

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Dewangga Nurgoho Anwar G1F024045</b>	<b>Class</b>	<b>12 September 2024.</b>

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

- 1) Uraikan permasalahan dan variabel  
Pada soal masih ada kode program yang kurang yang pertama terletak pada line 6 kurang nya tipe data *String rambut*. Selanjutnya pada line 13 pada objek *new manusia1* yang seharusnya tidak memasukan angka 1.

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

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan tipe data String rambut pada line 6 dan menghapus angka 1 pada objek ney manusia1.
- 2) Perbaiki kode program dengan cara menambahkan String rambut pada line 6 dan menghapus angka 1 pada objek ney manusia1.

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

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (a) Mulai
  - (b) Deklarasi Kelas Manusia
  - (c) Deklarasi Constructor Manusia
  - (d) Method Utama (main)
  - (e) Membuat Objek Manusia
  - (f) Eksekusi Constructor
  - (g) Akhir Program
- 2) Kode program dan luaran .
  - a) Screenshot/ Capture potongan kode dan hasil luaran.

```

1 //Mulai
2 public class Manusia { // deklarasi kelas
3     //deklarasi atribut Manusia dalam variabel
4     String nama, rambut, pekerjaan, jeniskelamin;
5     int umur;
6
7     //deklarasi constructor
8     public Manusia (String nama, String rambut, String pekerjaan,
9         String jeniskelamin, int umur) {
10         System.out.println(" Nama saya : "+ nama +
11             "\n Warna Rambut : " + rambut+
12             "\n Pekerjaan Joko : " + pekerjaan +
13             "\nJenis Kelamin Joko : " + jeniskelamin +
14             "\nUmur : " + umur);
15     }
16
17     //deklarasi method utama
18     public static void main( String[] args) {
19         Manusia satu = new Manusia("Joko", "hitam", "Polisi",
20             "Laki Laki", 23);
21     }
22 }

```

Gambar input 1.0

```
java -cp /tmp/0SwHqBtSlp/Manusia  
Nama saya : Joko  
Warna Rambut : hitam  
Pekerjaan Joko : Polisi  
Jenis Kelamin Joko : Laki Laki  
Umur : 23  
  
=== Code Execution Successful ===|
```

Gambar Output 1.1

- a) Analisa luaran yang dihasilkan.  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

### [No.1] Kesimpulan

#### 1. Kreasi

- a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?.

Jawab.

Pada kode jawaban, ada pengembangan pengetahuan dengan menambahkan lebih banyak atribut untuk mendeskripsikan objek Manusia. Konsep baru yang dikembangkan adalah memperluas cakupan informasi yang disimpan dalam objek, yang dapat mendukung deskripsi manusia yang lebih detail (misalnya, pekerjaan dan jenis kelamin). Ini merupakan solusi yang tepat jika program membutuhkan representasi yang lebih kompleks dari manusia.

- b) Susunlah hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada).

Jawab.

Hubungan antar variabel menjadi lebih jelas dalam mendeskripsikan karakteristik manusia. Variabel nama, rambut, pekerjaan, jenis kelamin, dan umur saling terkait dalam memberikan informasi yang lebih lengkap tentang objek **Manusia**. Hubungan ini mencerminkan realitas di mana seseorang tidak hanya memiliki nama dan warna rambut, tetapi juga pekerjaan, jenis kelamin, dan umur, yang semuanya berperan dalam membentuk identitas individu. Variabel-variabel tersebut bersama-sama memberikan gambaran manusia yang lebih utuh dan fungsional dalam konteks aplikasi yang memerlukan data manusia.

### [No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel  
Pada soal masih ada kode program yang kurang yang pertama terletak pada line 2 yaitu kode *String rambut* dan pada class public yang seharusnya huruf o besar namun di kode memakai huruf o kecil.

### [No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan tipe data *String* rambut pada line 2 dan mengubah huruf o kecil pada constructor *public ortu* menjadi huruf besar karena tidak sesuai dengan method utama.
- 2) Perbaiki kode program dengan cara menambahkan *String* rambut pada line 2 dan mengubah huruf o kecil pada constructor *public ortu* menjadi huruf besar

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

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (a) Mulai
  - (b) Deklarasi Kelas Ortu.
  - (c) Deklarasi Constructor Ortu.
  - (d) Method Utama.
  - (e) Eksekusi Constructor
  - (f) Akhir Program
- 2) Kode program dan luaran .
  - a. Screenshot/ Capture potongan kode dan hasil luaran.

```
1+ public class Ortu {
2    //deklarasi constructor (variabel constructor)
3+ public Ortu (String nama, String rambut, int umur, String
    jeniskelamin) {
4+     System.out.println(" Nama saya : " + nama +
5         "\n Warna Rambut : " + rambut +
6         "\n Umur : " + umur +
7         "\n Jenis Kelamin : " + jeniskelamin);
8 }
9+ public static void main (String[] args) {
10     Ortu satu = new Ortu("Putri", "hitam", 25, "P");
11 }
12 }
```

Gambar input 2.0

```
java -cp /tmp/wNqGIDaW3Z/Ortu
Nama saya : Putri
Warna Rambut : hitam
Umur : 25
Jenis Kelamin : P

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

Gambar Output 2.1

- b) Analisa luaran yang dihasilkan.  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

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

### **1) Kreasi**

- a) Pada kode jawaban, ada pengembangan pengetahuan dengan mencerminkan peningkatan signifikan dalam desain program Java. Kode awal mengalami kesalahan sintaksis karena constructor yang tidak sesuai nama kelas dan kekurangan parameter untuk menginisialisasi variabel nama dan rambut. Perbaikan di Kode 2 memperbaiki nama constructor menjadi Ortu, yang sesuai dengan nama kelas, dan menambahkan parameter nama, rambut, umur, dan jeniskelamin untuk memberikan informasi yang lebih lengkap. Dengan penyesuaian ini, constructor kini dapat menyimpan dan menampilkan data yang lebih detail, menjadikan objek Ortu lebih informatif dan fungsional.

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

1. Berdasarkan Contoh 3 dan Latihan 3.2. simpulkan perbedaan:
  - a) constructor overloading dan overriding
  - b) method overloading, dan method overriding
  - c) method yang mengembalikan nilai dan method tidak mengembalikan nilai.

Jawab:

- a) Letak kesalahan pada kode tersebut pada line 6 yang seharusnya public Manusia, lalu terletak pada void sukaNonton kurang nya tipe data (String film).
- b) Overloading adalah teknik mendefinisikan beberapa metode dengan nama yang sama dalam satu kelas. Sedangkan Overriding adalah teknik untuk mendefinisikan kembali metode yang sudah ada dalam superclass di dalam subclass.
- c) Method yang mengembalikan nilai terletak pada kode sukaNonton(int episode, int durasi). Sedangkan method yang tidak mengembalikan nilai terletak pada kode sukaNonton(String film)

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

- 1) Saya mengusulkan untuk menghapus angka 1 pada deklarasi constructor menjadi Manusia, selanjutnya menambahkan String pada deklarasi method menjadi void sukaNonton (String film), dan pada deklarasi method int sukaNonton di tambahkan String dan menjadi int SukaNonton( String episode, String durasi) lalu tambahkan fungsi return pada episode\*durasi.
- 2) Perbaiki kode program dengan cara menambahkan String rambut pada line 6 dan menghapus angka 1 pada objek ney manusia1.

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

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (a) Mulai
  - (b) Deklarasi kelas
  - (c) Implementasi constructor
  - (d) Declarasi method
  - (e) Selesai
- b) Kode program dan luaran .

```

1 public class Manusia {
2     // Deklarasi atribut Manusia
3     String nama, rambut;
4
5     // Deklarasi constructor
6     public Manusia(String nama, String rambut) {
7         System.out.println("Nama saya: " + nama +
8             "\nWarna Rambut: " + rambut);
9     }
10
11     // Deklarasi method pertama
12     void sukaNonton(String film) {
13         System.out.println("Hobi Menonton: " + film);
14     }
15
16     // Deklarasi method kedua (overloading) dengan dua parameter numerik
17     int sukaNonton(int episode, int durasi) {
18         return episode * durasi;
19     }
20
21     // Deklarasi method utama
22     public static void main(String[] args) {
23         Manusia satu = new Manusia("DewanggaNurgohoAnwar", "hitam");
24         satu.sukaNonton("Drakor");
25
26         // Memanggil method yang mengembalikan jam nonton
27         int jumlahJam = satu.sukaNonton(2, 2);
28         System.out.println("Jam nonton = " + jumlahJam + " jam");
29     }
30 }
31

```

Gambar input 3.0

- a) Screenshot/ Capture potongan kode dan hasil luaran.

```

Nama saya: DewanggaNurgohoAnwar
Warna Rambut: hitam
Hobi Menonton: Drakor
Jam nonton = 4 jam

```

Gambar Output 3.1

- b) Analisa luaran yang dihasilkan.  
 Luaran sudah sesuai dengan program yang disusun.  
 Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

### [No.3] Kesimpulan

#### 1) Evaluasi.

- a) Pada kode jawaban, ada pengembangan pengetahuan dengan menambahkan lebih banyak atribut untuk mendeskripsikan objek Manusia. Konsep baru yang dikembangkan adalah memperluas cakupan informasi yang disimpan dalam objek, yang dapat mendukung deskripsi manusia yang lebih detail (misalnya, pekerjaan dan jenis kelamin). Ini merupakan solusi yang tepat jika program membutuhkan representasi yang lebih kompleks dari manusia. Hubungan antar variabel menjadi lebih jelas dalam mendeskripsikan karakteristik manusia. Variabel nama, rambut, pekerjaan, jenis kelamin, dan umur saling terkait dalam memberikan informasi yang lebih lengkap tentang objek **Manusia**. Hubungan ini mencerminkan realitas di mana seseorang tidak hanya memiliki nama dan warna rambut, tetapi juga pekerjaan, jenis kelamin, dan umur, yang semuanya berperan dalam membentuk identitas individu. Variabel-variabel tersebut bersama-sama memberikan gambaran manusia yang lebih utuh dan fungsional dalam konteks aplikasi yang memerlukan data manusia.

#### [No. 4] Identifikasi Masalah:

- 1) Apakah penulisan method ini sudah efisien?  
Penulisan ulang method `sukaMenonton(String a)` di kelas `Anak` tidak efisien jika implementasinya sama dengan di kelas `Ortu`. Ini menyebabkan redundansi. Jika tidak ada perubahan dalam fungsionalitasnya, sebaiknya method ini tidak di-override di kelas `Anak`.
- 2) Setelah dirunning di JDoodle, catat waktu eksekusinya. Rekomendasikan perbaikan penulisan kode method untuk dapat mengefisienkan waktu eksekusi!  
Program tersebut dapat di running dalam waktu CPU Time: 0.07 sec(s) Memory: 38840 kilobyte(s) dan Compiled and executed in 2.658 sec(s). Untuk efisiensi, kita dapat menggunakan method di kelas `anak` hanya di-override jika ada perubahan yang signifikan dalam fungsionalitas. Method yang tidak memerlukan perubahan sebaiknya diwariskan langsung dari kelas induk. Penggunaan method overloading harus dilakukan dengan tujuan menambahkan fungsionalitas baru, bukan hanya untuk mendeklarasikan ulang method yang ada.

#### [No.4] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan tipe data `String` rambut pada line 6 dan menghapus angka 1 pada objek `ney manusia1`.
- 2) Perbaikan kode program dengan cara menambahkan `String` rambut pada line 6 dan menghapus angka 1 pada objek `ney manusia1`.

#### [No.4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (a) Mulai
  - (b) Deklarasi Kelas `Ortu`.
  - (c) Deklarasi Constructor `Ortu`.
  - (d) Method Utama.
  - (e) Eksekusi Constructor
  - (f) Akhir Program
- 2) Kode program dan luaran .
  - a) Screenshot/ Capture potongan kode dan hasil luaran.

```
1- public class Ortu { // membuat kelas induk
2- void sukaMenonton(String a) { // method induk spesifik
3-     System.out.println("Nonton " + a);
4- }
5-
6- void sukaMembaca(String a) { // method induk umum bisa diubah anak
7-     System.out.println("Suka Baca " + a);
8- }
9-
10- public static void main(String[] args) {
11-     System.out.println("Sifat Orang Tua :");
12-     Ortu objek0 = new Ortu(); // memanggil objek induk
13-     objek0.sukaMenonton("Berita"); // memanggil sifat spesifik induk
14-     objek0.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
15-
16-     System.out.println("\nSifat Anak :");
17-     Anak objekA = new Anak(); // memanggil objek anak
18-     objekA.sukaMenonton(2, "Film Action"); // memanggil method overloading di anak
19-     objekA.sukaMembaca("Komik Doraemon"); // memanggil method yang di-override di anak
20- }
21- }
22-
23- class Anak extends Ortu {
24-     // Method overloading: Menambahkan variant baru untuk sukaMenonton
25- void sukaMenonton(int a, String b) {
26-     System.out.println("Nonton Jam " + a + " Malam " + b);
27- }
28-
29- // Method overriding: Mengubah implementasi method sukaMembaca
30- @Override
31- void sukaMembaca(String a) {
32-     System.out.println("Suka Baca " + a);
33- }
34- }
```

Gambar Input 4.0

```
java -cp /tmp/8Q0Zb0hnsW/Ortu  
Sifat Orang Tua :  
Nonton Berita  
Suka Baca Koran  
  
Sifat Anak :  
Nonton Jam 2 Malam Film Action  
Suka Baca Komik Doraemon  
  
=== Code Execution Successful ===
```

Gambar Output 4.1

- a) Analisa luaran yang dihasilkan.  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

#### [No.4] Kesimpulan

- 1) Evaluasi
  - a) Program ini mengilustrasikan penggunaan konsep inheritance dalam Java dengan mengimplementasikan method overloading dan method overriding. Kelas Ortu sebagai kelas induk memiliki dua method: `sukaMenonton(String a)` yang menampilkan informasi tentang menonton dan `sukaMembaca(String a)` yang menampilkan informasi tentang membaca. Kelas Anak sebagai subclass meng-extend kelas Ortu dengan menambahkan method baru `sukaMenonton(int a, String b)` yang memperluas fungsionalitas menonton dengan informasi tambahan seperti waktu. Selain itu, kelas Anak juga meng-override method `sukaMembaca(String a)` dari kelas Ortu untuk menyesuaikan dengan kebutuhan khusus, meskipun dalam hal ini implementasinya sama dengan kelas induk. Program ini menunjukkan bagaimana subclass dapat memperluas dan memodifikasi fungsionalitas dari superclass dengan cara yang terstruktur dan efisien.