

Nama & NPM	Topik:	Tanggal:
G. Krisna Prabowo Barus G1F024062	Kelas Java	19 September 2024
[Nomor Soal] Identifikasi Masalah:		
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).		
[Nomor Soal] Analisis dan Argumentasi		
1) Uraikan rancangan solusi yang diusulkan. 2) Analisis solusi, kaitkan dengan permasalahan.		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
1) Rancang desain solusi atau algoritma 2) Tuliskan kode program dan luaran <ul style="list-style-type: none"> a) Beri komentar pada kode b) Uraikan luaran yang dihasilkan c) Screenshot/ Capture potongan kode dan hasil luaran 		
[Nomor Soal] Kesimpulan		
1) Analisa <ul style="list-style-type: none"> a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? 2) Evaluasi <ul style="list-style-type: none"> a) Apa konsekuensi dari skenario pemrograman ini? b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada) 3) Kreasi <ul style="list-style-type: none"> a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi? b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada) 		

LATIHAN 1

[No. 1] Identifikasi Masalah:

Pada Latihan ini kita di perlukan untuk memperbaiki apa yang salah dengan kode di contoh 1, dan menganalisa ciri-ciri lain Kelas Manusia yang dapat menjadi atribut variabel, dan perilaku/ behavior.

[No.1] Analisis dan Argumentasi

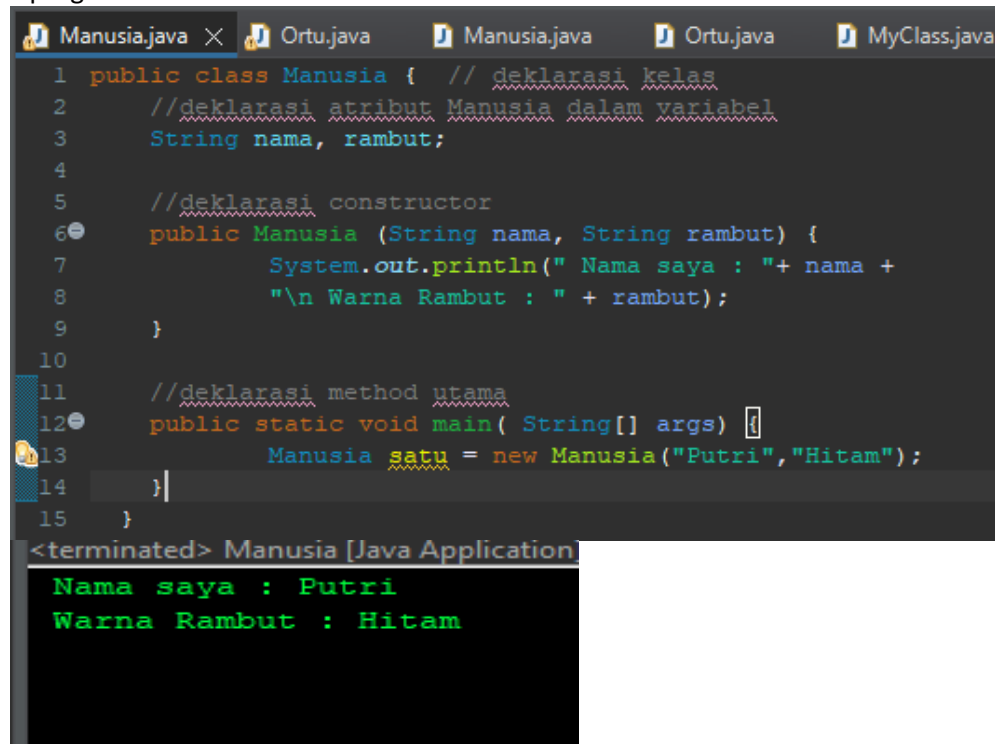
Permasalahan pada kode dapat diperbaiki dengan menyelaraskan nama kelas dan konstruktor menjadi Manusia, serta menambahkan parameter kedua (rambut) pada konstruktor agar sesuai dengan pemanggilan di metode main. Selain itu, kelas Manusia yang sederhana hanya dengan dua atribut dapat diperluas dengan menambahkan atribut seperti usia, tinggi, berat, dan jenisKelamin, serta perilaku seperti berbicara, berjalan, dan makan. Ini membuat kelas lebih lengkap dan realistis, sesuai dengan konsep OOP yang menekankan pada data dan perilaku objek.

[No.1] Penyusunan Algoritma dan Kode Program

Algoritma:

- (a) Mulai
- (b) Membuka Eclipse
- (c) Membuat Java Project baru
- (d) Membuat class baru di dalamnya
- (e) Memasukan code yang diberikan
- (f) Menganalisis dan memperbaiki code
- (g) Menekan run pada Eclipse
- (h) Selesai

Kode program dan luaran



```
Manusia.java X Ortu.java Manusia.java Ortu.java MyClass.java
1 public class Manusia { // deklarasi kelas
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 utama
12     public static void main( String[] args) {
13         Manusia satu = new Manusia("Putri","Hitam");
14     }
15 }
```

<terminated> Manusia [Java Application]

```
Nama saya : Putri
Warna Rambut : Hitam
```

Analisa luaran yang dihasilkan:

Program yang dibuat ialah sebuah program yang digunakan untuk mendemostrasikan bagaimana kelas berfungsi dalam java, dan Luaran sudah benar

No.1] Kesimpulan

Analisa

Kelas Manusia merepresentasikan entitas manusia dengan atribut seperti nama dan rambut, sementara objek adalah instansiasi dari kelas tersebut. Konstruktor digunakan untuk menginisialisasi objek dengan nilai tertentu, yang secara otomatis dipanggil saat objek dibuat di metode main(). Kesalahan awal dalam program terkait ketidakkonsistenan nama kelas dan konstruktor, serta ketidaksesuaian parameter, telah diperbaiki dengan menambahkan parameter yang sesuai, sehingga program berjalan lancar.

LATIHAN 2

[No. 2] Identifikasi Masalah:

Pada Latihan berikut ini kita diperlukan untuk mengevaluasi dan memperbaiki kode yang ada di contoh 2, kemudian apabila Ortu memiliki data variabel umur = 25 dan jenis kelamin = P (untuk Perempuan), rekomendasikan constructor dengan parameter yang baru untuk ditambahkan dalam program

[No.2] Analisis dan Argumentasi

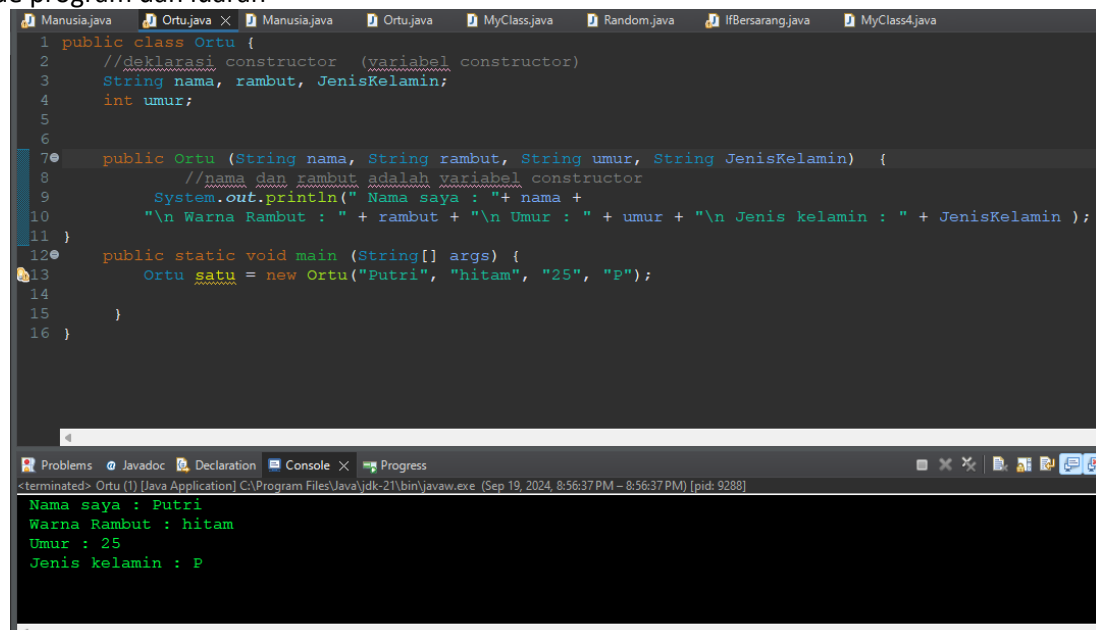
Saya mengusulkan kode ini dapat diperbaiki dengan mengubah nama konstruktor, sehingga Ortu bukan ortu, dan konstruktor perlu menerima parameter yang sesuai dengan pemanggilan di metode main(). Setelah perbaikan, konstruktor yang benar dideklarasikan dengan parameter String nama dan String rambut, yang sesuai dengan pemanggilan new Ortu("Putri", "hitam"). Selanjutnya, untuk memperluas kelas Ortu dengan atribut tambahan seperti umur dan jenisKelamin, konstruktor perlu ditambahkan untuk menerima parameter baru tersebut.

[No.2] Penyusunan Algoritma dan Kode Program

Algoritma:

- (a) Mulai
- (b) Membuka Eclipse
- (c) Membuat Java Project baru
- (d) Membuat class baru di dalamnya
- (e) Memasukkan kode yang diberikan
- (f) Melakukan perbaikan pada kode
- (g) Menambahkan constructor baru
- (h) Klik run pada Eclipse
- (i) Selesai

Kode program dan luaran



```
1 public class Ortu {
2     //deklarasi constructor (variabel constructor)
3     String nama, rambut, JenisKelamin;
4     int umur;
5
6
7     public Ortu (String nama, String rambut, String umur, String JenisKelamin) {
8         //nama dan rambut adalah variabel constructor
9         System.out.println(" Nama saya : " + nama +
10             "\n Warna Rambut : " + rambut + "\n Umur : " + umur + "\n Jenis kelamin : " + JenisKelamin );
11     }
12     public static void main (String[] args) {
13         Ortu satu = new Ortu("Putri", "hitam", "25", "P");
14     }
15 }
16 }
```

Problems Javadoc Declaration Console Progress

<terminated> Ortu (1) [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (Sep 19, 2024, 8:56:37 PM - 8:56:37 PM) [pid: 9288]

```
Nama saya : Putri
Warna Rambut : hitam
Umur : 25
Jenis kelamin : P
```

Analisa luaran yang dihasilkan:

Program ini adalah sebuah program yang digunakan untuk bagaimana atribut objek dapat diatur dan ditampilkan dalam program Java. Code dan Luaran sudah benar.

[No.2] Kesimpulan

Analisa

Kode ini bertujuan untuk memperkenalkan dan mengilustrasikan konsep dasar pemrograman berorientasi objek (OOP) dalam Java, khususnya tentang kelas, objek, atribut, dan konstruktor. Analisis kesalahan pada kode menunjukkan pentingnya kesesuaian nama konstruktor dengan nama kelas serta kesesuaian parameter yang diterima konstruktor dengan pemanggilan di metode main(). Setelah perbaikan, kode dapat mendemonstrasikan bagaimana konstruktor digunakan untuk menginisialisasi objek dengan atribut seperti nama, rambut, umur, dan jenisKelamin.

LATIHAN 3

[No. 3] Identifikasi Masalah:

Pada Latihan ini kita diperlukan untuk mengevaluasi dan memperbaiki kode tersebut, kemudian mengubah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior kita sendiri, dan terakhir 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

[No.3] Analisis dan Argumentasi

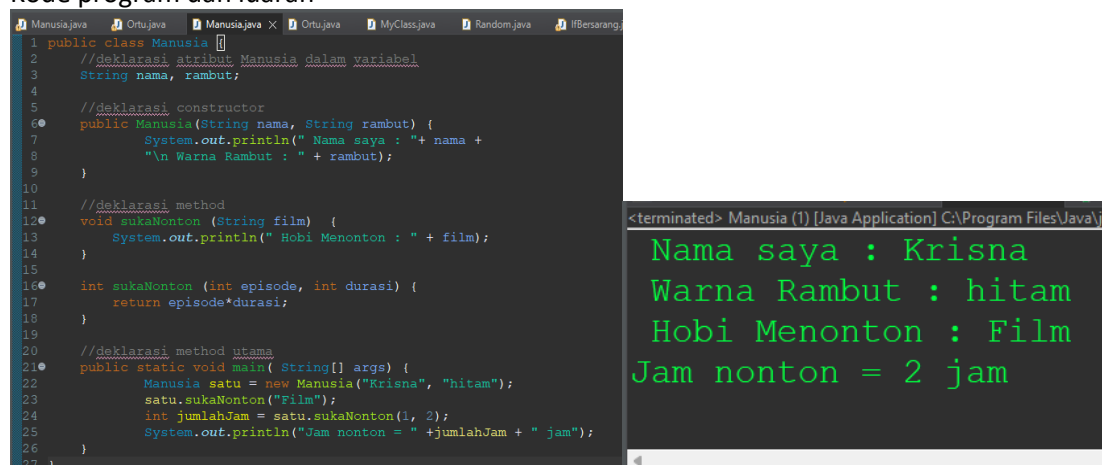
Kesalahan dalam kode awal terkait nama konstruktor yang tidak sesuai dengan nama kelas dan deklarasi metode yang tidak konsisten diperbaiki dengan menyamakan nama konstruktor dengan nama kelas Manusia dan memisahkan metode sukaNonton menjadi dua versi dengan parameter berbeda, menunjukkan method overloading. Perubahan ini juga menjelaskan method overloading dan method overriding, di mana metode dengan nama yang sama dapat ada dengan parameter berbeda dalam satu kelas, sedangkan overriding melibatkan implementasi metode yang sama dalam subclass. Selain itu, kode ini membedakan antara metode yang mengembalikan nilai dan yang tidak, dengan metode yang mengembalikan nilai seperti sukaNonton(int episode, int durasi) digunakan untuk menghitung hasil, sementara metode tanpa nilai pengembalian seperti sukaNonton(String film) hanya menampilkan informasi.

[No.3] Penyusunan Algoritma dan Kode Program

Algoritma:

- a) Mulai
- b) Membuka Eclipse
- c) Membuat Java Project baru
- d) Membuat class baru di dalamnya
- e) Memasukan code dari E-Learning
- f) Memberbaiki kode
- g) Mengubah method constructor
- h) Klik Run dan program akan mulai
- i) Selesai

Kode program dan luaran



```
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     int sukaNonton (int episode, int durasi) {
17         return episode*durasi;
18     }
19
20     //deklarasi method utama
21     public static void main( String[] args) {
22         Manusia satu = new Manusia("Krisna", "hitam");
23         satu.sukaNonton("Film");
24         int jumlahJam = satu.sukaNonton(1, 2);
25         System.out.println("Jam nonton = " +jumlahJam + " jam");
26     }
27 }
```

<terminated> Manusia (1) [Java Application] C:\Program Files\Java\j
Nama saya : Krisna
Warna Rambut : hitam
Hobi Menonton : Film
Jam nonton = 2 jam

Analisa luaran yang dihasilkan

Program ini digunakan untuk Menampilkan dan Menghitung Data yang telah dimasukkan. Code dan Luaran sudah benar.

[No.3] Kesimpulan

Analisa

Dalam analisis kode, dapat ditemukan kesalahan terkait nama konstruktor yang tidak sesuai dengan nama kelas serta deklarasi metode yang tidak konsisten. Perbaikan dilakukan dengan menyamakan nama konstruktor dan nama kelas serta memisahkan metode `sukaNonton` menjadi dua metode berbeda, memperlihatkan method overloading. Selain itu, kode ini menunjukkan perbedaan antara metode yang mengembalikan nilai dan yang tidak, di mana metode yang mengembalikan nilai seperti `sukaNonton(int episode, int durasi)` digunakan untuk menghitung hasil, sementara metode tanpa nilai pengembalian seperti `sukaNonton(String film)` hanya menampilkan informasi.

LATIHAN 4

[No. 4] Identifikasi Masalah:

Pada Latihan ini kita diperlukan untuk mengevaluasi method yang dimiliki class Anak extends Ortu dengan method di class Ortu, kemudian memasukkan kode ke dalam JDoodle dan catat waktu eksekusinya, seterusnya merekomendasikan perbaikan penulisan kode method untuk dapat mengefisienkan waktu eksekusi.

[No.4] Analisis dan Argumentasi

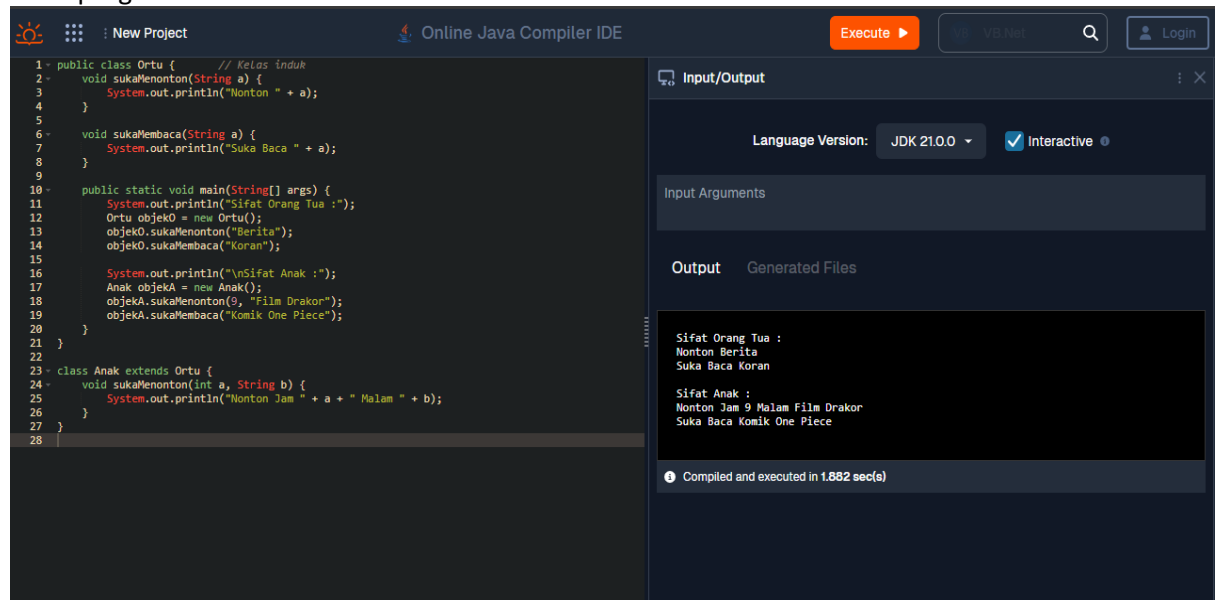
Kelas Ortu memiliki metode dasar seperti `sukaMenonton(String a)` dan `sukaMembaca(String a)`, sedangkan kelas Anak, yang mewarisi dari Ortu, menambahkan metode overload `sukaMenonton(int a, String b)` untuk menangani konteks yang berbeda. Meskipun overload ini meningkatkan fleksibilitas, penggunaan kembali metode `sukaMenonton(String a)` dan `sukaMembaca(String a)` di kelas Anak tanpa perubahan signifikan dari implementasi di kelas Ortu menunjukkan duplikasi yang tidak perlu. Waktu eksekusi kode ialah 3.07s atau 3.07 detik, agar kode berjalan lebih cepat kita perlu menghapus metode `sukaMenonton(String a)` dari kelas Anak jika tidak memerlukan perubahan dari implementasi di kelas Ortu. Ini mengurangi kompleksitas dan potensi kebingungan.

[No.4] Penyusunan Algoritma dan Kode Program

Algoritma:

- Mulai
- Membuka JDoodle
- Memasukkan kode ke JDoodle
- Memperbaiki kode agar lebih efisien
- Klik execute pada JDoodle
- Selesai

Kode program dan luaran



The screenshot shows an online Java compiler interface. On the left, the code editor contains the following Java code:

```
1 public class Ortu { // Kelas Induk
2     void sukaMenonton(String a) {
3         System.out.println("Nonton " + a);
4     }
5
6     void sukaMembaca(String a) {
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 objekO = new Ortu();
13        objekO.sukaMenonton("Berita");
14        objekO.sukaMembaca("Koran");
15
16        System.out.println("\nSifat Anak :");
17        Anak objekA = new Anak();
18        objekA.sukaMenonton(9, "Film Drakor");
19        objekA.sukaMembaca("Komik One Piece");
20    }
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 }
28
```

On the right, the 'Input/Output' panel shows the output of the program:

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

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

At the bottom of the output panel, it states: "Compiled and executed in 1.882 sec(s)".

Analisa luaran yang dihasilkan

Kode ini digunakan untuk mendemonstrasikan konsep inheritance dan method overriding dalam pemrograman berorientasi objek (OOP) dengan Java. Code dan Luaran sudah benar dan sesuai aturan.

[No.4] Kesimpulan

Analisa

Dalam kode ini, kelas Anak mewarisi metode dari kelas Ortu, menunjukkan bagaimana subclass dapat menggunakan dan memperluas fungsionalitas superclass. Metode sukaMenonton di kelas Anak menunjukkan method overloading, di mana dua versi metode dengan parameter berbeda digunakan untuk menangani situasi yang berbeda. Selain itu, method overriding terlihat dengan metode sukaMenonton dan sukaMembaca di kelas Anak yang menggantikan implementasi metode yang sama di kelas Ortu, memungkinkan subclass untuk menyesuaikan perilaku metode sesuai kebutuhan.

Refleksi

Pengalaman saya saat belajar materi kelas java ini saya lumayan mengerti apa yang dijelaskan didalam materi ini, dan akan lebih giat belajar dalam mata kuliah ini.