

Nama & NPM	Topik:	Tanggal:
YULINDA SAMOSIR G1F024020	Kelas, Objek, Method	18 SEPTEMBER 2024

Latihan 1

[No. 1] Identifikasi Masalah:

Uraikan permasalahan dan variabel!

- 1.1. Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi
 - a. atribut variabel, dan
 - b. perilaku/ behavior untuk method!

[No. 2] Analisis dan Argumentasi

Atribut dalam kelas Manusia mewakili karakteristik yang dimiliki oleh setiap objek dari kelas tersebut. Dalam contoh atribut yang ada adalah:

String nama: Mewakili nama dari individu. Ini adalah informasi identitas yang penting.

String rambut: Mewakili warna rambut individu. Ini juga bisa menjadi ciri fisik yang relevan.

B). Perilaku dalam kelas Manusia diwakili oleh metode yang mendefinisikan tindakan yang dapat dilakukan oleh objek. Contoh metode yang ada adalah:

void sukaMendengarkanMusik(String Lagu):

Metode ini menunjukkan hobi individu dalam mendengarkan musik.

Dapat digunakan untuk mencetak hobi seseorang, dan memberikan informasi tambahan tentang karakteristik objek Manusia.

[No. 3] Penyusunan Kode Program

- a. Penyusunan
 - 1). Buatlah public class Manusia
 - 2). Masukkan deklarasi variabel, deklarasi constructor, dan deklarasi method utama
 - 3). Membuat objek manusia dan memanggil constructor.
 - 4). Jalankan Program
- b. Kode program dan luaran

```

Main.java
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             "\nWarna Rambut : " + rambut);
9     }
10
11     //deklarasi method
12     void sukaMendengarkanMusik(String lagu) {
13         System.out.println("Hobi Mendengarkan Musik: " + lagu);
14     }
15
16     //deklarasi method utama
17     public static void main( String[] args) {
18         //Membuat objek Manusia dan memanggil constructor
19         Manusia satu = new Manusia("Yulinda Samosir", "hitam");
20         satu.sukaMendengarkanMusik("lagu pop");
21     }
22 }
  
```

Output

```

java -cp /tmp/UsHUca2nB6/Manusia
Nama saya : Yulinda Samosir
Warna Rambut : hitam
Hobi Mendengarkan Musik: lagu pop

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

- a) Analisa luaran yang dihasilkan
Luaran yang dihasilkan sudah sesuai dengan yang diinginkan dari ketentuan soal.

[No. 4] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan!
Untuk Program ini saya dapat melihat dan memahami ciri-ciri umum Kelas Manusia yang dapat menjadi atribut variabel, dan perilaku/ behavior untuk method. Seperti String nama itu mewakili nama dari individu. String nama ini adalah informasi identitas yang penting. Lalu String rambut itu mewakili warna rambut individu. String rambut ini juga bisa menjadi ciri fisik yang tepat. Dan yang terakhir perilaku untuk method contohnya, void sukaMendengarkanMusik(String Lagu). Metode ini menunjukkan hobi individu dalam mendengarkan musik. Dapat digunakan untuk mencetak hobi seseorang, dan memberikan informasi tambahan tentang karakteristik objek Manusia.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Alasannya supaya mengetahui ciri ciri umum kelas manusia.

Latihan 2

[No. 1] Identifikasi Masalah:

Uraikan permasalahan dan variabel!

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

[No. 2] Analisis dan Argumentasi

Sifat(atribut) yang dapat diturunkan apabila saya memiliki keturunan, yaitu nama, karena nama yang unik atau cara penamaan keturunan saya nanti, lalu yang kedua itu warna rambut, karena genetik dapat mempengaruhi warna rambut yang akan di wariskan nantinya ke keturunan saya, dan yang terakhir itu hobi, hobi mendengarkan musik bisa di turunkan karena faktor selera musik yang sama. Kemudian untuk perilaku positif yang dapat diturunkan yaitu kreativitas dalam berpartisipasi dalam kegiatan yang berbau musik.

[No. 3] Penyusunan Variabel Operator Aritmatika

- a. Penyusunan
 - 1). Buatlah public class ortu
 - 2). Masukkan variabel nama, rambut, warnakulit, warnamata, dan bentukrambut
 - 3). Tambahkan system out println nama saya, warna rambut, warna kulit, dan bentuk rambut
 - 4). Buatlah public static void main, untuk membuat objek baru dari kelas ortu
- b. Kode program dan luaran

Main.java	Output
<pre> 1- public class Ortu { 2- // Deklarasi constructor, method khusus yang akan dipanggil saat objek kelas dibuat 3- public Ortu(String nama, String rambut, String warnakulit, String warnamata, String bentukrambut) { 4- // Variabel nama, rambut, warnakulit, warnamata, dan bentukrambut adalah parameter yang diterima oleh constructor 5- // Constructor ini akan mencetak informasi tentang nama dan karakteristik fisik lainnya 6- System.out.println("Nama saya: " + nama + 7- "\nWarna Rambut: " + rambut + 8- "\nWarna Kulit: " + warnakulit + 9- "\nWarna Mata: " + warnamata + 10- "\nBentuk Rambut: " + bentukrambut); 11- } 12- 13- // Method main, sebagai titik awal program untuk dieksekusi 14- public static void main(String[] args) { 15- // membuat objek baru dari kelas Ortu dengan menggunakan constructor yang telah didefinisikan 16- Ortu satu = new Ortu("Yulinda Samosir", "hitam", "sawo matang", "hitam", "lurus"); 17- } 18- } </pre>	<pre> java -cp ./tmp/EZLzPLBfTd/Ortu Nama saya: Yulinda Samosir Warna Rambut: hitam Warna Kulit: sawo matang Warna Mata: hitam Bentuk Rambut: lurus === Code Execution Successful === </pre>

a) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan permintaan soal.

[No. 4] Kesimpulan

1) Analisa

a) Susunlah kesimpulan!

Kesimpulannya, dari program yang saya buat, saya bisa melihat dan memahami analisa dari sifat(atribut), constructor, dan perilaku positif(behavior) yang akan diturunkan kepada keturunan saya.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Alasannya adalah supaya bisa melihat analisa dari sifat(atribut), constructor, dan perilaku positif(behavior) yang akan diturunkan kepada keturunan saya.

Latihan 3

[No. 1] Identifikasi Masalah:

Uraikan permasalahan kode program tersebut!

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. 2] Analisis dan Argumentasi

1). Perbedaan deklarasi Konstruktor, Method, dan Method Utama

Konstruktor itu sendiri memiliki nama yang sama dengan nama kelas. Contohnya public Manusia(String nama, String rambut). Sedangkan Method itu memiliki tipe pengembalian. Contohnya, void, int, String dan dapat memiliki nama yang berbeda dari nama kelas. Contoh: void sukaMendengarkanMusik(String lagu). Kemudian Method Utama itu untuk mengeksekusi program java. Contohnya, public static void main(String[] args).

2). Kapan menggunakan Konstruktor dan Method?

Konstruktor digunakan saat kita ingin menginisialisasi atribut objek ketika objek tersebut dibuat. Contohnya, saat kita membutuhkan nilai awal untuk atribut. Sedangkan Method digunakan untuk mendefinisikan perilaku objek. Jika kita ingin objek melakukan aksi tertentu setelah dibuat, kita akan menggunakan method.

3). Perbedaan

A. Konstruktor Overloading dan Overriding

Konstruktor Overloading ketika dua atau lebih metode di class yang sama memiliki nama yang sama tetapi parameternya berbeda, sedangkan Konstruktor Overriding tidak dapat dioverride karena mereka tidak memiliki tipe pengembalian, dan setiap kelas memiliki konstruktor dengan nama yang sama dengan kelasnya.

B. Method Overloading

Method Overloading mengizinkan lebih dari satu metode dengan nama yang sama tetapi dengan parameter yang berbeda baik itu tipe atau jumlah. Berguna untuk memberikan fungsionalitas yang berbeda dengan nama yang sama. Sedangkan Method Overriding mengizinkan subclass untuk memberikan implementasi khusus untuk metode yang sudah didefinisikan di superclass. Ini berguna untuk memungkinkan metode dipanggil berdasarkan objek nyata atau tipe classnya.

C). Method yang mengembalikan nilai dan yang tidak mengembalikan nilai

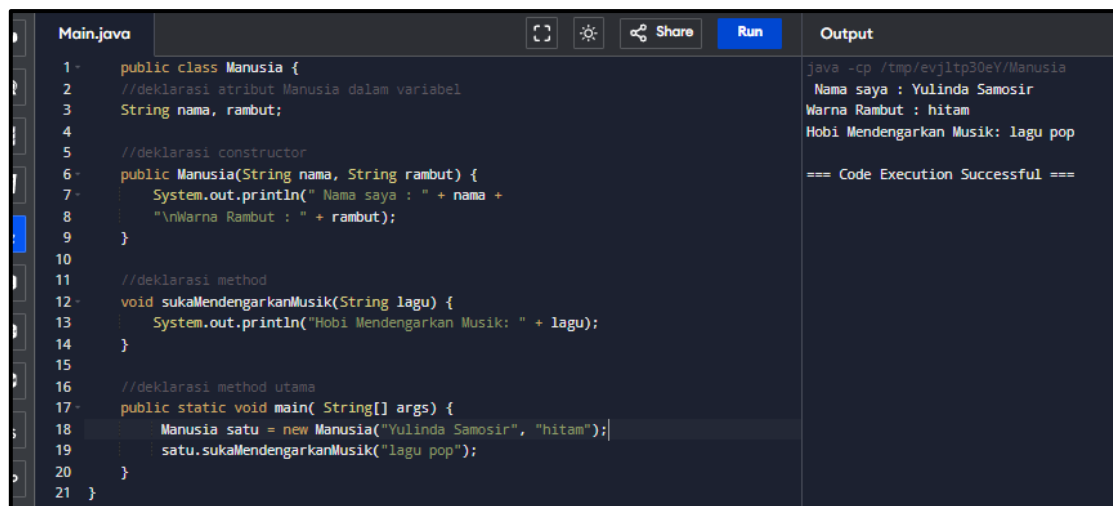
Method yang mengembalikan nilai memiliki tipe pengembalian. Misalnya int, String dan menggunakan kata kunci return untuk mengembalikan nilai.. Sedangkan Method yang tidak mengembalikan nilai biasanya digunakan untuk melakukan aksi, seperti mencetak atau mengubah atribut.

[No. 3] Penyusunan Potongan Kode Program

a. Penyusunan

- 1). Buatlah public class manusia
- 2). Masukkan string nama, rambut
- 3). Masukkan void suka mendengarkan musik
- 4). Masukkan static void main manusia satu
- 5). Jalankan Program

b. Kode program dan luaran



```
Main.java
1 public class Manusia {
2     //deklarasi atribut Manusia dalam variabel
3     String nama, rambut;
4
5     //deklarasi konstruktor
6     public Manusia(String nama, String rambut) {
7         System.out.println(" Nama saya : " + nama +
8             "\nWarna Rambut : " + rambut);
9     }
10
11    //deklarasi method
12    void sukaMendengarkanMusik(String lagu) {
13        System.out.println("Hobi Mendengarkan Musik: " + lagu);
14    }
15
16    //deklarasi method utama
17    public static void main( String[] args) {
18        Manusia satu = new Manusia("Yulinda Samosir", "hitam");
19        satu.sukaMendengarkanMusik("lagu pop");
20    }
21 }
```

Output

```
java -cp /tmp/evj1tp30eY/Manusia
Nama saya : Yulinda Samosir
Warna Rambut : hitam
Hobi Mendengarkan Musik: lagu pop

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

a) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan yang diinginkan.

[No. 4] Kesimpulan

1) Analisa

a) Susunlah kesimpulan!

Kesimpulanya, dari program diatas saya bisa melihat dan memahami perbedaan deklarasi konstruktor, method, dan method utama. Lalu saya bisa tahu kapan menggunakan konstruktor dan method, juga mengetahui dan memahami perbedaan dari konstruktor overloading dan konstruktor overriding , method

overloading, dan method overriding, serta mengetahui method yang mengembalikan nilai dan method tidak mengembalikan nilai.

- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Alasannya supaya bisa memahami perbedaan perbedaan deklarasi constructor, method, dan method utama, tahu kapan menggunakan constructor dan method, juga mengetahui dan memahami perbedaan dari constructor overloading dan constructor overriding , method overloading, dan method overriding, serta mengetahui method yang mengembalikan nilai dan method tidak mengembalikan nilai.

Latihan 4

[No. 1] Identifikasi Masalah:

Uraikan Permasalahan

- 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!

[No. 2] Analisis dan Argumentasi Koversi ke Tipe Data Lain

Class Ortu:

1. void sukaMenonton(String a) - Method ini tidak dapat diubah oleh anak (spesifik)
1. void sukaMembaca(String a) - Method ini bisa diubah oleh anak (umum).

Class Anak:

1. void sukaMenonton(int a, String b) - Method ini adalah overload dari method sukaMenonton yang ada di Ortu. Ini memungkinkan anak untuk menerima dua parameter, yaitu jam dan jenis film.
2. void sukaMenonton(String a) - Method ini mengoverride method dari Ortu, sehingga anak dapat memberikan implementasi yang berbeda.
3. void sukaMembaca(String a) - Method ini juga mengoverride method dari Ortu, memberikan implementasi baru di class Anak.

Penambahan method pada anak:

Method sukaMemasak(String Roti): Ini adalah method baru yang ditambahkan ke class Anak. Method ini tidak ada di class Ortu dan menunjukkan kebiasaan baru anak dalam berolahraga.

Dalam main, kita memanggil sukaMemasak menggunakan objek Anak untuk menunjukkan implementasinya.

[No. 3] Penyusunan Kode Program

- a. Penyusunan
 - 1). Buatlah public class Ortu, Anak
 - 2). Masukkan void suka menonton, suka membaca, suka memasak
 - 3). Masukkan public static void main suka menonton, suka membaca, suka memasak
 - 4). Tambahkan system out println sifat ortu, sifat anak
 - 5). Jalankan Program
- b. Kode program dan luaran

```

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     public static void main(String [] args) {
10        System.out.println("Sifat Orang Tua :");
11        Ortu objekO = new Ortu(); // memanggil objek induk
12        objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
13        objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat
14        diubah
15        System.out.println("\n Sifat Anak :");
16        Anak objekA = new Anak(); //memanggil objek anak
17        objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak
18        objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
19        diturunkan tanpa deklarasi ulang di anak
20    }
21
22    class Anak extends Ortu {
23        void sukaMenonton(int a, String b) {
24            System.out.println("Nonton Jam " + a + " Malam " + b);
25        }
26        void sukaMenonton(String a) { // method induk spesifik
27            System.out.println("Nonton " + a);
28        }
29    }

```

Output

```

java -cp /tmp/pVYahn9mGz/Ortu
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece

=== Code Execution Successful ===

```

```

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 umum bisa diubah anak
29     System.out.println("Suka Baca " + a);
30 }
31 void sukaMasak(String a) { // method baru
32     System.out.println("suka Masak " + a);
33 }
34
35 public static void main(String [] args) {
36     System.out.println("Sifat Orang Tua :");
37     Ortu objekO = new Ortu(); // memanggil objek induk
38     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
39     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat
40     diubah
41     System.out.println("\n Sifat Anak :");
42     Anak objekA = new Anak(); //memanggil objek anak
43     objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang
44     diturunkan induk
45     objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
46     diturunkan tanpa deklarasi ulang di anak
47     objekA.sukaMasak("Roti"); //memanggil method di anak
48 }

```

Output

```

java -cp /tmp/pVYahn9mGz/Ortu
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece

=== Code Execution Successful ===

```

a) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan program yang disusun.

[No. 4] Kesimpulan

1) Analisa

A). Susunlah kesimpulan!

Kesimpulannya saya bisa melihat dan memahami perbandingan class Anak extends Ortu dengan method di class Ortu.

b). Dasar alasan pengambilan keputusan Anda untuk kasus ini?

Supaya bisa melihat dan memahami perbandingan class Anak extends Ortu dengan method di class Ortu.