek MahasiswaBaru dan panggil method tampilkanInfo() untuk menampilkan informasi lengkap mahasiswa.
- 8). Jalankan program

2) Kode program dan luaran

```
//Kelas Induk Mahasiswa
class Mahasiswa {
    String nama;
    int umur;
    String NIM;

    // Constructor
    public Mahasiswa(String nama, int umur, String NIM) {
        this.nama = nama;
        this.umur = umur;
        this.NIM = NIM;
    }

    // Method untuk menampilkan informasi mahasiswa
    public void tampilkanInfo() {
        System.out.println("\nNama : " + nama);
        System.out.println("Umur : " + umur);
        System.out.println("NIM : " + NIM);
    }
}

//Kelas Anak MahasiswaBaru (turunan dari Mahasiswa)
class MahasiswaBaru extends Mahasiswa {
    String jurusan;

    // Constructor
    public MahasiswaBaru(String nama, int umur, String NIM, String jurusan) {
        super(nama, umur, NIM); // Memanggil constructor kelas induk
        this.jurusan = jurusan;
    }

    // Override method tampilkanInfo() untuk menampilkan informasi tambahan
    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo(); // Memanggil method dari kelas induk
        System.out.println("Jurusan: " + jurusan);
    }
}

//Contoh penggunaan kelas
public class Main {
    public static void main(String[] args) {
        MahasiswaBaru Mhs1 = new MahasiswaBaru("MUHAMMAD ARIQ HAFIDL
FATANA", 19, "G1F024014", "SISTEM INFORMASI");
        Mhs1.tampilkanInfo();
        MahasiswaBaru Mhs2 = new MahasiswaBaru("MUHAMMAD HOVID ARMAN", 17,
"G1F024034", "SISTEM INFORMASI");
        Mhs2.tampilkanInfo();
        MahasiswaBaru Mhs3 = new MahasiswaBaru("ARIQ ASHAR SOFYAN", 17,
"G1F024052", "SISTEM INFORMASI");
        Mhs3.tampilkanInfo();
    }
}
```

```
}  
}
```

Luaran :

Nama : MUHAMMAD ARIQ HAFIDL FATANAH

Umur : 19

NIM : G1F024014

Jurusan: SISTEM INFORMASI

Nama : MUHAMMAD HOVID ARMAN

Umur : 17

NIM : G1F024034

Jurusan: SISTEM INFORMASI

Nama : ARIIQ ASHAR SOFYAN

Umur : 17

NIM : G1F024052

Jurusan: SISTEM INFORMASI

- a) Screenshot/ Capture potongan kode dan hasil luaran  
Berikan komentar pada kode yang di Screenshot

The screenshot shows an IDE with a Java file named `Mahasiswa.java`. The code defines a base class `Mahasiswa` with attributes `nama`, `umur`, and `NIM`, and a method `tampilkanInfo()`. It also defines a subclass `MahasiswaBaru` that inherits from `Mahasiswa` and adds a `Jurusan` attribute. A `Main` class is used to create three instances of `MahasiswaBaru` and call their `tampilkanInfo()` methods. The output window on the right shows the results of these method calls, displaying the name, age, NIM, and faculty for each student.

```
1 //Kelas induk Mahasiswa  
2 class Mahasiswa {  
3     String nama;  
4     int umur;  
5     String NIM;  
6  
7     // Constructor  
8     public Mahasiswa(String nama, int umur, String NIM) {  
9         this.nama = nama;  
10        this.umur = umur;  
11        this.NIM = NIM;  
12    }  
13  
14    // Method untuk menampilkan informasi mahasiswa  
15    public void tampilkanInfo() {  
16        System.out.println("\nNama : " + nama);  
17        System.out.println("Umur : " + umur);  
18        System.out.println("NIM : " + NIM);  
19    }  
20 }  
21  
22 //Kelas Anak MahasiswaBaru (turunan dari Mahasiswa)  
23 class MahasiswaBaru extends Mahasiswa {  
24     String Jurusan;  
25  
26     // Constructor  
27     public MahasiswaBaru(String nama, int umur, String NIM, String Jurusan) {  
28         super(nama, umur, NIM); // Memanggil constructor kelas induk  
29         this.Jurusan = Jurusan;  
30     }  
31  
32     // Override method tampilkanInfo() untuk menampilkan informasi tambahan  
33     @Override  
34     public void tampilkanInfo() {  
35         super.tampilkanInfo(); // Memanggil method dari kelas induk  
36         System.out.println("Jurusan: " + Jurusan);  
37     }  
38 }  
39  
40 //Contoh penggunaan kelas  
41 public class Main {  
42     public static void main(String[] args) {  
43         MahasiswaBaru Mhs1 = new MahasiswaBaru("MUHAMMAD ARIQ HAFIDL FATANAH", 19, "G1F024014", "SISTEM INFORMASI");  
44         Mhs1.tampilkanInfo();  
45         MahasiswaBaru Mhs2 = new MahasiswaBaru("MUHAMMAD HOVID ARMAN", 17, "G1F024034", "SISTEM INFORMASI");  
46         Mhs2.tampilkanInfo();  
47         MahasiswaBaru Mhs3 = new MahasiswaBaru("ARIQ ASHAR SOFYAN", 17, "G1F024052", "SISTEM INFORMASI");  
48         Mhs3.tampilkanInfo();  
49     }  
50 }  
51 }
```

Input/Output

Language Version: JDK 21.0.0 Interactive

Input Arguments

Output Generated Files

Nama : MUHAMMAD ARIQ HAFIDL FATANAH  
Umur : 19  
NIM : G1F024014  
Jurusan: SISTEM INFORMASI

Nama : MUHAMMAD HOVID ARMAN  
Umur : 17  
NIM : G1F024034  
Jurusan: SISTEM INFORMASI

Nama : ARIQ ASHAR SOFYAN  
Umur : 17  
NIM : G1F024052  
Jurusan: SISTEM INFORMASI

Compiled and executed in 1.236 sec(s)

- b) Analisa luaran yang dihasilkan  
Luaran menampilkan hasil yang sesuai dengan yang diinginkan pemrogram ketika menyusun program ini, menampilkan nama, umur, dan NIM pada kelas induk, serta jurusan di kelas turunan.

## [No.1] Kesimpulan

### 1) Analisa

1. 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)!...

Jawab:

**Kelas Induk Mahasiswa:**

- Atribut: nama, umur, NIM
- Method: tampilkanInfo()

- Constructor: Menginisialisasi nama, umur, dan NIM

#### **Kelas Anak MahasiswaBaru:**

- Atribut tambahan: jurusan
- Method: Override method tampilkanInfo() untuk menambahkan informasi jurusan
- Constructor: Memanggil constructor kelas induk menggunakan super() dan menambahkan atribut jurusan.

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

Jawab:

- Kelas induk Mahasiswa hanya berfokus pada data umum mahasiswa (nama, umur, NIM).
- Kelas anak MahasiswaBaru menambahkan informasi khusus seperti jurusan, dan meng-overwrite method dari kelas induk untuk menambahkan informasi tersebut.

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

Jawab:

- Atribut yang dapat digunakan bersama: nama, umur, NIM.
- Method yang dapat digunakan bersama: tampilkanInfo() dengan overriding di kelas anak.
- Constructor: Constructor dari kelas induk dapat digunakan kembali dengan menambahkan parameter tambahan di kelas anak (seperti jurusan).

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

Jawab:

- Kode telah dirancang dengan baik menggunakan prinsip pewarisan (inheritance), di mana kelas induk dan kelas anak saling terhubung. Kelas anak menggunakan method dan constructor dari kelas induk serta menambahkan fitur tambahan.

#### **Refleksi**

Pada minggu ini, saya mempelajari lebih dalam tentang konsep pewarisan dalam pemrograman berorientasi objek (OOP) menggunakan Java. Saya mendapatkan pemahaman yang lebih baik tentang cara mendesain kelas induk dan kelas anak, serta bagaimana memanfaatkan konsep overriding untuk memperluas fungsionalitas kelas. Tantangan yang dihadapi adalah dalam memahami bagaimana method dari kelas induk dapat digunakan secara efisien dalam kelas anak tanpa mengulangi kode. Namun, melalui implementasi ini, saya belajar pentingnya keterbacaan dan modularitas kode agar program lebih mudah dikembangkan di masa depan.