

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Muhammad Nabiil Musyaffa' G1F024030</b>	<b>Percabangan: If dan Swtich</b>	<b>7 Oktober 2024</b>
<b>[Nomor Soal] Identifikasi Masalah:</b>		
1) Uraikan permasalahan dan variabel 2) Rincikan sumber informasi yang relevan (buku / webpage) 3) Uraikan rancangan solusi yang diusulkan (jika ada). 4) Analisis susunan solusi, parameter solusi (jika ada).		
<b>[Nomor Soal] Analisis dan Argumentasi</b>		
1) Uraikan rancangan solusi yang diusulkan. 2) Analisis solusi, kaitkan dengan permasalahan.		
<b>[Nomor Soal] Penyusunan Algoritma dan Kode Program</b>		
1) Rancang desain solusi atau algoritma 2) Tuliskan kode program dan luaran <ul style="list-style-type: none"> <li>a) Beri komentar pada kode</li> <li>b) Uraikan luaran yang dihasilkan</li> <li>c) Screenshot/ Capture potongan kode dan hasil luaran</li> </ul>		
<b>[Nomor Soal] Kesimpulan</b>		
1) Analisa <ul style="list-style-type: none"> <li>a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!</li> <li>b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?</li> </ul> 2) Evaluasi <ul style="list-style-type: none"> <li>a) Apa konsekuensi dari skenario pemrograman ini?</li> <li>b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)</li> </ul> 3) Kreasi <ul style="list-style-type: none"> <li>a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?</li> <li>b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)</li> </ul>		

## **LATIHAN 1**

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

Pada Latihan ini kita diperlukan untuk menganalisa kode program yang tepat untuk menganalisa ciri-ciri umum Kelas Manusia yang dapat menjadi atribut variabel, dan perilaku/ behavior untuk method.

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

Dari kode yang diberikan, kita dapat melihat bahwa kelas Manusia memiliki dua atribut: nama dan rambut. Atribut-atribut ini dideklarasikan sebagai variabel bertipe String, yang berarti dapat menampung nilai teks. Kelas ini juga memiliki konstruktor default public Manusia() yang mencetak pesan ke konsol ketika sebuah objek dari kelas dibuat.

Berdasarkan analisis, kita dapat berpendapat bahwa kelas Manusia dirancang untuk merepresentasikan seseorang atau manusia. Atribut nama merepresentasikan nama orang tersebut, sedangkan atribut rambut merepresentasikan jenis atau karakteristik rambutnya. Konstruktor default menunjukkan bahwa ketika objek Manusia dibuat, ia harus mencetak pesan yang mengindikasikan bahwa objek tersebut adalah orang tanpa nama.

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

Algoritma:

- (a) Identifikasi Atribut
- (b) Mendefinisikan Atribut
- (c) Membuat Konstruktor
- (d) Menentukan Metode
- (e) Menerapkan Metode
- (f) Menguji Kelas

### **[No.1] Kesimpulan**

#### **Analisa**

Sebagai kesimpulan, kita telah menganalisis dan berargumen tentang karakteristik kelas Manusia, mengembangkan algoritma untuk membuat kelas yang lebih komprehensif dan serbaguna, dan mengimplementasikan kelas tersebut dengan atribut dan perilaku yang penting.

Melalui latihan ini, kita telah menunjukkan pentingnya mempertimbangkan kelengkapan dan keserbagunaan sebuah kelas dalam pemrograman berorientasi objek. Dengan menambahkan lebih banyak atribut dan metode, kita dapat membuat sebuah kelas yang lebih berguna dan fleksibel, dan dapat digunakan dalam aplikasi yang lebih luas.

Secara keseluruhan, latihan ini telah menyoroti betapa pentingnya pemikiran dan analisis sistematis dalam pengembangan perangkat lunak, dan telah menunjukkan manfaat dari membuat kelas yang komprehensif dan serbaguna yang dapat digunakan dalam berbagai aplikasi.

## LATIHAN 2

### [No. 2] Identifikasi Masalah:

Dalam analisis kode, kita dapat mengidentifikasi beberapa masalah yang perlu diatasi, yaitu atribut yang tidak digunakan seperti namaAnak, umurAnak, dan sekolahAnak, mekanisme inheritance yang tidak efektif, dokumentasi yang tidak memadai, fleksibilitas dan skalabilitas yang kurang, serta perilaku yang tidak spesifik seperti bermain, belajar, makan, menggambar, membacaBuku, dan bermainMusik.

### [No.2] Analisis dan Argumentasi

Kita perlu mempertimbangkan apakah atribut-atribut anak harus diwarisi dari kelas Ortu atau tidak. Jika ya, maka kita perlu menambahkan mekanisme inheritance yang lebih baik. Jika tidak, maka kita perlu mempertimbangkan bagaimana cara menghubungkan atribut-atribut anak dengan atribut-atribut ortu.

### [No.2 ] Penyusunan Algoritma dan Kode Program

Algoritma:

Bagian 2.1

1. Identifikasi atribut yang dibutuhkan dan hapus atribut yang tidak digunakan.
2. Perbaiki mekanisme inheritance agar lebih efektif dan logis.
3. Tambahkan dokumentasi yang memadai untuk memudahkan pemahaman kode.
4. Buat mekanisme yang memungkinkan penambahan atau penghapusan atribut dan perilaku anak dengan mudah.
5. Spesifikkan perilaku anak agar lebih unik dan tidak dapat diaplikasikan pada kelas Ortu

Kode program dan luar

```
1- public class Ortu {
2-     // deklarasi atribut
3-     private String nama;
4-     private String rambut;
5-     private int umur;
6-     private String pekerjaan;
7-     private String alamat;
8-
9-     // deklarasi constructor
10-    public Ortu(String nama, String rambut, int umur, String pekerjaan, String
        alamat) {
11-        this.nama = nama;
12-        this.ambut = rambut;
13-        this.umur = umur;
14-        this.pekerjaan = pekerjaan;
15-        this.alamat = alamat;
16-
17-        System.out.println(" Nama saya : " + nama);
18-        System.out.println(" Warna Rambut : " + rambut);
19-        System.out.println(" Umur saya : " + umur);
20-        System.out.println(" Pekerjaan saya : " + pekerjaan);
21-        System.out.println(" Alamat saya : " + alamat);
22-    }
23-
24-    public static void main(String[] args) {
25-        Ortu satu = new Ortu("Uno", "hitam", 18, "Engineer", "Surabaya");
26-    }
27- }
```

```
java -cp ./src/Ortu/Ortu
Nama saya : Uno
Warna Rambut : hitam
Umur saya : 18
Pekerjaan saya : Engineer
Alamat saya : Surabaya

=== Code Execution Successful ===
```

## **[No.2] Kesimpulan**

### **Analisa**

Dalam analisis kode, terdapat beberapa masalah yang perlu diatasi, seperti atribut yang tidak terpakai, mekanisme pewarisan yang tidak efektif, dokumentasi yang tidak memadai, kurangnya fleksibilitas dan skalabilitas, dan perilaku yang tidak spesifik. Untuk mengatasi masalah-masalah ini, kami telah mengembangkan algoritma dan kode program yang lebih efektif dan efisien. Meskipun perbaikan lebih lanjut masih diperlukan, kode program telah menunjukkan peningkatan dalam hal fleksibilitas dan skalabilitas.

## **LATIHAN 3**

### **[No. 3] Identifikasi Masalah:**

- 3.1. Analisa perbedaan deklarasi constructor, method, dan method utama!
- 3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?
- 3.3. Uraikan perbedaan berikut:
  - a) constructor overloading dan overriding
  - b) method overloading, dan method overriding
  - c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

### **[No. 3] Analisis dan Argumentasi**

Dalam analisis kode program sebelumnya, kita dapat melihat bahwa terdapat beberapa masalah yang perlu diperhatikan. Nama constructor harus sama dengan nama kelas karena constructor digunakan untuk membuat objek dari kelas tersebut, dan inisialisasi atribut harus lengkap agar objek yang dibuat memiliki nilai yang sesuai. Akses modifier harus sesuai dengan kebutuhan karena menentukan siapa yang dapat mengakses method atau atribut, dan penanganan error harus memadai agar program dapat menangani kesalahan yang terjadi. Selain itu, atribut yang relevan dengan manusia harus diikutsertakan dalam kelas Manusia agar kelas tersebut dapat merepresentasikan manusia dengan lebih baik. Dengan analisis dan argumentasi tersebut, kita dapat melihat bahwa perbaikan kode program yang telah dilakukan adalah sangat penting untuk membuat kode program yang lebih baik dan efektif.

### **Argumentasi**

Constructor Overloading ini terjadi ketika beberapa konstruktor dengan daftar parameter yang berbeda didefinisikan dalam sebuah kelas. Konstruktor yang akan dipanggil ditentukan oleh jumlah dan jenis argumen yang dilewatkan ketika objek dibuat. Overriding ini tidak mungkin dilakukan di Java, karena konstruktor tidak dapat diwariskan oleh subkelas. Namun, subkelas bisa menyediakan konstruktor sendiri yang memiliki nama yang sama dengan konstruktor superclass, tetapi dengan daftar parameter yang berbeda.

### **[No. 3] Penyusunan Algoritma dan Kode Program**

1. Deklarasi Kelas
2. Deklarasi Atribut
3. Deklarasi Konstruktor
4. Deklarasi Metode
5. Membuat Metode Utama

### **[No. 3] Kesimpulan**

#### **Analisa**

Dapat disimpulkan bahwa kode program tersebut mendeskripsikan sebuah class Human dengan dua atribut yaitu nama dan rambut, memiliki konstruktor yang mencetak nama dan warna rambut, serta method likeWatch yang mencetak film favorit, dan program tersebut akan mencetak output "My name is : Putri, Warna Rambut : Hitam, Film Favorit: Drakor" setelah mengoreksi kesalahan penamaan konstruktor dari Human1 menjadi Human.

**Refleksi**

Pengalaman saya saat belajar materi dalam mata kuliah ini ialah perlu lebih banyak belajar lagi karena saya kurang mengerti apa yang di sampaikan dalam materi ini.