

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Nadya Putri Marsela G1F024036</b>	<b>Unit 1 Kelas</b>	<b>18 September 2024</b>
<b>[Nomor Soal] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <pre>public class Manusia { // deklarasi kelas     // deklarasi variabel     String nama;     String rambut;     // deklarasi constructor tanpa parameter     public Manusia() {         System.out.println("Kelas Manusia tanpa nama");     } }</pre> <p>Latihan 1:</p> <p>1.1. Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi</p> <ol style="list-style-type: none"> <li>atribut variabel, dan</li> <li>perilaku/ behavior untuk method!</li> </ol> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Video materi 1 tentang Kelas, Objek, Method -  <a href="https://www.youtube.com/watch?v=60ldOc8m8Es%C2%A0">https://www.youtube.com/watch?v=60ldOc8m8Es%C2%A0</a></p>		
<b>[Nomor Soal] Analisis dan Argumentasi</b>		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <ol style="list-style-type: none"> <li>Atribut Variabel (Properties/Attributes). Atribut mewakili data atau informasi yang dimiliki oleh objek dari kelas Manusia. Contohnya, nama (String nama), warna kulit (String warnaKulit), warna rambut (String rambut)</li> <li>Perilaku (Behavior/Method). Perilaku adalah tindakan atau fungsi yang dapat dilakukan oleh objek dari kelas Manusia.</li> </ol> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <ol style="list-style-type: none"> <li>Atribut variabel yang didefinisikan dalam solusi menangkap ciri-ciri umum manusia yang relevan, baik dari aspek identitas (nama, jenis kelamin, usia, kebangsaan) maupun dari aspek fisik (tinggi, berat, warna rambut).</li> <li>Perilaku yang dirancang sebagai metode menangkap kegiatan umum manusia seperti berbicara, berjalan, makan, tidur, dan memperkenalkan diri, yang semuanya merepresentasikan tindakan yang sering dilakukan manusia.</li> <li>Constructor dengan parameter memberikan fleksibilitas yang memungkinkan kita membuat objek Manusia dengan atau tanpa informasi awal.</li> </ol>		
<b>[Nomor Soal] Penyusunan Algoritma dan Kode Program</b>		
<p>1) Rancang desain solusi atau algoritma</p> <ol style="list-style-type: none"> <li>Salin potongan kode pada VSCode</li> <li>Deklarasi kelas Manusia dengan variabel nama dan rambut. \</li> </ol>		

3. Membuat constructor untuk menginisialisasi variabel.
4. Menambahkan method perkenalkanDiri() untuk memperkenalkan diri.
5. Selesai.

2) Tuliskan kode program dan luaran

```
1 public class Manusia { // deklarasi kelas
2     // deklarasi variabel
3     String nama;
4     String rambut;
5
6     // deklarasi constructor tanpa parameter
7     public Manusia(String nama, String rambut) {
8         System.out.println("Nama : " + nama + "\nWarna rambut : " + rambut);
9     }
10
11     void sukaMenonton (String anime) {
12         System.out.println("Hobi menonton : " + anime);
13     }
14
15     public static void main(String[] args) {
16         Manusia random = new Manusia ("Toji", "Hitam");
17         random.sukaMenonton("Menonton");
18     }
19 }
20 // kelas main
```

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Run: Manusia + - - - ^
PS C:\Users\ASUS> & 'C:\Program Files\Java\jdk1.8.0_341\bin\java.exe' '-cp' 'C:\Users\ASUS\AppData\Local\Temp\vscodesws_6
ac06\jdt_ws\jdt.ls-java-project\bin' 'Manusia'
Nama : Toji
Warna rambut : Hitam
Hobi menonton : Menonton
PS C:\Users\ASUS>
```

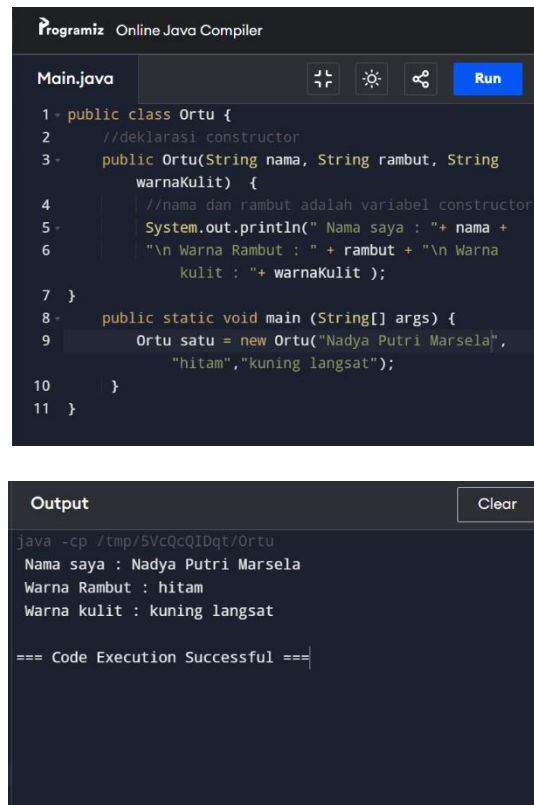
### [Nomor Soal] Kesimpulan

1) Analisa

Kelas Manusia dirancang dengan atribut yang mencerminkan ciri-ciri umum manusia seperti nama, jenis kelamin, usia, tinggi, berat badan, rambut, dan kebangsaan, yang merepresentasikan karakteristik penting manusia dalam pemrograman berorientasi objek. Perilaku manusia seperti berbicara, berjalan, makan, tidur, dan memperkenalkan diri diimplementasikan sebagai metode yang menggambarkan tindakan yang dapat dilakukan oleh objek Manusia. Dengan menambahkan constructor tanpa dan dengan parameter, kelas ini menjadi fleksibel untuk menciptakan objek dengan atau tanpa data awal, sehingga solusi ini berhasil memodelkan manusia secara realistis dalam kode program.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Nadya Putri Marsela G1F024036</b>	<b>Unit 2 Objek</b>	<b>18 September 2024</b>
<b>[Nomor Soal] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <pre> public class Ortu {     //deklarasi constructor     public Ortu(String nama, String rambut) {         //nama dan rambut adalah variabel constructor         System.out.println(" Nama saya : "+ nama +             "\n Warna Rambut : " + rambut);     }     public static void main (String[] args) {         Ortu satu = new Ortu("Putri", "hitam");     } } </pre> <p><b>Luaran 2:</b>  Nama saya : Putri  Warna Rambut : hitam</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Video materi 1 tentang Kelas, Objek, Method -  <a href="https://www.youtube.com/watch?v=60IdOc8m8Es%C2%A0">https://www.youtube.com/watch?v=60IdOc8m8Es%C2%A0</a></p>		
<b>[Nomor Soal] Analisis dan Argumentasi</b>		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <ol style="list-style-type: none"> <li>1. Deklarasi kelas Ortu yang akan merepresentasikan objek orang tua dengan beberapa atribut.</li> <li>2. Constructor dengan parameter nama dan rambut.</li> <li>3. Cetak informasi untuk mencetak nama dan warna rambut saat objek diciptakan.</li> <li>4. Pembuatan Objek dalam main:</li> <li>5. Constructor akan dipanggil secara otomatis dan menampilkan keluaran sesuai dengan nilai yang diberikan.</li> </ol> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi menggunakan constructor dengan parameter untuk menginisialisasi atribut nama dan rambut dan mencetaknya sesuai kebutuhan soal. Pendekatan ini efektif dan langsung, dengan implementasi sederhana melalui metode main() yang membuat objek dan menghasilkan output yang diinginkan.</p>		
<b>[Nomor Soal] Penyusunan Algoritma dan Kode Program</b>		
<p>3) Rancang desain solusi atau algoritma</p> <ol style="list-style-type: none"> <li>1. Salin potongan kode</li> <li>2. Deklarasi kelas Ortu dengan variabel tambahan (jika ada)</li> <li>3. Implementasikan constructor untuk menerima variabel baru dan menampilkan informasi.</li> <li>4. Method main untuk membuat objek Ortu dan menampilkan informasi.</li> </ol>		

4) Tuliskan kode program dan luaran



The screenshot shows the Programiz Online Java Compiler interface. The code editor displays a Java program named Main.java. The code defines a class Ortu with a constructor that takes three parameters: nama, rambut, and warnakulit. The constructor prints the name, hair color, and skin color. The main method creates an instance of Ortu with the values "Nadya Putri Marsela", "hitam", and "kuning langsung". The output window shows the execution results: "Nama saya : Nadya Putri Marsela", "Warna Rambut : hitam", and "Warna kulit : kuning langsung". The code execution was successful.

```
1 public class Ortu {
2     //deklarasi constructor
3     public Ortu(String nama, String rambut, String
    warnakulit) {
4         //nama dan rambut adalah variabel constructor
5         System.out.println(" Nama saya : "+ nama +
6         "\n Warna Rambut : " + rambut + "\n Warna
            kulit : "+ warnakulit );
7     }
8     public static void main (String[] args) {
9         Ortu satu = new Ortu("Nadya Putri Marsela",
            "hitam", "kuning langsung");
10    }
11 }
```

Output

```
java -cp /tmp/5VcQcQIDqt/Ortu
Nama saya : Nadya Putri Marsela
Warna Rambut : hitam
Warna kulit : kuning langsung

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

#### [Nomor Soal] Kesimpulan

2) Analisa

Solusi yang diimplementasikan telah berhasil menyelesaikan permasalahan dengan cara sederhana dan tepat. Kelas Ortu menggunakan constructor yang menerima parameter untuk inisialisasi atribut nama dan rambut, kemudian langsung mencetak informasi tersebut saat objek dibuat. Metode ini sesuai dengan soal, menghasilkan output yang diinginkan tanpa memerlukan metode tambahan. Dengan pendekatan ini, program menjadi lebih efisien dan mudah dipahami, serta memenuhi kebutuhan untuk mencetak informasi nama dan warna rambut secara langsung

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Nadya Putri Marsela G1F024036</b>	<b>Unit 3 Method</b>	<b>18 September 2024</b>
<b>[Nomor Soal] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <pre> public class Manusia {     //deklarasi atribut Manusia dalam variabel     String nama, rambut;      //deklarasi constructor     public Manusia1(String nama, String rambut) {         System.out.println(" Nama saya : "+ nama +             "\n Warna Rambut : " + rambut);     }      //deklarasi method     void sukaNonton(String film) {         System.out.println(" Hobi Menonton : " + film);     }      //deklarasi method utama     public static void main( String[] args) {         Manusia satu = new Manusia("Putri", "hitam");         satu.sukaNonton("Drakor");     } } </pre> <p><b>Luaran 3:</b>  Nama saya : Putri  Warna Rambut : hitam  Hobi Menonton : Drakor</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Video materi 1 tentang Kelas, Objek, Method -  <a href="https://www.youtube.com/watch?v=60ldOc8m8Es%C2%A0">https://www.youtube.com/watch?v=60ldOc8m8Es%C2%A0</a></p>		
<b>[Nomor Soal] Analisis dan Argumentasi</b>		
<p>1) Analisa perbedaan deklarasi constructor, method, dan method utama</p> <ul style="list-style-type: none"> <li>• Constructor Merupakan method khusus yang digunakan untuk menginisialisasi objek Ketika dibuat. Dalam contoh kode program, constructor dinyatakan sesuai dengan nama kelas, missal nama kelasnya adalah Manusia maka constructor juga memiliki nama Manusia. Constructor dipanggil secara otomatis saat objek baru dibuat.</li> <li>• Method Merupakan blok kode yang bisa dieksekusi oleh objek untuk melakukan tindakan tertentu, method memiliki nama yang bisa bebas dipilih dan bisa dipanggil kapanpun oleh objek.</li> <li>• Method utama</li> </ul>		

Merupakan titik awal eksekusi dalam program di Java, method utama merupakan method yang akan dijalankan pertama kali Ketika program dijalankan.

2) Kapan constructor dan method perlu digunakan

- Constructor

Digunakan untuk menginisialisasi objek saat pertama kali dibuat, jika suatu objek memerlukan nilai awal untuk atribut-atributnya maka constructor diperlukan.

- Method

Digunakan Ketika ingin objek melakukan suatu Tindakan setelah objek dibuat, method bisa dipanggil kapan saja sesuai kebutuhan.

3) Uraikan perbedaan berikut

- Constructor overloading dan overriding

Constructor overloading terjadi ketika sebuah kelas memiliki beberapa constructor dengan parameter yang berbeda, memungkinkan objek dibuat dengan cara yang berbeda. Namun, constructor tidak bisa di-*override* karena tidak diwariskan oleh subclass, sehingga konsep overriding tidak berlaku untuk constructor.

- Method overloading dan overriding

Method overloading adalah saat kita membuat beberapa method dengan nama yang sama tetapi berbeda parameter dalam satu kelas. Ini memungkinkan pemanggilan method dengan berbagai input. Di sisi lain, method overriding adalah ketika sebuah subclass menyediakan implementasi yang berbeda untuk method yang sudah ada di superclass, menggunakan nama dan parameter yang sama, namun dengan perilaku yang berbeda.

- Method yang mengembalikan nilai dan method yang tidak mengembalikan nilai  
Method yang mengembalikan nilai memberikan output berupa tipe data tertentu setelah eksekusi dan menggunakan return untuk mengirimkan nilai tersebut. Sementara itu, method yang tidak mengembalikan nilai menggunakan tipe void dan hanya melakukan aksi tertentu tanpa memberikan nilai balik.

**[Nomor Soal] Penyusunan Algoritma dan Kode Program**

1) Rancang desain solusi atau algoritma

1. Salin potongan kode
2. Deklarasikan kelas Manusia dengan atribut nama dan rambut.
3. Implementasikan constructor Manusia yang menerima parameter nama dan rambut.
4. Implementasikan method sukaNonton untuk mencetak hobi menonton film.
5. Implementasikan method main untuk membuat objek Manusia dan memanggil method sukaNonton.

2) Tuliskan kode program dan luaran

```

public class Manusia {
    //deklarasi atribut Manusia dalam variabel
    String nama, rambut;

    //deklarasi constructor
    public Manusia(String nama, String rambut) {
        System.out.println(" Nama saya : " + nama +
            "\n Warna Rambut : " + rambut);
    }

    //deklarasi method
    void sukaNonton(String film) {
        System.out.println(" Hobi Menonton : " + film);
    }

    //deklarasi method utama
    public static void main( String[] args) {
        Manusia satu = new Manusia("Putri", "hitam");
        satu.sukaNonton("Drakor");
    }
}

```

#### Output

```

java -cp ./tmp/cfDgiWw9/Manusia
Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor

=== Code Execution Successful ===

```

### [Nomor Soal] Kesimpulan

#### 1) Analisa

Kode program dengan perbaikan pada constructor dan implementasi method menunjukkan cara kerja yang benar dari constructor, method, dan method utama. Constructor menginisialisasi objek, method sukaNonton memberikan perilaku tambahan, dan method main menjalankan program.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Nadya Putri Marsela G1F024036</b>	<b>Unit 4 Extends</b>	<b>18 September 2024</b>

**[Nomor Soal] Identifikasi Masalah:**

1) Uraikan permasalahan dan variable

```

public class Ortu {    // membuat kelas induk
    void sukaMenonton(String a) { // method induk spesifik
        System.out.println("Nonton " + a);
    }
    void sukaMembaca(String a) { // method induk umum bisa diubah anak
        System.out.println("Suka Baca " + a);
    }
    public static void main(String [] args) {
        System.out.println("Sifat Orang Tua :");
        Ortu objekO = new Ortu(); // memanggil objek induk
        objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
        objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah

        System.out.println("\n Sifat Anak :");
        Anak objekA = new Anak(); //memanggil objek anak
        objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan
        induk
        objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
        diturunkan tanpa deklarasi ulang di anak
    } }
class Anak extends Ortu {
    void sukaMenonton(int a, String b) {
        System.out.println("Nonton Jam " + a + " Malam " + b);
    }
    void sukaMenonton(String a) { // method induk spesifik
        System.out.println("Nonton " + a);
    }
    void sukaMembaca(String a) { // method induk umum bisa diubah anak
        System.out.println("Suka Baca " + a);
    }
}

public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu(); // memanggil objek induk
    objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
    objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak(); //memanggil objek anak
    objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan
    induk
    objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
    diturunkan tanpa deklarasi ulang di anak
} }

```



Luaran 4:

Sifat Orang Tua :

Nonton Berita

Suka Baca Koran

Sifat Anak :

Nonton Jam 9 Malam Film Drakor

Suka Baca Komik One Piece

**Latihan 4:**

4.1. Bandingkan method yang dimiliki class Anak extends Ortu dengan method di class Ortu!

4.2. Ubahlah Contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

- 2) Rincikan sumber informasi yang relevan (buku / webpage)

Video materi 1 tentang Kelas, Objek, Method -

<https://www.youtube.com/watch?v=60ldOc8m8Es%C2%A0>

**[Nomor Soal] Analisis dan Argumentasi**

- 1) Uraikan rancangan solusi yang diusulkan.

Tambahkan method baru `sukaBerolahraga` di kelas Anak. Modifikasi kode untuk menunjukkan perbedaan antara method yang di-override dari kelas Ortu dan method baru yang ditambahkan. Tunjukkan hasil output yang sesuai untuk membandingkan method.

- 2) Analisis solusi, kaitkan dengan permasalahan.

Dengan menambahkan method baru di kelas Anak, kita dapat melihat bagaimana kelas anak dapat menambahkan fungsionalitas tambahan yang tidak ada di kelas induk. Overriding method yang ada di kelas Ortu memungkinkan kelas Anak untuk mengubah cara kerja method tersebut, sedangkan method overloading memberikan opsi untuk menggunakan nama method yang sama dengan parameter berbeda.

**[Nomor Soal] Penyusunan Algoritma dan Kode Program**

- 1) Rancang desain solusi atau algoritma

1. Salin potongan kode
2. Definisi Kelas Ortu: Definisikan kelas dengan method `sukaMenonton` dan `sukaMembaca`.
3. Definisi Kelas Anak: Definisikan kelas yang mewarisi Ortu, override method `sukaMenonton` dan `sukaMembaca`, tambahkan method baru `sukaBerolahraga`.
4. Main Method: Tampilkan penggunaan method di kelas Ortu dan Anak.

- 2) Tuliskan kode program dan luaran

```
MultiJava
1 public class Ortu { // membuat kelas induk
2     void sukaMenonton(String a) { // method induk spesifik
3         System.out.println("Nonton " + a);
4     }
5     void sukaMembaca(String a) { // method induk anak bisa diarahkan
6         System.out.println("Suka Baca " + a);
7     }
8 }
9 public static void main(String[] args) {
10     System.out.println("Sifat Orang Tua :");
11     Ortu objOrtu = new Ortu(); // memanggil objek induk
12     objOrtu.sukaMenonton("Berita"); // memanggil sifat spesifik induk
13     objOrtu.sukaMembaca("Koran"); // memanggil method dengan variabel input diarah
14 }
15 System.out.println("\n Sifat Anak :");
16 Anak objAnak = new Anak(); // memanggil objek anak
17 objAnak.sukaMenonton("Film Anime"); // memanggil sifat spesifik anak yang diturunkan induk
18 objAnak.sukaMembaca("Novel Sherlock"); // memanggil method ke induk yang kemudian diarahkan kepa downwards yang di anak
19 }
20 }
21 class Anak extends Ortu {
22     void sukaMenonton(int a, String b) {
23         System.out.println("Nonton Jam " + a + " Malam " + b);
24     }
25     void sukaMenonton(String a) { // method induk spesifik
26         System.out.println("Nonton " + a);
27     }
28     void sukaMembaca(String a) { // method induk anak bisa diarahkan
29         System.out.println("Suka Baca " + a);
30     }
31 }
32 public static void main(String[] args) {
33     System.out.println("Sifat Orang Tua :");
34     Ortu objOrtu = new Ortu(); // memanggil objek induk
35     objOrtu.sukaMenonton("Berita"); // memanggil sifat spesifik induk
36     objOrtu.sukaMembaca("Koran"); // memanggil method dengan variabel input diarah
37 }
38 System.out.println("\n Sifat Anak :");
39 Anak objAnak = new Anak(); // memanggil objek anak
40 objAnak.sukaMenonton("Film Anime"); // memanggil sifat spesifik anak yang diturunkan induk
41 objAnak.sukaMembaca("Novel Sherlock"); // memanggil method ke induk yang kemudian diarahkan kepa downwards yang di anak
42 }
43 }
```

```
Output
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Anime
Suka Baca Novel Sherlock

--- Code Execution Successful ---
```

## [Nomor Soal] Kesimpulan

### 1) Analisa

Kesimpulan dari pemrograman ini adalah kelas Anak mewarisi sifat-sifat atau method dari kelas Ortu, namun juga bisa mengubah atau menambahkan perilaku spesifiknya sendiri. Pada kelas Anak, method sukaMenonton() di-overload, sehingga ada dua versi: satu menerima parameter string (seperti di kelas Ortu), dan satu lagi menerima parameter integer dan string untuk menampilkan informasi yang lebih spesifik. Selain itu, method sukaMembaca() dapat diubah di kelas Anak untuk memberikan output yang berbeda dari kelas Ortu. Dengan pewarisan ini, kelas anak dapat memanfaatkan sifat-sifat induk, namun tetap fleksibel untuk mendefinisikan perilaku yang berbeda sesuai kebutuhan.