

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Mellinna Husadya G1F024006</b>	<b>Unit 1 Kelas (class)</b>	<b>16 September 2024</b>

#### [No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
public class Manusia { // deklarasi kelas
    // deklarasi variabel
    String nama;
    String rambut;

    // deklarasi constructor tanpa parameter
    public Manusia() {
        System.out.println("Kelas Manusia tanpa nama");
    }
}
```

#### Latihan 1:

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

#### [No.1] Analisis dan Argumentasi

##### Analisa ciri umum pada kelas manusia

1. Nama, Rambut, Tinggi Badan, Hobi dll (variabel)
2. Makan, tidur, berjalan, Membaca dll (perilaku)

Pada kode program kali ini kami di minta untuk menganalisis ciri ciri umum pada kelas manusia, pada atribut variabel pada manusia bisa berupa nama, rambut, tinggi badan, hobi, dan pada perilaku manusia bisa berupa makan, tidur, berjalan, membaca dll.

Misalnya String nama; maksud dari variabel ini adalah menyimpan nama dari individu dalam tipe data string Dan String rambut; yaitu menyimpan warna rambut dari individu.

Pada program kali ini juga saya memperbaiki sedikit kode program dengan menambahkan public static void main(String[] args) { ini adalah kode program penting karena metode utama untuk menjalankan program dan saya juga menambahkan Manusia cantik = new Manusia(): kode ini berfungsi untuk membuat objek baru dari kelas manusia dan untuk memanggil con(): kode ini berfungsi untuk membuat objek baru dari kelas manusia dan untuk memanggil constructor manusia, yang akan mencetak pesan dengan begitu kode program yang telah diberikan akan sukses dengan luaran kelas manusia tanpa nama.

#### [No.1 ] Penyusunan Algoritma dan Kode Program

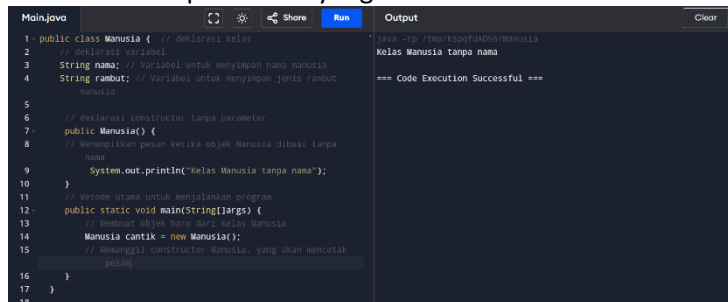
- 1) Algoritma
 

Algoritma adalah langkah-langkah penyelesaian masalah.

  - (a) Masuk ke web java compiler
  - (b) Masukkan kode program yang telah diberikan
  - (c) Tambahkan kode program untuk memperbaiki kode yang kurang agar kode sukses
  - (d) Run kan kode program
  - (e) Selesai

## 2) Kode program dan luaran

- a) Screenshot/ Capture potongan kode dan hasil luaran  
Beri komentar pada kode yang di Screenshot



```
1 public class Manusia { // deklarasi kelas
2     // deklarasi variabel
3     String nama; // variabel untuk menyimpan nama manusia
4     String rambut; // Variabel untuk menyimpan jenis rambut
5     // deklarasi constructor tanpa parameter
6     public Manusia() {
7         // Menampilkan pesan ketika objek Manusia dibuat tanpa
8         // nama
9         System.out.println("Kelas Manusia tanpa nama");
10    }
11    // kode utama untuk menjalankan program
12    public static void main(String[] args) {
13        // Membuat objek baru dari kelas Manusia
14        Manusia cantik = new Manusia();
15        // Menanggil constructor Manusia, yang akan mencetak
16        // pesan
17    }
18 }
```

Output

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

- b) Analisa luaran yang dihasilkan  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

## [No.1] Kesimpulan

### Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  
Jawab:  
Pada kode program kali ini saya memperbaiki dan menambah kode program yang telah diberikan sebelumnya dengan begitu kode program akan berjalan dengan sukses dan mengeluarkan output “kelas manusia tanpa nama” disini saya belajar apa itu variabel kelas pada manusia dan perilaku pada individu itu dengan cara membuat kode program nya
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  
Jawab:  
Disini saya banyak belajar tentang apa itu kelas pada manusia dan apa saja perilaku pada manusia dengan menggunakan program java dengan begitu disini saya menambahkan kode program tersebut agar dapat berjalan dengan sukses dan sesuai dengan permintaan soal.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Mellinna Husadya G1F024006</b>	<b>Unit 2 objek</b>	<b>16 September 2024</b>

### [No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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");
    }
}
```

#### Luaran 2:

```
Nama saya : Putri
Warna Rambut : hitam
```

#### 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?

### [No.2] Analisis dan Argumentasi

- 1) disini saya menambahkan variabel baru yang dimiliki data diri saya yaitu tinggi badan , warna mata , dan hobi
- 2) saat saya memiliki keturunan maka perilaku positif yang akan diturunkan adalah tinggi badan, warna mata, warna rambut

### [No.2 ] Penyusunan Algoritma dan Kode Program

- 3) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (f) Membuka web java compiler online
  - (g) Memulai kelas
  - (h) Masukkan kode yang telah diberikan
  - (i) Menambah variabel dengan data atau identitas diri kita
  - (j) Run kode program
  - (k) Selesai
- 4) Kode program dan luaran
  - 2) Screenshot/ Capture potongan kode dan hasil luaran  
Beri komentar pada kode yang di Screenshot

Main.java	Output
<pre>1 // Kelas Ortu yang menyimpan informasi tentang orang tua 2 public class Ortu { 3     // deklarasi constructor 4     public Ortu(String nama, String rambut, String tinggi, String       mata, String hobi) { 5         // Menggunakan parameter yang diterima untuk mencetak           informasi ke layar 6         System.out.println(" Nama saya : " + nama + 7         "\n Warna Rambut : " + rambut + 8         "\n Tinggi badan : " + tinggi + 9         "\n Warna mata : " + mata + 10        "\n Hobi : " + hobi); 11    } 12 // Metode main sebagai titik awal eksekusi program 13 public static void main (String[] args) { 14     // Membuat objek 'satu' dari kelas Ortu dengan       informasi yang diberikan 15     Ortu satu = new Ortu("Melin", "coklat", "160 cm"       ,"coklat", "membaca"); 16 } 17 }</pre>	<pre>java -cp /tmp/mIkEvrHTy0/Ortu Nama saya : Melin Warna Rambut : coklat Tinggi badan : 160 cm Warna mata : coklat Hobi : membaca  === Code Execution Successful ===</pre>

3) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

## [No.2] Kesimpulan

### Analisa

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

Jawab:

Pada program kali ini kami diminta untuk menambahkan data diri atau ciri ciri diri kedalam kode program yang akan dibuat, serta disini saya menganalisis apa saja ciri ciri yang akan di turunkan kepada keturunan saya nanti.

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

Jawab:

Disini saya mengikuti intruksi dari soal untuk membuat atau menambah ciri ciri data diri ke dalam kode program yang kita buat agar sesuai dengan permintaan soal.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Mellinna Husadya G1F024006</b>	<b>Unit 3 Method</b>	<b>16 September 2024</b>

**[No. 1] Identifikasi Masalah:**

- 2) 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");
    }
}
```

**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

**[No.1] Analisis dan Argumentasi**

- 1) Perbedaan antara constructor, method, dan method utama terletak pada fungsinya dalam kelas. Constructor digunakan untuk menginisialisasi objek dengan nilai awal saat objek dibuat, memiliki nama yang sama dengan kelas, dan tidak memiliki tipe pengembalian. Di sisi lain, method adalah fungsi yang menjalankan tugas tertentu dalam kelas, memiliki tipe pengembalian, dan dapat dipanggil secara eksplisit oleh objek. Sementara itu, method utama (main) merupakan titik awal eksekusi program Java, harus dideklarasikan dengan format `public static void main(String[] args)`, dan berfungsi untuk menjalankan logika program, bukan untuk menginisialisasi objek.

- 2) Constructor digunakan saat menginisialisasi variabel instance dengan nilai tertentu ketika objek dibuat, memastikan objek tersebut siap dan valid. Sebaliknya, method digunakan untuk mendefinisikan perilaku objek yang bisa dipanggil setelah objek dibuat, berguna untuk menjalankan tugas berulang, memproses data, atau melakukan perhitungan tanpa perlu dilakukan saat inisialisasi objek.

- 3) Uraikan perbedaan berikut

- a) constructor overloading dan overriding

Constructor Overloading terjadi ketika ada lebih dari satu constructor dalam sebuah kelas dengan nama yang sama tetapi memiliki parameter yang berbeda. Ini memungkinkan penciptaan objek dengan cara yang berbeda berdasarkan parameter yang diberikan. Sebaliknya, Constructor Overriding tidak ada dalam Java karena constructor bukan metode. Walaupun demikian, subclass dapat mengubah perilaku constructor yang diwarisi melalui pewarisan, tetapi konsep overriding seperti pada metode tidak berlaku untuk constructor.

- b) method overloading, dan method overriding

Method Overloading memungkinkan sebuah kelas memiliki beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda (tipe atau jumlah). Ini berguna untuk meningkatkan fleksibilitas dalam penggunaan metode. Di sisi lain, Method Overriding terjadi ketika subclass memiliki metode dengan nama dan parameter yang sama dengan metode di superclass. Ini memungkinkan subclass untuk memberikan implementasi yang lebih spesifik atau berbeda untuk metode yang diwarisi.

- c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

Method yang Mengembalikan Nilai memiliki tipe pengembalian yang dideklarasikan di depan nama metode (misalnya, int, String) dan menggunakan pernyataan return untuk mengembalikan hasil kepada pemanggil. Sebaliknya, Method Tidak Mengembalikan Nilai dideklarasikan dengan tipe pengembalian void, yang berarti tidak ada nilai yang dikembalikan kepada pemanggil; metode ini digunakan untuk menjalankan aksi tanpa menghasilkan output.

#### **[No.1 ] Penyusunan Algoritma dan Kode Program**

- 5) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

Misalkan algoritma memasak mi instan:

- (l) Membuka web java compiler online
- (m) Memulai kelas
- (n) Masukkan kode program
- (o) Run kode program
- (p) Selesai

- 6) Kode program dan luaran

- c) Screenshot/ Capture potongan kode dan hasil luaran  
Beri komentar pada kode yang di Screenshot

```
Main.java
1 // Kelas Manusia yang merepresentasikan data tentang seorang manusia
2 public class Manusia {
3     // Deklarasi atribut Manusia dalam variabel
4     String nama, rambut;
5
6     // Deklarasi constructor
7     public Manusia(String nama, String rambut) {
8         // Menginisialisasi atribut
9         this.nama = nama;
10        this.ambut = rambut;
11
12        // Mencetak informasi nama dan warna rambut ke konsol
13        System.out.println(" Nama saya : " + nama +
14                            "\n Warna Rambut : " + rambut);
15    }
16
17    // Deklarasi method
18    void sukaNonton(String film) {
19        // Mencetak informasi hobi menonton film ke konsol
20        System.out.println(" Hobi Menonton : " + film);
21    }
22
23    // Deklarasi method utama
24    public static void main(String[] args) {
25        // Membuat objek satu dari kelas Manusia dengan nama
26        // dan warna rambut tertentu
27        Manusia satu = new Manusia("Melin", "coklat");
28        // Memanggil method sukaNonton untuk objek satu dengan
29        // film yang dipilih
30        satu.sukaNonton("DraChin");
31    }
32 }
```

```
Output
java -cp /tmp/vRyn8tS1f/Manusia
Nama saya : Melin
Warna Rambut : coklat
Hobi Menonton : DraChin

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

d) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

## [No.1] Kesimpulan

### 1) Analisa

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

Jawab:

Program ini berhasil merepresentasikan data tentang seorang manusia dengan menyimpan informasi seperti nama dan warna rambut, serta hobi menontonnya. Dengan mendefinisikan kelas `Manusia`, kita dapat membuat objek yang menyimpan atribut-atribut ini dan menampilkan informasi tersebut ke konsol melalui constructor. Selain itu, method `sukaNonton` memungkinkan kita untuk mengekspresikan hobi menonton film, menjadikan program ini lebih interaktif. Dengan struktur yang sederhana dan jelas, kode ini memberikan pemahaman yang baik tentang bagaimana mengelola data dalam konteks objek.

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

Jawab:

Dasar pengambilan keputusan ini adalah karena saya mengikuti perintah dari soal yang telah diberikan dengan cara membuat kode program di web java compiler dengan benar dan sesuai dengan perintah yang telah diberikan hingga memunculkan output/luaran yang sukses tanpa kendala atau pun error

Nama & NPM	Topik:	Tanggal:
Mellinna Husadya G1F024006	Unit 4 Kelas (class)	16 September 2024

#### [No. 4] Identifikasi Masalah:

```

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
}

```

Luaran 4:

Sifat Orang Tua :

Nonton Berita

Suka Baca Koran

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!

#### [No.4] Analisis dan Argumentasi

Dalam kode Java ini, terdapat dua kelas yaitu `Ortu` sebagai kelas induk dan `Anak` sebagai kelas turunan. Kelas `Ortu` memiliki dua method: `sukaMenonton(String a)` yang menampilkan aktivitas menonton, serta `sukaMembaca(String a)` untuk aktivitas membaca. Sementara itu, kelas `Anak` mengoverride method dari `Ortu` dengan menambahkan method baru, yaitu `sukaBermain(String game)`, yang menunjukkan hobi bermain game. Ketika dijalankan, program ini akan mencetak sifat orang tua dan anak, termasuk aktivitas menonton, membaca, dan bermain game, sehingga memberikan gambaran yang jelas tentang berbagai hobi dan minat yang dimiliki.

#### [No.4] Penyusunan Algoritma dan Kode Program

##### 7) Algoritma

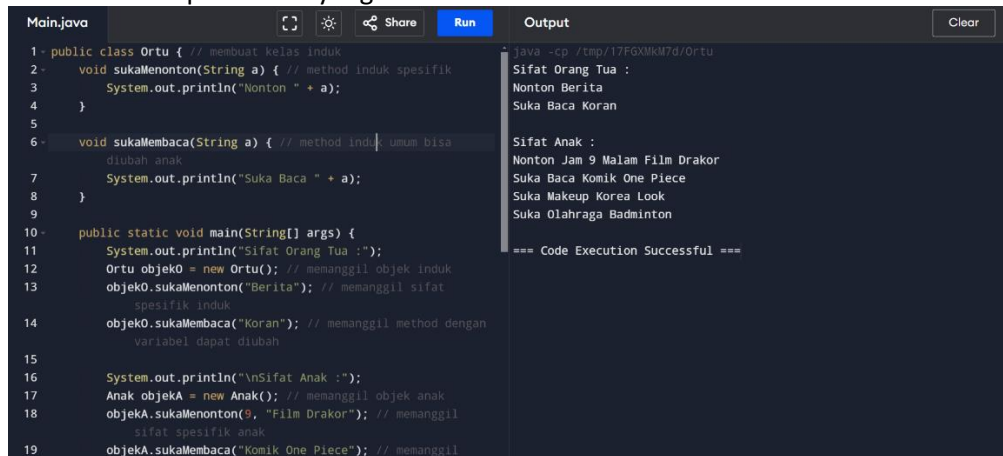
Algoritma adalah langkah-langkah penyelesaian masalah.

- (q) Masuk ke web java compiler
- (r) Mulai kelas
- (s) Masukkan kode program yang telah diberikan
- (t) Tambahkan objek anak dengan method yang berbeda
- (u) Run kode program
- (v) Selesai

##### 8) Kode program dan luaran

##### e) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

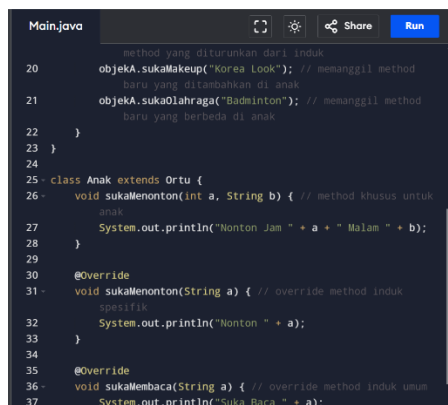


```
Main.java
1 public class Ortu { // membuat kelas induk
2     void sukaMenonton(String a) { // method induk spesifik
3         System.out.println("Nonton " + a);
4     }
5
6     void sukaMembaca(String a) { // method induk umum bisa
7         System.out.println("Suka Baca " + a);
8     }
9
10    public static void main(String[] args) {
11        System.out.println("Sifat Orang Tua :");
12        Ortu objekO = new Ortu(); // memanggil objek induk
13        objekO.sukaMenonton("Berita"); // memanggil sifat
14        objekO.sukaMembaca("Koran"); // memanggil method dengan
15        variabel dapat diubah
16
17        System.out.println("\nSifat Anak :");
18        Anak objekA = new Anak(); // memanggil objek anak
19        objekA.sukaMenonton(9, "Film Drakor"); // memanggil
20        sifat spesifik anak
21        objekA.sukaMembaca("Komik One Piece"); // memanggil
```

```
Output
java -cp /tmp/17FGXMKd7d/Ortu
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
Suka Makeup Korea Look
Suka Olahraga Badminton

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



```
Main.java
20 objekA.sukaMakeup("Korea Look"); // memanggil method
21 objekA.sukaOlahraga("Badminton"); // memanggil method
22 }
23
24
25 class Anak extends Ortu {
26     void sukaMenonton(int a, String b) { // method khusus untuk
27         anak
28         System.out.println("Nonton Jam " + a + " Malam " + b);
29     }
30
31     @Override
32     void sukaMenonton(String a) { // override method induk
33         spesifik
34         System.out.println("Nonton " + a);
35     }
36
37     @Override
38     void sukaMembaca(String a) { // override method induk umum
39         System.out.println("Suka Baca " + a);
40     }
41 }
```

```

38     }
39
40     void sukaMakeup(String a) { // method baru untuk anak
41         System.out.println("Suka Makeup " + a);
42     }
43
44     void sukaOlahraga(String a) { // method baru untuk anak
45         System.out.println("Suka Olahraga " + a);
46     }
47 }
48

```

- f) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

#### [No.4] Kesimpulan

##### Analisa

- g) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab ;

Kesimpulan dari permasalahan yang dihadapi adalah bahwa kita perlu mengembangkan sebuah program yang menggambarkan sifat dan hobi orang tua dan anak menggunakan konsep pewarisan dalam pemrograman berorientasi objek. Dalam algoritma, kita mengidentifikasi bahwa kelas *Ortu* akan berfungsi sebagai kelas induk dengan method untuk menampilkan aktivitas menonton dan membaca, sedangkan kelas *Anak* akan mewarisi method dari *Ortu* dan menambahkan method baru untuk menunjukkan hobi tambahan seperti makeup dan olahraga. Kode program yang dihasilkan dengan mengikuti struktur ini berhasil menciptakan objek dari kelas induk dan anak, serta memberikan output yang jelas mengenai aktivitas masing-masing, sehingga mencerminkan hubungan pewarisan dan penambahan fitur dalam pemrograman.

- h) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab:

Dasar alasan pengambilan keputusan dalam kasus ini berfokus pada penggunaan konsep pewarisan untuk memudahkan pengorganisasian dan pengelolaan kode. Dengan menjadikan *Ortu* sebagai kelas induk, kita dapat menghindari duplikasi kode dan memanfaatkan method yang ada di kelas induk dalam kelas anak. Penambahan method baru di kelas *Anak* menunjukkan fleksibilitas dalam mengembangkan fungsionalitas sesuai kebutuhan, serta meningkatkan keterbacaan dan pemeliharaan kode, dan disini saya juga melaksanakan permintaan dari soal yang telah diberikan sebelumnya.