

Template Lembar Kerja Individu

| | | |
|---|---------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| Nafiisah Calista Hersa Agustine_G1F024077 | Kelas Java | 16 September 2024 |
| [Nomor 1] Identifikasi Masalah: | | |
| <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>1.1. Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi</p> <p>a. atribut variabel, dan</p> <p>Atribut variable dalam kelas ini sudah ada, yaitu nama dan rambut. Nama adalah sebuah string yang menyimpan identitas individu, sedangkan rambut juga merupakan string yang menggambarkan jenis rambut seseorang. Atribut ini memberikan informasi dasar tentang karakteristik fisik dan identitas dari objek manusia.</p> <p>b. perilaku/ behavior untuk method!</p> <p>Perilaku dalam kelas manusia meliputi method seperti constructor yang digunakan untuk menginisialisasi objek dengan atribut nama dan rambut, serta method tampilkan informasi yang bertugas untuk menampilkan informasi mengenai objek tersebut. Method ini menggambarkan tindakan atau aksi yang dapat dilakukan oleh objek manusia, menjadikannya lebih interaktif dan fungsional</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Video Materi 1 tentang Kelas, Objek, Method – https://www.youtube.com/watch?v=60ldOc8m8Es</p> <p>Video Materi 2 tentang – https://www.youtube.com/watch?v=6qULMlcv-eg</p> | | |
| [Nomor 1] Analisis dan Argumentasi | | |
| <p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Rancangan solusi adalah kelas manusia, yang menggambarkan individu dengan dua atribut, nama untuk menyimpan nama dan rambut untuk jenis rambut. Kelas ini memiliki constructor untuk menginisialisasi objek dan method tampilkan informasi untuk menampilkan informasi tersebut.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi ini menjawab kebutuhan untuk mempresentasikan individu dalam system. Kelas manusia memudahkan penyimpanan data teratur dan dapat diperluas dengan atribut atau method baru. Dengan method tampilkan informasi, pengguna dapat dengan mudah melihat informasi yang ada, menjadikannya interaktif dan fungsional.</p> | | |
| [Nomor 1] Penyusunan Algoritma dan Kode Program | | |
| <p>1) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> • Mulai • Definisikan kelas manusia • Deklarasikan atribut (nama, rambut) • Buat constructor • Buat method • Akhiri <p>2) Tuliskan kode program dan luaran</p> <p>a) Beri komentar pada kode</p> <p>Kelas manusia digunakan untuk menggambarkan individu dengan dua atribut, yaitu nama dan rambut, yang disimpan sebagai private untuk menjaga keamanan data. Constructor kelas ini</p> | | |

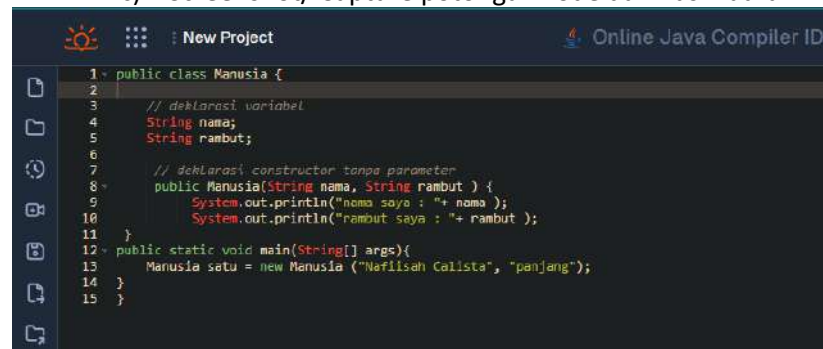
menginisialisasi objek dengan nilai nama dan rambut yang diberikan

b) Uraikan luaran yang dihasilkan

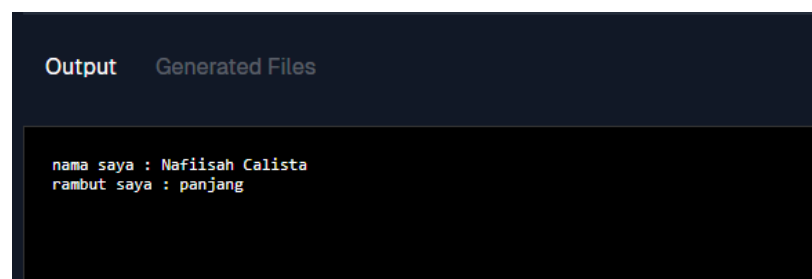
nama saya : Nafiisah Calista

rambut saya : panjang

c) Screenshot/ Capture potongan kode dan hasil luaran

A screenshot of a Java IDE window titled "New Project" and "Online Java Compiler ID". The code defines a class named "Manusia" with two attributes, "nama" and "rambut", both of type "String". It includes a constructor that takes "nama" and "rambut" as parameters and prints them. A main method is also present, which creates an instance of "Manusia" with the values "Nafiisah Calista" and "panjang".

```
1 public class Manusia {  
2  
3     // deklarasi variabel  
4     String nama;  
5     String rambut;  
6  
7     // deklarasi constructor tanpa parameter  
8     public Manusia(String nama, String rambut ) {  
9         System.out.println("nama saya : "+ nama );  
10        System.out.println("rambut saya : "+ rambut );  
11    }  
12    public static void main(String[] args){  
13        Manusia satu = new Manusia ("Nafiisah Calista", "panjang");  
14    }  
15 }
```

A screenshot of the "Output" tab in the IDE, showing the results of the program execution. It displays two lines of text: "nama saya : Nafiisah Calista" and "rambut saya : panjang".

```
Output    Generated Files  
  
nama saya : Nafiisah Calista  
rambut saya : panjang
```

[Nomor 1] Kesimpulan

1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Konsekuensi dari skenario pemrograman ini termasuk kemampuan untuk menyimpan data individu secara terstruktur, yang memudahkan pengelolaan informasi. Kelas manusia menawarkan fleksibilitas, karena dapat diperluas dengan menambahkan atribut dan method baru sesuai kebutuhan di masa depan. Interaktivitas meningkat melalui method tampilkan informasi, yang memungkinkan pengguna untuk melihat informasi dengan mudah.

Template Lembar Kerja Individu

| | | |
|--|---------------|-----------------|
| Nama & NPM | Topik: | Tanggal: |
| Nafiisah Calista Hersa Agustine_G1F024077 | Kelas Java | 16 September |
| [Nomor 2] Identifikasi Masalah: | | |
| <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>2.2. Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan?</p> <p>Jika saya memiliki keturunan, atribut yang akan diturunkan meliputi nama, warna rambut, jenis kelamin, warna kulit, tinggi badan, berat badan, dan warna mata. Constructor dapat dirancang untuk menerima semua atribut tersebut, sehingga keturunan dapat menginisialisasi ciri ciri mereka saat objek dibuat. Perilaku positif yang diharapkan termasuk kemampuan berinteraksi dengan orang lain, menunjukkan empati, berkreasi, membangun kemandirian, dan memiliki rasa tanggung jawab terhadap tindakan mereka.</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Video Materi 1 tentang Kelas, Objek, Method – https://www.youtube.com/watch?v=60ldOc8m8Es</p> <p>Video Materi 2 tentang – https://www.youtube.com/watch?v=6qULMlcv-eg</p> | | |
| [Nomor 2] Analisis dan Argumentasi | | |
| <p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Rancangan solusi yang diusulkan mencakup pembuatan kelas keturunan yang mewarisi atribut dan perilaku dari kelas ortu. Kelas ini akan memiliki constructor yang menerima atribut yang diperlukan, serta method untuk mengimplementasikan perilaku positif seperti berinteraksi, berempati, dan berkreasi. Dengan menggunakan prinsip pewarisan, kelas keturunan akan dapat menggunakan dan mengubah atribut dari kelas ortu, sekaligus menambahkan perilaku baru yang sesuai dengan karakteristik keturunan.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi ini efektif karena memanfaatkan konsep pemrograman berorientasi objek, khususnya pewarisan, untuk menciptakan hubungan yang logis antara kelas ortu dan keturunan.. dengan mengintegrasikan atribut dan perilaku positif dalam desain kelas, kita dapat memastikan bahwa keturunan tidak hanya mewarisi ciri fisik, tetapi juga nilai-nilai yang diinginkan. Pendekatan ini juga memungkinkan fleksibilitas dan pengembangan lebih lanjut, seperti penambahan metode baru untuk perilaku yang lebih spesifik, sehingga dapat disesuaikan dengan kebutuhan nyata yang mungkin muncul seiring waktu.</p> | | |
| [Nomor 2] Penyusunan Algoritma dan Kode Program | | |
| <p>1) Rancang desain solusi atau algoritma</p> <ul style="list-style-type: none"> • Definisikan kelas induk (ortu) • Definisikan kelas anak (keturunan) • Implementasikan logika program <p>2) Tuliskan kode program dan luaran</p> <p>a) Beri komentar pada kode</p> <p>Kelas ortu menyimpan atribut ciri ciri individu seperti nama, warna rambut, dan tinggi badan, serta memiliki constructor untuk menginisialisasi atribut dan metode tampilkan info untuk menampilkan informasi. Kelas keturunan, yang merupakan subclass dari ortu, menambahkan</p> | | |

atribut hobi dan talenta, serta memiliki metode untuk perilaku positif seperti berinteraksi, berempati, dan berkreasi. Kelas utama membuat objek keturunan, menginisialisasi dengan ciri ciri, dan memanggil metode untuk menampilkan informasi dan perilaku positif.

b) Uraikan luaran yang dihasilkan

Nama saya: Nafiisah Calista Hersa Agustine

Warna Rambut: hitam

Jenis Kelamin: perempuan

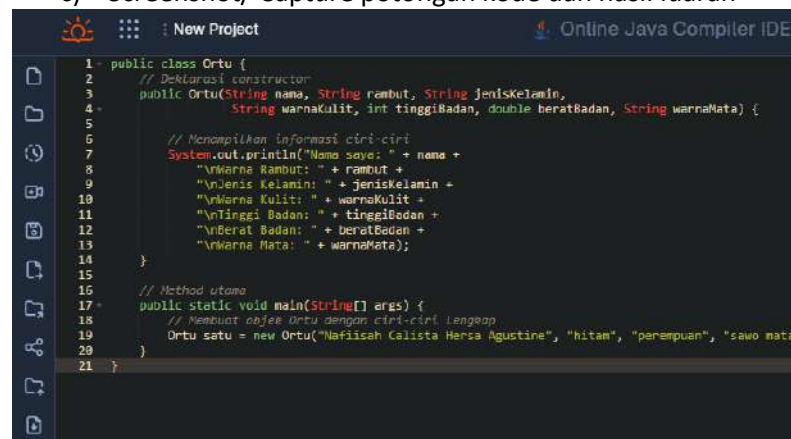
Warna Kulit: sawo matang

Tinggi Badan: 153

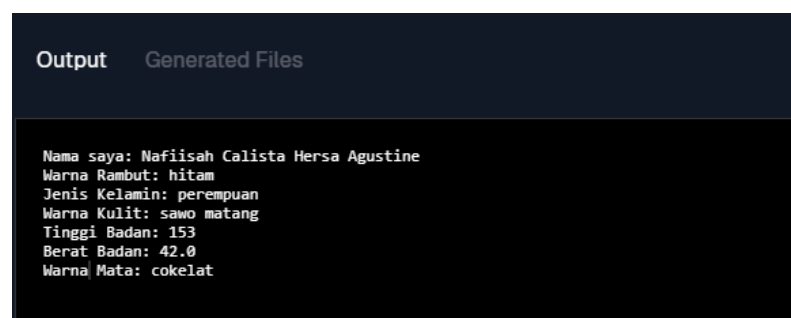
Berat Badan: 42.0

Warna Mata: coklat

c) Screenshot/ Capture potongan kode dan hasil luaran



```
1 public class Ortu {
2     // Deklarasi constructor
3     public Ortu(String nama, String rambut, String jenisKelamin,
4         String warnaKulit, int tinggiBadan, double beratBadan, String warnaMata) {
5
6         // Menampilkan informasi ciri-ciri
7         System.out.println("Nama saya: " + nama +
8             "\nWarna Rambut: " + rambut +
9             "\nJenis Kelamin: " + jenisKelamin +
10            "\nWarna Kulit: " + warnaKulit +
11            "\nTinggi Badan: " + tinggiBadan +
12            "\nBerat Badan: " + beratBadan +
13            "\nWarna Mata: " + warnaMata);
14     }
15
16     // Method utama
17     public static void main(String[] args) {
18         // Membuat objek Ortu dengan ciri-ciri lengkap
19         Ortu satu = new Ortu("Nafiisah Calista Hersa Agustine", "hitam", "perempuan", "sawo mati
20     }
21 }
```



```
Output    Generated Files

Nama saya: Nafiisah Calista Hersa Agustine
Warna Rambut: hitam
Jenis Kelamin: perempuan
Warna Kulit: sawo matang
Tinggi Badan: 153
Berat Badan: 42.0
Warna Mata: coklat
```

[Nomor 2] Kesimpulan

1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Konsekuensi dari skenario pemrograman ini meliputi efisiensi melalui pewarisan, yang mengurangi redundansi dan memudahkan pemeliharaan kode. Ini juga memberikan fleksibilitas untuk memperluas kelas keturunan dengan atribut dan perilaku baru. Selain itu, pengelolaan data menjadi lebih terstruktur, dan kode yang lebih terorganisir meningkatkan keterbacaan dan kolaborasi di antara pengembang.

Template Lembar Kerja Individu

| | | |
|--|---------------|-------------------|
| Nama & NPM | Topik: | Tanggal: |
| Nafiisah Calista Hersa Agustine_G1F024077 | Kelas Java | 16 September 2024 |

[Nomor 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

```
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");
    }
}
```

3.1. Analisa perbedaan deklarasi constructor, method, dan method utama!

Constructor : ini adalah cara untuk membuat objek baru dan mengatur nilai awalnya. Nama constructor sama dengan nama kelas dan tidak punya tipe pengembalian.

Method : ini adalah fungsi yang dilakukan oleh objek. Method bias mengembalikan nilai(seperti angka atau teks) dan bias mengambil input.

Method Utama : ini adalah bagian awal program yang dijalankan saat kita menjalankan aplikasi. Harus memiliki format tertentu dan tidak bias diganti namanya.

3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?

Constructor : digunakan ketika ingin menginisialisasi atribut dari objek pada saat objek tersebut dibuat. Misalnya, ketika ingin memberikan nilai awal untuk nama dan rambut dalam kelas manusia.

Method : digunakan untuk melakukan operasi atau aksi tertentu pada objek yang sudah ada. Misalnya, jika ingin objek manusia untuk mengekspresikan hobi mereka, akan menggunakan method sukaNonton.

3.3. Uraikan perbedaan berikut:

a) constructor overloading dan overriding

Constructor Overloading : Ini ketika memiliki beberapa constructor dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas.

Constructor Overriding : ini tidak ada karena constructor tidak dapat diwarisi dari kelas lain.

b) method overloading, dan method overriding

Method Overloading : ini ketika memiliki beberapa method dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas.

Method Overriding : ini ketika kelas turunan (subclass) mengubah cara kerja method yang sudah ada di kelas induk (superclass).

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

Method yang Mengembalikan Nilai : ini adalah method yang mengembalikan hasil (seperti angka atau teks) setelah dijalankan.

Method Tidak Mengembalikan Nilai : ini adalah method yang hanya melakukan aksi tanpa memberikan hasil kembali.

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

Video Materi 1 tentang Kelas, Objek, Method –

<https://www.youtube.com/watch?v=60IdOc8m8Es>

Video Materi 2 tentang – <https://www.youtube.com/watch?v=6qULMlcV-eg>

[Nomor 3] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.

Kami membuat kelas manusia dalam java yang menggambarkan seseorang dengan atribut nama untuk menyimpan nama dan rambut untuk warna rambut. Constructor kelas ini mengisi atribut tersebut dan mencetak informasi saat objek dibuat. Selain itu, terdapat metode suka nonton yang menampilkan hobi menonton film. Dalam metode utama, kami membuat objek manusia dan memanggil metode hobi tersebut.

- 2) Analisis solusi, kaitkan dengan permasalahan.

Solusi ini mudah dibaca dan dipahami, melindungi data dari perubahan sembarangan, serta memungkinkan penambahan informasi baru di masa depan. Dengan demikian, kelas manusia memberikan cara terstruktur untuk mendefinisikan objek, jelas dan siap untuk pengembangan lebih lanjut.

[Nomor 3] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma

- Definisikan kelas manusia
- Deklarasi atribut
- Buat constructor
- Buat metode sukaNonton
- Metode utama
- Akhiri

- 2) Tuliskan kode program dan luaran

- a) Beri komentar pada kode

Kelas manusia menyimpan informasi dengan atribut privat nama dan rambut. Constructor mengisi atribut dan mencetak informasi saat objek dibuat. Metode sukaNonton mencetak hobi menonton film. Pada metode utama, objek manusia bernama satu dibuat dengan nama "Nafiisah" dan warna rambut "hitam", lalu metode sukaNonton dipanggil untuk menampilkan hobi menonton "Drakor". Kode ini menciptakan struktur yang jelas untuk mendefinisikan individu dan perilakunya.

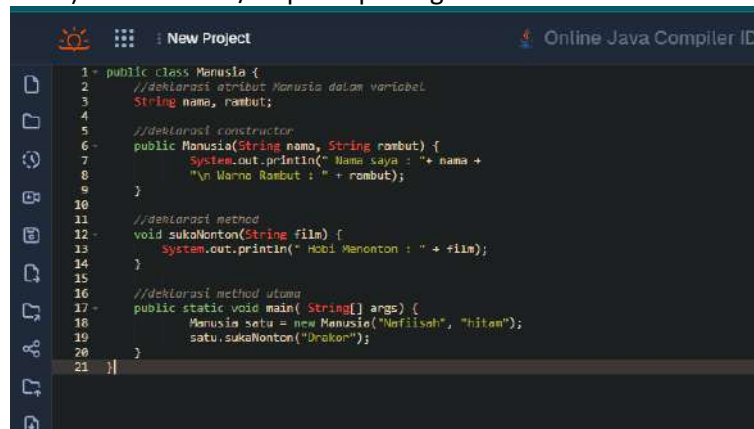
- b) Uraikan luaran yang dihasilkan

Nama saya : Nafiisah

Warna Rambut : hitam

Hobi Menonton : Drakor

- c) Screenshot/ Capture potongan kode dan hasil 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     //deklarasi method utama
17     public static void main( String[] args) {
18         Manusia satu = new Manusia("Nafiisah", "hitam");
19         satu.sukaNonton("Drakor");
20     }
21 }
```

| | |
|--|--|
| <div><div>Output</div><div>Generated Files</div></div> <div>Nama saya : Nafiisah Warna Rambut : hitam Hobi Menonton : Drakor</div> | |
| <div><div>[Nomor 3] Kesimpulan</div><div><div>1) Evaluasi</div><div>a) Apa konsekuensi dari skenario pemrograman ini?</div><div>Konsekuensi dari pemrograman ini adalah bahwa kode yang terstruktur memudahkan pemahaman, dan fitur baru bias ditambahkan dengan mudah di masa depan. Atribut privat menjaga data agar tidak bias diubah sembarangan, tetapi kita juga perlu memperhatikan penggunaan sumber daya jika program semakin kompleks.</div></div></div> | |

Template Lembar Kerja Individu

| | | |
|--|-------------------|--------------------------|
| Nama & NPM | Topik: | Tanggal: |
| Nafiisah Calista Hersa Agustine_G1F024077 | Kelas Java | 18 September 2024 |

[Nomor 4] 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 objek0 = new Ortu();        // memanggil objek induk
    objek0.sukaMenonton("Berita");    // memanggil sifat spesifik induk
    objek0.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 objek0 = new Ortu();          // memanggil objek induk
    objek0.sukaMenonton("Berita");      // memanggil sifat spesifik induk
    objek0.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
}
}
```

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

Kelas anak memiliki metode baru `sukaMenonton` yang tidak ada di kelas `Ortu`, sementara metode `sukaMenonton` dan `sukaMembaca` di `override` untuk memanggil implementasi dari kelas `Ortu`. Metode `sukaMemakan` tetap sama dan tidak diubah, menunjukkan bahwa tidak semua metode perlu dimodifikasi saat mewasi dari kelas induk.

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

Video Materi 1 tentang Kelas, Objek, Method –

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

Video Materi 2 tentang – <https://www.youtube.com/watch?v=6qULMlcv-eg>

[Nomor 4] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.

Rancangan solusi menggunakan pemrograman berorientasi objek dengan dua kelas : ortu sebagai kelas induk yang memiliki metode untuk sifat orang tua, dan anak sebagai kelas turunan yang menambahkan metode baru serta mengoverride metode dari kelas induk.

- 2) Analisis solusi, kaitkan dengan permasalahan.

Solusi ini membantu mengorganisir sifat orang tua dan anak dengan jelas, memberikan struktur yang mudah dipahami, fleksibilitas dalam menambah fungsional, dan reusabilitas kode.

Pendekatan ini juga mencerminkan hubungan nyata antara orang tua dan anak, sehingga lebih relevan dan mudah dimengerti.

[Nomor 4] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma

- Inisialisasi kelas ortu
- Inisialisasi kelas anak
- Implementasi metode main

- 2) Tuliskan kode program dan luaran

- a) Beri komentar pada kode

Kode terdiri dari dua kelas : ortu, yang mendefinisikan sifat dasar orang tua, dan anak, yang mewarisi sifat tersebut serta menambahkan metode baru untuk menonton dengan waktu.

Metode main di kelas ortu menguji kedua kelas dengan membuat objek dan memanggil metode terkait. Kelas anak mengoverride beberapa metode untuk menyesuaikan perilaku sambil memanggil metode dari kelas induk.

- b) Uraikan luaran yang dihasilkan

Sifat Orang Tua :

Nonton Berita

Suka Baca Koran

Sifat Anak :

Nonton Jam 9 Malam Film Drakor

Suka Baca Komik One Piece

Suka Makan Mie Goreng

- c) Screenshot/ Capture potongan kode dan hasil luaran

```
New Project Online Java Compiler II

1 public class Ortu {
2     void sukaMenonton(String a) {
3         System.out.println("Nonton " + a);
4     }
5     void sukaMembaca(String a) {
6         System.out.println("Suka Baca " + a);
7     }
8     void sukaMemakan(String a) {
9         System.out.println("Suka Makan " + a);
10    }
11
12    public static void main(String [] args) {
13        System.out.println("Sifat Orang Tua :");
14        Ortu objek0 = new Ortu();
15        objek0.sukaMenonton("Berita");
16        objek0.sukaMembaca("Koran");
17
18        System.out.println("\n Sifat Anak :");
19        Anak objekA = new Anak();
20        objekA.sukaMenonton(9, "Film Drakor");
21        objekA.sukaMembaca("Komik One Piece");
22        objekA.sukaMemakan("Mie Goreng");
23    }
24
25    class Anak extends Ortu {
26        void sukaMenonton(int a, String b) {
27            System.out.println("Nonton Jam " + a + " Malam " + b);
28        }
29        void sukaMenonton(String a) {
30            System.out.println("Nonton " + a);
31        }
32        void sukaMembaca(String a) {
33            System.out.println("Suka Baca " + a);
34        }
35
36        public static void main(String [] args) {
37            System.out.println("Sifat Orang Tua :");
38            Ortu objek0 = new Ortu();
39            objek0.sukaMenonton("Berita");
40            objek0.sukaMembaca("Koran");
41
42            System.out.println("\n Sifat Anak :");
43            Anak objekA = new Anak();
44            objekA.sukaMenonton(9, "Film Drakor");
45            objekA.sukaMembaca("Komik One Piece");
46            objekA.sukaMemakan("Mie Goreng");
47        }
48    }
```

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 Makan Mie Goreng
```

[Nomor 4] Kesimpulan

- 1) Evaluasi
 - a) Apa konsekuensi dari skenario pemrograman ini?

Penggunaan pemrograman berorientasi objek meningkatkan pengorganisasian dan reusabilitas kode, namun dapat menambah kompleksitas. Fleksibilitas dalam memperluas fungsionalitas juga ada, tetapi terlalu banyak override metode dapat menyebabkan kebingungan. Selain itu, kesalahan dalam pewarisan atau overrid berisiko menyebabkan perilaku yang tidak diinginkan.