

Nama & NPM	Topik:	Tanggal:
Cut Fitriani G1F024079	Unit 1 Kelas (class)	17 September 2024
[Nomor Soal] Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variabel</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"> atribut variabel, dan perilaku/ behavior untuk method! <p>Jawab:</p> <p>a. Atribut Variabel</p> <p>Atribut variabel adalah data yang dimiliki oleh setiap objek dari kelas tersebut. Pada kelas Manusia, atribut variabelnya adalah:</p> <ul style="list-style-type: none"> Tipe data, String nama; Tipe data, String rambut; <p>Variabel yang menyimpan informasi mengenai setiap objek Manusia. Dalam hal ini, nama menyimpan nama manusia dan rambut menyimpan informasi tentang rambut manusia.</p> <p>b. Perilaku/Behavior untuk Method</p> <p>Perilaku atau behavior dari sebuah kelas biasanya diimplementasikan melalui method. Pada kelas Manusia, method yang ada adalah:</p> <ul style="list-style-type: none"> public Manusia() <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Jawab :</p> <p>Informasi yang saya dapatkan ada pada materi yang telah disampaikan dan juga pada chanel yt rumah ilmu raflesia.</p>		
[Nomor 2] Analisis dan Argumentasi		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Jawab: Solusi yang dihasilkan adalah dengan menambahkan kode deklarasi method utama Supaya luaran kode dapat dihasilkan.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Jawab: Kode program sudah benar tetapi saya menambahkan diakhir pada kode deklarasi metho utama agar output dapat dihasilkan.</p>		
[Nomor 3] Penyusunan Algoritma dan Kode Program		
1) Rancang desain solusi atau algoritma		

Jawab:

Algoritma :

- a) Deklarasi Kelas bernama Manusia.
- b) Mendeklarasikan dua variabel bertipe String
- c) Mencetak "Kelas Manusia tanpa nama" ke layar.
- d) Metode main, membuat objek manusia baru dengan nama satu menggunakan operator new.
- e) Output dihasilkan.
- f) Selesai.

Main.java	Output
<pre>1 public class Manusia { // deklarasi kelas 2 // deklarasi variabel 3 String nama; 4 String rambut; 5 6 // deklarasi constructor tanpa parameter 7 public Manusia() { 8 System.out.println("Kelas Manusia tanpa nama"); 9 } 10 public static void main(String[] args){ 11 Manusia satu = new Manusia (); 12 } 13 }</pre>	<pre>java -cp /tmp/ZzA3mwhg21/Manusia Kelas Manusia tanpa nama === Code Execution Successful ===</pre>

a) Beri komentar pada kode

public class Manusia { , Mendefinisikan sebuah kelas publik bernama Manusia.
String nama; dan String rambut; , Mendeklarasikan dua variabel bertipe String untuk menyimpan nama dan warna rambut dari objek Manusia.
public Manusia() { : Ini adalah konstruktor kelas Manusia.
System.out.println("Kelas Manusia tanpa nama"); : Perintah ini akan mencetak pesan "Kelas Manusia tanpa nama" ke layar.
public static void main(String[] args) { : Ini adalah metode main, titik awal eksekusi program Java.

b) Uraikan luaran yang dihasilkan

jawab: Luaran yang dihasilkan telah sesuai dengan apa yang diinginkan.

[Nomor 4] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

- Kode program yang diberikan berhasil membuat sebuah kelas Manusia sederhana. Ketika program dijalankan, sebuah objek Manusia baru akan dibuat dan pesan "Kelas Manusia tanpa nama" akan dicetak ke layar.
- Algoritma:
 - 1) Deklarasi Kelas: Mendefinisikan sebuah kelas bernama Manusia.
 - 2) Deklarasi Variabel: Mendefinisikan variabel nama dan rambut untuk menyimpan informasi tentang manusia.
 - 3) Konstruktor: Membuat konstruktor tanpa parameter yang akan mencetak pesan saat objek dibuat.
 - 4) Metode Utama: Membuat metode main sebagai titik awal eksekusi program.
 - 5) Pembuatan Objek: Membuat objek Manusia baru dengan nama bowo.
 - 6) Luaran: Ketika kode dijalankan, akan dihasilkan output: Kelas Manusia tanpa nama.

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

Jawab: Pada kasus kode program di atas, pengambilan keputusan yang mendasarinya adalah pemilihan bahasa pemrograman, pemilihan Java sebagai bahasa pemrograman menunjukkan bahwa Java adalah bahasa pemrograman berorientasi objek (OOP). Konsep OOP sangat cocok untuk memodelkan seperti “Manusia” dalam bentuk kelas dan objek.

Nama & NPM	Topik:	Tanggal:
Cut Fitriani G1F024079	Unit 2 Objek	17 September 2024
[Nomor 1] Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variabel</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>Luaran 2: Nama saya : Putri Warna Rambut : hitam</p> <p>Latihan 2: 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? Jawaban:</p> <ul style="list-style-type: none"> ➤ 2.1. Saya sudah Menyusun kode programnya dan akan ditampilkan pada nomor 3. Dengan Menambahkan variabel warna kulit dan warna mata pada kelas Ortu, memperbarui konstruktor untuk menerima parameter tambahan (warna kulit dan warna mata), dan menampilkan informasi lengkap pada output. ➤ 2.2 1) Sifat (atribut) yang diturunkan yaitu, seperti warna rambut, warna mata, dan warna kulit. 2) Constructor ini memanggil super (warna rambut, warna kulit, dan warna mata) untuk menginisialisasi atribut yang diwarisi dari kelas ortu. 3) Perilaku positif (Behavior) yang diturunkan kelas keturunan mewarisi metode dari kelas Ortu. Metode ini dapat digunakan langsung untuk memberikan implementasi khusus di kelas keturunan. <p>2)Rincikan sumber informasi yang relevan (buku / webpage) Jawab : Informasi yang saya dapatkan ada pada materi yang telah disampaikan dan juga pada chanel yt rumah ilmu raflesia.</p>		
[Nomor 2] Analisis dan Argumentasi		
<p>1) Uraikan rancangan solusi yang diusulkan. Jawab: Solusi yang diusulkan yaitu dengan menambahkan ciri-ciri saya di dalam variabel constructor dan mendeklarasikannya pada public ortu dengan menggunakan tipe data string.</p> <p>2)Analisis solusi, kaitkan dengan permasalahan. Jawab: Agar output yang diinginkan sesuai dan ditampilkan ke layar, yaitu kode program yang benar dengan menambahkan ciri-ciri saya yang akan diturunkan kepada keturunan.</p>		
[Nomor 3] Penyusunan Algoritma dan Kode Program		


1) Rancang desain solusi atau algoritma

Jawab:

Algoritma unit 2 objek :

- a) Mendeklarasikan sebuah kelas bernama Ortu
- b) Mendefinisikan sebuah constructor yang menerima empat parameter
- c) Mencetak hasil
- d) Selesai

2) Tuliskan kode program dan luaran

Main.java	  	Output 
<pre>1- public class Ortu { 2- //deklarasi constructor 3- public Ortu(String nama, String rambut, String kulit, String mata 4-) { 5- //nama dan rambut adalah variabel constructor 6- System.out.println(" Nama saya : " + nama + 7- "\n Warna Rambut : " + rambut + 8- "\n Warna kulit : " + kulit + 9- "\n Warna Mata : " + mata); 10- } 11- public static void main (String[] args) { 12- Ortu satu = new Ortu("Fitri", "hitam", "sawo matang", "coklat" 13-); 14- } 15- }</pre>		<pre>java -cp /tmp/PpJS8JzgWd/Ortu Nama saya : Fitri Warna Rambut : hitam Warna kulit : sawo matang Warna Mata : coklat === Code Execution Successful ===</pre>

a) Beri komentar pada kode

jawab: public class Ortu { , deklarasi kelas utama. Kelas ini bernama Ortu.
public Ortu(String nama, String rambut, String kulit, String mata) { , ini adalah konstruktor kelas Ortu.
System.out.println("Nama saya: " + nama);
System.out.println("Warna Rambut: " + rambut);
System.out.println("Warna kulit: " + kulit);
System.out.println("Warna Mata: " + mata); Baris-baris ini mencetak informasi tentang orang tersebut ke layar.
public static void main(String[] args) { , ini adalah metode utama (main method) dari program. Program akan dimulai dari sini.
Ortu satu = new Ortu("Fitri", "hitam", "sawo matang", "coklat"); , di sini kita membuat objek baru dari kelas Ortu dan menyimpannya dalam variabel satu.

b) Uraikan luaran yang dihasilkan

jawab: Output ini sesuai dengan informasi yang diberikan sebagai parameter saat membuat objek Ortu. Setiap baris mencetak satu atribut dari objek tersebut.

[Nomor 4] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

- Kode ini mencoba merepresentasikan konsep "orang tua" dengan atribut-atribut seperti nama, warna rambut, kulit, dan mata. Konstruktor digunakan untuk menginisialisasi objek dengan nilai-nilai spesifik. Program dirancang untuk mencetak informasi tentang objek yang telah dibuat.
- Algoritma:
Algoritma yang digunakan dalam kode ini adalah algoritma sederhana untuk membuat dan menampilkan data objek. Langkah-langkahnya adalah:
 - Definiskan kelas: Buat kelas Ortu untuk merepresentasikan objek "orang tua".
 - Tambahkan atribut: Tambahkan atribut (nama, warna rambut, dll.) ke kelas.
 - Buat konstruktor: Buat konstruktor untuk menginisialisasi objek.

- Buat metode main: Buat metode main sebagai titik masuk program.
- Buat objek: Buat objek Ortu baru.
- Tampilkan data: Panggil metode untuk menampilkan data objek ke konsol

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

Jawab: Agar output yang dihasilkan sesuai dengan pa yang diinginkan, maka dilakukannya perubahan pada kode pemrograman.

Nama & NPM	Topik:	Tanggal:
Cut Fitriani G1F024079	Unit 3 Method	17 September 2024

[Nomor 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

```
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");
    }
}
```

Luaran 3:

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

Latihan 3:

- 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

Jawaban:

1. Perbedaan constructor, method, dan method utama :

1) Constructor adalah metode khusus yang dipanggil ketika objek dari kelas ini dibuat. Tujuannya adalah untuk menginisialisasi objek. Pada kode ini, constructor seharusnya memiliki nama yang sama dengan nama kelas (Manusia). Namun, nama constructor Manusia1 tidak sesuai dengan nama kelas Manusia, sehingga constructor ini tidak akan dipanggil saat membuat objek dari kelas Manusia

- **Deklarasi:** public Manusia1(String nama, String rambut)

2) Method adalah fungsi yang terikat pada objek dan dapat dipanggil pada objek tersebut. Method ini menerima parameter (film) dan mencetak informasi tentang hobi menonton. Method ini tidak mengembalikan nilai (tipe pengembalian void).

Deklarasi: void sukaNonton(String film)

- 3) Method Utama adalah entry point untuk eksekusi program Java. Method main bersifat static, yang berarti dapat dipanggil tanpa membuat instance dari kelas. Ini adalah tempat di mana program Java mulai dieksekusi. Method ini tidak mengembalikan nilai (tipe pengembalian void).

Deklarasi: public static void main(String[] args)

2. Kapan perlu menggunakan constructor dan method

Constructor, digunakan saat ingin menginisialisasi objek dengan nilai awal atau melakukan setup yang diperlukan saat objek dibuat. Misalnya, menginisialisasi atribut objek atau mengatur kondisi awal. Constructor juga penting untuk menjamin objek dimulai dalam keadaan valid.

Method, digunakan untuk melakukan operasi atau tindakan tertentu pada objek setelah objek dibuat. Misalnya, method dapat digunakan untuk memanipulasi data, melakukan perhitungan, atau menghasilkan output. Method menyediakan fungsionalitas yang dapat dipanggil kapan saja setelah objek diinisialisasi.

3. Uraikan perbedaan berikut :

a. Constructor overloading dan overriding

Constructor overloading, yaitu mengizinkan lebih dari satu constructor dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas. Ini memberikan beberapa cara untuk menginisialisasi objek dari kelas yang sama dengan parameter yang berbeda. **Constructor overriding**, tidak ada istilah "constructor overriding" dalam Java. Constructor tidak dapat di-override karena constructor bukan bagian dari inheritance seperti method biasa. Namun, constructor dari superclass dapat dipanggil di subclass menggunakan super(), yang disebut sebagai constructor chaining.

b. Method Overloading dan Overriding

Method Overloading, yaitu mengizinkan beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas. Ini memberikan kemampuan untuk mendefinisikan metode yang melakukan tindakan yang mirip tetapi dengan cara yang berbeda.

Method Overriding, yaitu mengizinkan subclass untuk menyediakan implementasi spesifik dari metode yang sudah didefinisikan di superclass. Metode yang di-override harus memiliki nama, parameter, dan tipe pengembalian yang sama dengan metode di superclass.

c. Method yang Mengembalikan Nilai dan Method Tidak Mengembalikan Nilai

Method yang Mengembalikan Nilai:, metode yang memiliki tipe pengembalian selain void dan mengembalikan nilai dari tipe tersebut. Metode ini sering digunakan untuk melakukan perhitungan atau operasi dan memberikan hasilnya.

Method Tidak Mengembalikan Nilai, metode yang dideklarasikan dengan tipe pengembalian void, yang berarti tidak mengembalikan nilai apa pun. Metode ini biasanya digunakan untuk melakukan aksi atau prosedur yang tidak memerlukan nilai balikan.

Permasalahan: Terdapat ketidaksesuaian antara nama kelas Manusia dengan nama konstruktor Manusia1. Nama konstruktor seharusnya sama dengan nama kelas.

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

Jawab :

Informasi yang saya dapatkan ada pada materi yang telah disampaikan dan juga pada chanel yt rumah ilmu raflesia.

[Nomor 3] Analisis dan Argumentasi

1)Uraikan rancangan solusi yang diusulkan.

Jawab: Nama constructor seharusnya sama dengan nama kelas yaitu class Manusia, sehingga kode pemrograman dapat berjalan dan tidak eror.

2)Analisis solusi, kaitkan dengan permasalahan.

Jawab: Constructor memiliki nama yang sama dengan nama kelas dan tidak memiliki tipe pengembalian (void). Sehingga nama constructor harus sama dengan nama kelas yaitu class manusia

[Nomor 3] Penyusunan Algoritma dan Kode Program

1)Rancang desain solusi atau algoritma

Jawab:

Algoritma Unit 3 Method:

- Mulai program
- Membuat sebuah kelas menjadi class Manusia
- Mendeklarasikan atribut manusia ke dalam variable
- Mendeklarasikan constructor, method dan method utama
- Mencetak hasil dengan perintah System.out.println
- Selesai

2)Tuliskan 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             "\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 }
```

Output

```
java -cp /tmp/1LUT0Sp0jT/Manusia
Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor

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

a. Beri komentar pada kode

Deklarasi Variabel: Komentar ini menjelaskan bahwa variabel nama dan rambut digunakan untuk menyimpan informasi tentang nama dan warna rambut seseorang.

Konstruktor: Komentar ini menjelaskan bahwa konstruktor digunakan untuk menginisialisasi objek Manusia dengan nilai nama dan warna rambut yang diberikan.

Metode: Komentar ini menjelaskan bahwa metode sukaNonton digunakan untuk mencetak hobi menonton seseorang.

Metode Utama: Komentar ini menjelaskan bahwa metode main adalah titik awal eksekusi program.

Pembuatan Objek: Komentar ini menjelaskan bahwa baris kode ini membuat objek Manusia baru dengan nama "Putri" dan warna rambut "hitam".

b. Uraikan luaran yang dihasilkan

Jawab: Luaran yang dihasilkan sesuai yang diinginkan setelah mengubah kode program yang benar.

[Nomor 4] Kesimpulan

4) Analisa

c) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

- Kode di atas mendefinisikan sebuah kelas sederhana bernama Manusia dalam bahasa pemrograman Java. Kelas ini merepresentasikan konsep manusia dengan atribut nama dan warna rambut, serta perilaku suka menonton.
- Algoritma:
Deklarasi Kelas: Mendefinisikan kelas Manusia dengan atribut dan metode.
Konstruktor: Membuat objek Manusia dengan memberikan nilai awal untuk atribut nama dan rambut.
Metode: Mendefinisikan perilaku objek, dalam hal ini adalah metode sukaNonton.
Program Utama: Membuat objek Manusia dan memanggil metode sukaNonton.

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

Jawab: Agar kode program dapat berjalan dengan baik dan mencetak hasil ke layar sesuai dengan output yang diinginkan.

Nama & NPM	Topik:	Tanggal:
Cut Fitriani G1F024079	Unit 4 Extends	
[Nomor 1] Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variabel</p> <pre> 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 } } </pre> <p>Luaran 4: Sifat Orang Tua : Nonton Berita Suka Baca Koran</p>		

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!

Jawab:

1. Terdapat perbedaan kelas ortu dan anak saya menganalisis metode class Anak memiliki tambahan metode `sukaMenonton(int a, String b)` yang tidak ada di kelas Ortu, sehingga memperluas fungsionalitas. Kelas Anak juga menimpa (override) metode `sukaMenonton(String a)` dari kelas Ortu dengan implementasi yang sama, tetapi dapat memiliki perilaku khusus jika diperlukan. Metode `sukaMembaca(String a)` di kelas Anak tetap sama seperti di kelas Ortu, menunjukkan bahwa metode tersebut tidak diubah oleh kelas turunan. Jumlah Metode: Kelas anak memiliki lebih banyak variasi metode `sukaMenonton` dibandingkan kelas induk.
2. Saya sudah mengubahnya dengan method yang berbeda.
 - 2) Rincikan sumber informasi yang relevan (buku / webpage)

Jawab : Informasi yang relevan ada pada chanel yt Rumah Ilmu Raflesia

[Nomor 3] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.
- 2) Analisis solusi, kaitkan dengan permasalahan.

[Nomor 3] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma

Algoritma:

- Mendefinisikan kelas menjadi Ortu: Orang tua suka menonton dan membaca.
- Menambahkan objek sifat ortu, anak dan anak2
- Mencetak hasil dengan perintah
- Selesai

- 2) Tuliskan kode program dan luaran

- a) Beri komentar pada kode
- b) Uraikan luaran yang dihasilkan

Jawab: Output yang menunjukkan aktivitas yang dilakukan oleh orang tua dan kedua anak, sesuai dengan metode yang mereka panggil.

[Nomor 4] Kesimpulan

- 1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

- Kode ini menunjukkan bagaimana konsep OOP dapat digunakan untuk memodelkan hubungan antara orang tua dan anak. Kelas Ortu merepresentasikan sifat umum, sedangkan kelas Anak dan Anak2 merepresentasikan variasi dari sifat tersebut. Penggunaan overloading dan overriding memungkinkan kita untuk membuat kode yang lebih fleksibel dan dapat disesuaikan dengan situasi yang berbeda.
- Algoritma:
Definisi Kelas:

* Ortu: Memiliki metode sukaMenonton (spesifik untuk orang tua) dan sukaMembaca (umum).

* Anak: Mewarisi Ortu, memiliki sukaMenonton yang overloaded dan sukaMembaca yang overridden.

* Anak2: Mewarisi Ortu, memiliki sukaMakan yang overloaded, sukaMembaca yang overridden, dan metode baru sukaMain`.

Pembuatan Objek:

* Membuat objek dari kelas Ortu, Anak, dan Anak2.

* Panggilan Metode:

* Memanggil berbagai metode pada objek-objek tersebut untuk menunjukkan perilaku yang berbeda berdasarkan kelasnya.

b) Dengan membuat kelas Ortu sebagai kelas induk dan kelas Anak dan Anak2 sebagai kelas anak, kita sedang memodelkan hubungan hierarkis antara orang tua dan anak dalam dunia nyata. Pemanfaatan Kode Induk, kelas anak mewarisi semua metode dari kelas induknya. Ini berarti kita tidak perlu menulis ulang kode yang sama untuk setiap kelas.

```
public class Ortu {
    // Method spesifik
    void sukaMenonton(String a) {
        System.out.println("Nonton " + a);
    }

    // Method umum yang bisa diubah anak
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }

    public static void main(String[] args) {
        System.out.println("Sifat Orang Tua :");
        Ortu objek0 = new Ortu();
        objek0.sukaMenonton("Berita");
        objek0.sukaMembaca("Koran");

        System.out.println("\nSifat Anak :");
        Anak objekA = new Anak();
        objekA.sukaMenonton(9, "Film Drakor"); // Overloaded method in Anak
        objekA.sukaMembaca("Komik One Piece");

        System.out.println("\nSifat Anak2 :");
        Anak2 objekB = new Anak2();
        objekB.sukaMakan(10, "Nasi Padang campur Caviar"); // Overloaded method in Anak2
        objekB.sukaMain("Genshin Impact");
    }
}

class Anak extends Ortu {
    // Overloaded method
    void sukaMenonton(int a, String b) {
        System.out.println("Nonton Jam " + a + " Malam " + b);
    }

    // Overridden method
    @Override
    void sukaMenonton(String a) {
        System.out.println("Nonton " + a);
    }

    // Overridden method
    @Override
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }
}

class Anak2 extends Ortu {
    // Overloaded method
    void sukaMakan(int a, String b) {
        System.out.println("Makan " + a + " piring " + b);
    }

    // Overridden method
    @Override
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }

    // New method specific to Anak2
    void sukaMain(String a) {
        System.out.println("Suka main game " + a);
    }
}
```

```
java -cp /tmp/5Mq0hBwWY0/Ortu
```

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

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

```
Sifat Anak2 :
Makan 10 piring Nasi Padang campur Caviar
Suka main game Genshin Impact
```

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