

## Tugas Nomor 1

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Vicky Andrian &amp; G1F024021</b>	<b>Unit 1 Kelas (Class)</b>	<b>18/09/2024</b>
<b>[Nomor 1] Identifikasi Masalah:</b>		
1.1. Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi a. atribut variabel, dan b. perilaku/ behavior untuk method!		
<b>[Nomor 1] Analisis dan Argumentasi</b>		
a.) atribut variable dari koding diatas String nama , String rambut b.) perilaku / behavior untuk method public void tidur		
<b>[Nomor 1] Penyusunan Algoritma dan Kode Program</b>		
1) Algoritma a.) Mulai b.) Deklarasi kelas c.) Deklarasi atribut variable d.) Deklarasi konstruktor e.) Deklarasi behavior dan method f.) Deklarasi main method g.) Selesai 2) Tuliskan kode program dan luaran		
<div><pre>1 public class Manusia { 2     String nama; 3     String rambut; 4 5     public Manusia() { 6         System.out.println("Kelas Manusia tanpa nama"); 7     } 8     public void tidur() { 9         System.out.println(nama + " sedang tidur."); 10    } 11    public static void main(String[] args) { 12        Manusia manusia = new Manusia(); 13    } 14 }</pre></div> <div><b>Input/Output</b>  Language Version: JDK 21.0.0 <input checked="" type="checkbox"/> Interactive  Input Arguments  Output Generated Files  Kelas Manusia tanpa nama</div>		
<b>[Nomor 1] Kesimpulan</b>		
1) Evaluasi a) Apa konsekuensi dari skenario pemrograman ini? Kodingan diatas hanya menampilkan manusia tanpa nama dan tidak berisi nama dan warna rambut		

## Tugas Nomor 2

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Vicky Andrian &amp; G1F024021</b>	<b>Unit 2 Objek</b>	<b>18/09/2024</b>

### [Nomor 2] Identifikasi Masalah:

1. Susun kembali kode di contoh 2 dengan menambahkan data ciri-ciri Anda di dalam variabel constructor!
2. Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan?

### [Nomor 2] Analisis dan Argumentasi

- 1) Saya menambahkan Warna kulit, mata dan hobi saya
- 2) Atribut : rambut hitam dan warna mata hitam  
Behavior : Saya sedang memancing

### [Nomor 2] Penyusunan Algoritma dan Kode Program

- 1.) Algoritma
  - a.) Mulai
  - b.) Deklarasi kelas Ortu
  - c.) Deklarasi Constructor Ortu
  - d.) Deklarasi Metode beraktivitas
  - e.) Deklarasi Metode keturunan
  - f.) Deklarasi main Method
  - g.) Selesai

```

1- public class Ortu {
2-     //deklarasi constructor
3-     public Ortu(String nama, String rambut, String hobi, String kulit, String mata) {
4-         //nama dan rambut adalah variabel constructor
5-         System.out.println(" Nama saya : " + nama +
6-             "\n Warna Rambut : " + rambut + "\n Hobi Saya : " + hobi + "\n Warna Kulit : "
7-             + kulit + "\n Warna mata : " + mata );
8-     }
9-     public void beraktivitas() {
10-         System.out.println("Saya sedang memancing");
11-     }
12-     public void keturunan() {
13-         System.out.println("Aku adalah keturunan dari Vicky");
14-     }
15-     public static void main (String[] args) {
16-         Ortu Satu = new Ortu("Vicky Andrian", "hitam", "Mancing", "Sawo Matang", "Hitam");
17-         Satu.beraktivitas();
18-         Ortu Dua = new Ortu("Keturunan", "hitam", "Membaca dan memancing", "Putih", "biru");
19-         Dua.keturunan();
20-         Dua.beraktivitas();
21-     }
22- }

```

Input/Output

Language Version: JDK 21.0.0

Interactive

Input Arguments

Output

Generated Files

```

Nama saya : Vicky Andrian
Warna Rambut : hitam
Hobi Saya : Mancing
Warna Kulit : Sawo Matang
Warna mata : Hitam
Saya sedang memancing
Nama saya : Keturunan
Warna Rambut : hitam
Hobi Saya : Membaca dan memancing
Warna Kulit : Putih
Warna mata : biru
Aku adalah keturunan dari Vicky
Saya sedang memancing
          
```

### [Nomor 2] Kesimpulan

- a) Evaluasi  
Apa konsekuensi dari skenario pemrograman ini?  
Didalam behaviour masih kurang fleksibel dikarenakan harus menulis sedang beraktivitas dari mana dan tidak bisa di tambah dari atribut variable

### Tugas Nomor 3

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Vicky Andrian &amp; G1F024021</b>	<b>Unit 3 Method</b>	<b>18/09/2024</b>

#### [Nomor 3] Identifikasi Masalah:

1. Analisa perbedaan deklarasi constructor, method, dan method utama!
2. Tentukan kapan Anda perlu menggunakan constructor dan method?
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

#### [Nomor 3] Analisis dan Argumentasi

- 1.) Constructor = sebagai patokan atribut objek , method = mendeskripsikan perilaku, method utama = tempat pendeskripsian
- 2.) Constructor saat ingin memulai/membuat atribut objek,method untuk mendeskripsikan perilaku berulang kali
- 3.) a.)Constructor Overloading = membuat kelas memiliki lebih dari satu constructor dengan nama yang sama tetapi dengan parameter yang berbeda, Constructor Overriding = membuat constructor di subclass dengan cara yang berbeda.  
b.)Method Overloading = Mengizinkan beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas, Method Overriding = Mengizinkan subclass untuk menyediakan implementasi yang berbeda untuk metode yang sudah ada di superclass  
c.) Method yang Mengembalikan Nilai = Memiliki tipe kembalian yang ditentukan dan menggunakan keyword return untuk mengembalikan hasil dari metode tersebut, Method Tidak Mengembalikan Nilai = Dideklarasikan dengan tipe kembalian void dan tidak menggunakan keyword return untuk mengembalikan nilai

#### [Nomor 3] Penyusunan Algoritma dan Kode Program

- 1.) Algoritma
  - a.) Mulai
  - b.) Deklarasi manusia
  - c.) Deklarasi atribut
  - d.) Deklarasi constructor
  - e.) Deklarasi metode
  - f.) Deklarasi main method
  - g.) Selesai

```

1 public class Manusia {
2     //deklarasi atribut Manusia dalam variabel
3     String nama, rambut;
4
5     //deklarasi constructor
6     public Manusia (String nama, String rambut) {
7         System.out.println(" Nama saya : " + nama +
8             "\n Warna Rambut : " + rambut);
9     }
10
11     //deklarasi method
12     void sukaNonton(String film) {
13         System.out.println(" Hobi Menonton : " + film);
14     }
15
16     //deklarasi method utama
17     public static void main (String[] args) {
18         Manusia satu = new Manusia("Putri", "hitam");
19         satu.sukaNonton("Drakor");
20     }
21 }
22

```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Output Generated Files

Nama saya : Putri  
Warna Rambut : hitam  
Hobi Menonton : Drakor

2.)

#### [Nomor 3] Kesimpulan

- 2) Analisa
  - a.) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  
Dari permasalahan diatas saya bisa membedakan apa itu method,konstruktur dan method utama
  - b.) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  
Dikarenakan perintah dari soal adalah menganalisa


#### Tugas Nomor 4

Nama & NPM	Topik:	Tanggal:
Vicky Andrean & G1F024021	Unit 1 Kelas (Class)	18/09/2024

#### [Nomor 4] Identifikasi Masalah:

- A.)Bandingkan method yang dimiliki class Anak extends Ortu dengan method di class Ortu!  
 B.)Ubahlah Contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

#### [Nomor 1] Analisis dan Argumentasi

- A.) Kelas Anak ada method sukaMenonton yang punya dua parameter, yang tidak ada di kelas Ortu, Kelas Anak juga mengoverride kedua method dari kelas Ortu, tapi sama dengan yang ada di kelas Ortu.  
 B.) Menambahkan metode bermain Mobile Legend

#### [Nomor 1] Penyusunan Algoritma dan Kode Program

- 3) Algoritma  
 A.) Mulai  
 B.) Deklarasi Kelas Ortu  
 C.) Deklarasi metode  
 D.) Deklarasi main method  
 E.) Deklarasi kelas anak  
 F.) Deklarasi metode anak  
 G.) Deklorasi main method anak  
 H.) Selesai

```

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 umum bisa diubah anak
6         System.out.println("Suka Baca " + a);
7     }
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
17 Anak objekA = new Anak(); //memanggil objek anak
18 objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
19 objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
20 objekA.sukaBermain("Mobile Legend");
21 }
22
23 class Anak extends Ortu {
24     void sukaMenonton(int a, String b) {
25         System.out.println("Nonton Jam " + a + " Malam " + b);
26     }
27     void sukaMenonton(String a) { // method induk spesifik
28         System.out.println("Nonton " + a);
29     }
30     void sukaMembaca(String a) { // method induk umum bisa diubah anak
31         System.out.println("Suka Baca " + a);
32     }
33     void sukaBermain(String game) {
34         System.out.println("Suka Bermain " + game);
35     }
36 }
37
38 public static void main(String [] args) {
39     System.out.println("Sifat Orang tua :");
40     Ortu objek0 = new Ortu(); // memanggil objek induk
41     objek0.sukaMenonton("Berita"); // memanggil sifat spesifik induk
42     objek0.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
43 }
44
45 Anak objekA = new Anak(); //memanggil objek anak
46 objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
47 objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
48 objekA.sukaBermain("Mobile Legend");
49 }
50 }
  
```

**Input/Output**

Language Version: JDK 21.0.0 Interactive

**Input Arguments**

**Output** Generated Files

```

Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
Suka Bermain Mobile Legend
  
```

Compiled and executed in 1.846 sec(s)

#### [Nomor 1] Kesimpulan

- c.) Evaluasi  
 Apa konsekuensi dari skenario pemrograman ini?  
 Terlalu panjang