

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Lina Rosalinda G1F024063	Kelas Java	11 september 2024
[Nomor 1] 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> <p>a. atribut variable</p> <p>Atribut variable pada kelas tersebut terdiri dari string nama untuk menyimpan nama manusia, dan variable string rambut menyimpan informasi tentang warna atau jenis rambut</p> <p>b. perilaku/ behavior untuk method!</p> <p>Perilaku dapat juga disebut Tindakan yang dapat dilakukan oleh objek dari kelas tersebut</p>		
[Nomor 1] Analisis dan Argumentasi		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Saya mengusulkan permasalahan ini dapat diselesaikan dengan rancangan solusi kelas manusia yaitu menyimpan dan menampilkan informasi tentang individu seperti nama dan rambut dengan menggunakan constructor yang langsung mencetak informasi ketika objek sedang dibuat dengan begitu akan mempermudah pengolahan data.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Setelah menganalisis yang saya lakukan saya menemukan bahwa kelas manusia ini efektif untuk menyimpan dan menampilkan informasi seperti nama dan rambut tadi, sehingga memenuhi kebutuhan dalam menghadapi permasalahan yang ada.</p>		
[Nomor 1] Penyusunan Algoritma dan Kode Program		
<p>1) Rancang desain solusi atau algoritma</p> <p>1. Algoritma</p> <ol style="list-style-type: none">MulaiBuat kelas dengan nama kelas manusiaTampilkan variabel nama dan rambutTampilkan constructor Manusia(String nama, String rambut)Buat metode main(String[] args)Buat objek dengan argumen (Lina Rosalinda, Hitam)Akhiri program dengan kurung kurawalAkhiri program <p>2) Tuliskan kode program dan luaran</p> <p>1. Kode program</p>		

```
Main.java
1 public class Manusia { //deklarasi kelas
2     //deklarasi variabel
3     String nama;
4     String rambut;
5
6     //deklarasi konstruktor tanpa parameter
7     public Manusia( String nama , String rambut){
8         System.out.println("Nama saya : " + nama + "\n" rambut : " +
          rambut);
9     }
10    public static void main (String[] args){
11        Manusia satu = new Manusia ("Lina Rosalinda","Hitam");
12    }
13 }
```

2. Luaran

```
Output

java -cp /tmp/kSOjHdRErC/Manusia
Nama saya : Lina Rosalinda
    rambut : Hitam

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

Luaran sudah sesuai dengan program yang disusun karena hasil yang dihasilkan oleh program telah sesuai dengan apa yang diharapkan. Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sesuai dengan spesifikasi yang diminta

[Nomor 1] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Dari permasalahan yang ada yaitu pengelolaan informasi dasar individu yaitu nama dan rambut, algoritma yang saya buat sudah sesuai dengan menunjukkan Langkah-langkah untuk mendefinisikan kelas tersebut, konstruktor menerima parameter nama dan rambut. Solusi ini tidak hanya memberikan cara yang cukup sederhana untuk pemahaman saya.
- b) Apakah dasar alasan pengambilan keputusan Ada untuk kasus ini?
Saya memilih menggunakan kelas manusia karena strukturnya yang jelas memuat kebutuhan yang jelas untuk menyimpan dan menampilkan data dengan cara sederhana.

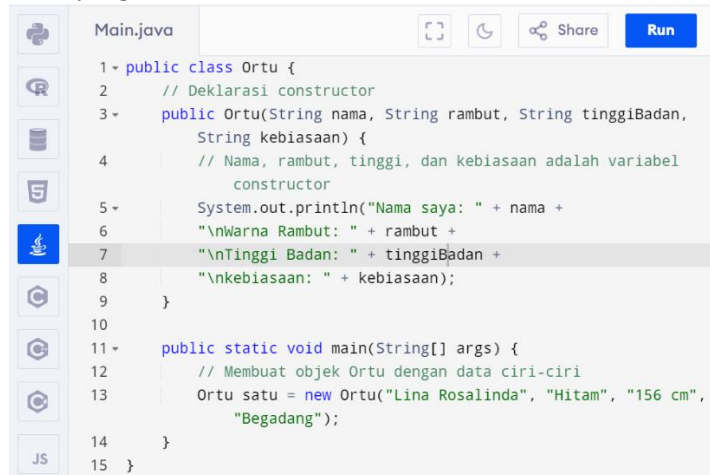
Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Lina Rosalinda G1F024063	Kelas Java	11 september 2024
[Nomor 2] 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! Pada kode ini constructor akan menerima empat parameter yaitu nama, rambut, tinggi badan dan kebiasaan. 2.2. Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan? Apabila saya memiliki keturunan sifat(atribut) yang mungkin akan saya turunkan yaitu seperti rambut ataupun jenis kelamin dan hobi dari kelas induk, sambil menambahkan atribut baru dan mengoverride metode untuk mencerminkan karakteristik unik mereka.</p>		
[Nomor 2] Analisis dan Argumentasi		
<p>1) Uraikan rancangan solusi yang diusulkan. Rancangan solusi yang saya usulkan yaitu kelas induk (Ortu) dengan atribut dan metode dasar, serta kelas turunan (Anak) yang mewarisi dan memperluas atribut serta perilaku, sehingga memungkinkan penggunaan kembali dan modifikasi yang lebih efisien.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan. Setelah melakukan analisis yang menunjukkan bahwa dengan menerapkan konsep pewarisan dalam pemrograman berorientasi objek, kita dapat mengatasi masalah duplikasi kode dan meningkatkan keterbacaan, sementara hubungan antara orang tua dan anak dalam kode mencerminkan struktur nyata dalam kehidupan sehari-hari.</p>		
[Nomor 2] Penyusunan Algoritma dan Kode Program		
<p>3) Rancang desain solusi atau algoritma</p> <ol style="list-style-type: none">Algoritma<ol style="list-style-type: none">MulaiBuat kelas dengan nama 'ortu'Deklarasikan constructor dengan parameter String nama, String rambut, String tinggi badan, dan String kebiasaan.Buat metode main dengan parameter public static void main(String[] args)		

- e. Masukkan nilai "Lina Rosalinda" untuk nama, "hitam" untuk rambut, "156" untuk tinggi badan dan "begadang" untuk kebiasaan.
- f. Akhiri dengan tanda kurung kurawal
- g. Akhiri program.

4) Tuliskan kode program dan luaran

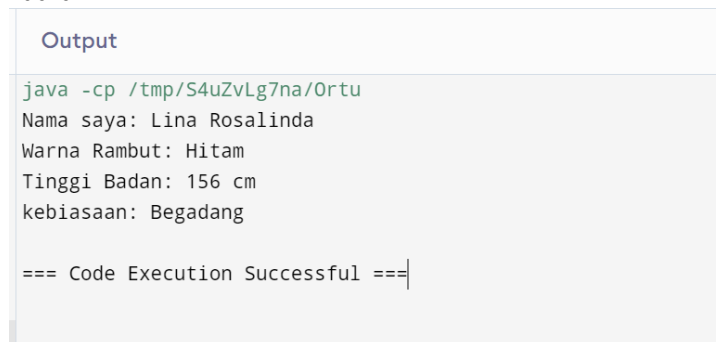
1. Kode program



```

Main.java
1 public class Ortu {
2     // Deklarasi constructor
3     public Ortu(String nama, String rambut, String tinggiBadan,
4         String kebiasaan) {
5         // Nama, rambut, tinggi, dan kebiasaan adalah variabel
6         // constructor
7         System.out.println("Nama saya: " + nama +
8             "\nWarna Rambut: " + rambut +
9             "\nTinggi Badan: " + tinggiBadan +
10            "\nkebiasaan: " + kebiasaan);
11    }
12
13    public static void main(String[] args) {
14        // Membuat objek Ortu dengan data ciri-ciri
15        Ortu satu = new Ortu("Lina Rosalinda", "Hitam", "156 cm",
16            "Begadang");
17    }
18 }
  
```

2. Luaran



```

Output

java -cp /tmp/S4uZvLg7na/Ortu
Nama saya: Lina Rosalinda
Warna Rambut: Hitam
Tinggi Badan: 156 cm
kebiasaan: Begadang

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

Luaran sudah sesuai dengan program yang disusun karena hasil yang dihasilkan oleh program telah sesuai dengan apa yang diharapkan. Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sesuai dengan spesifikasi yang diminta

[Nomor 2] Kesimpulan

2) Analisa

- c) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Berdasarkan permasalahan, algoritma dan kode program saya dapat menyimpulkan bahwa dengan mendefinisikan kelas ortu yang menggunakan constructor untuk menginisialisasi atribut yang diinginkan. Algoritma yang disusun pun mendukung struktur ini menunjukkan Langkah-langkah pembuatan objek dan penyajian informasi.
- d) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Saya memilih keputusan ini karena memungkinkan perorganisasian yang lebih baik sehingga mempermudah pemeliharaan dan pengembangan lebih lanjut. Serta dapat mempresentasikan karakteristik objek secara terstruktur dan jelas.

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Lina Rosalinda G1F024063	Kelas Java	11 september 2024

[Nomor 3] Identifikasi Masalah:

2) 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!

Constructor digunakan untuk menginisialisasi objek dengan atribut, method digunakan untuk proses, dan method utama digunakan untuk mengeksekusi program dan harus dijalankan dengan parameter public static void main (String[] args).

3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?

Constructor digunakan pada saat ingin membuat kelas dan menginisialisasi atribut tersebut. Sedangkan method digunakan pada saat ingin melakukan tindakan setelah objek diciptakan dengan melakukan operasi tertentu.

3.3. Uraikan perbedaan berikut:

a) constructor overloading dan overriding

Constructor overloading memungkinkan beberapa constructor dengan beberapa parameter berbeda dalam satu kelas. Sedangkan overloading memungkinkan subclass mengganti implementasi method dari superclass.

b) method overloading, dan method overriding

Method overloading memungkinkan beberapa method dengan nama yang sama tapi parameter berbeda dalam satu kelas. Sedangkan method overloading memungkinkan subclass memberikan implementasi baru dengan nama dan parameter yang sama.

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

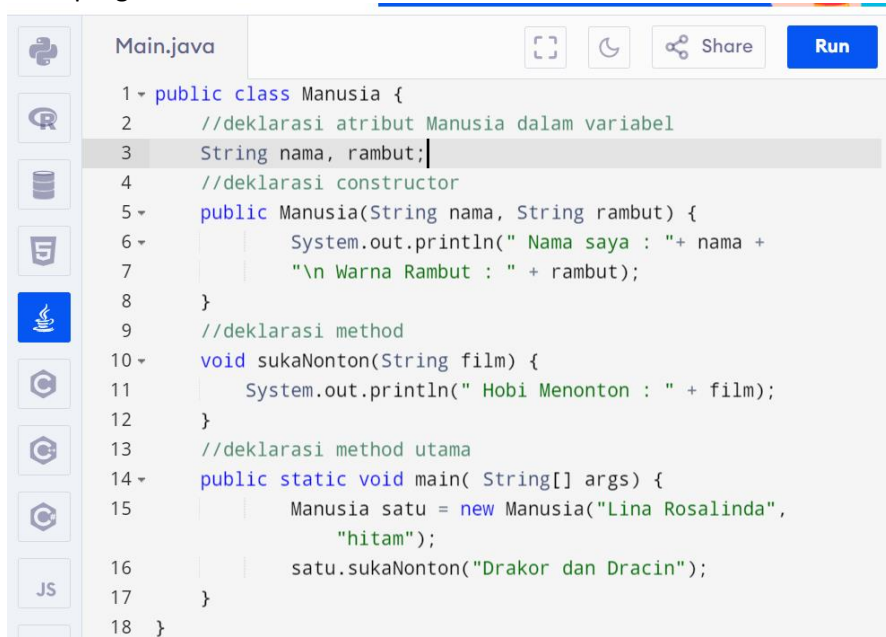
Method mengembalikan nilai memiliki tipe pengembalian tertentu dengan menggunakan keyword return. Sedangkan method tidak mengembalikan nilai mendeklarasikan nilai dengan void bukan return.

[Nomor 3] Analisis dan Argumentasi

- 5) Uraikan rancangan solusi yang diusulkan.
Saya mengusulkan solusi ini dengan cara melakukan perbaikan kode dengan mengganti nama constructor agar sesuai serta memperbanyak method untuk meningkatkan fungsi dan keterbacaan output.
- 6) Analisis solusi, kaitkan dengan permasalahan.
Karena solusi ini menunjukkan bahwa dengan memperbaiki nama constructor dan mengganti nama constructor maka kode akan bebas dari error yang tidak diinginkan sehingga mengatasi permasalahan yang disebabkan oleh ketidaksesuaian nama serta meningkatkan ketidakmampuan objek untuk mencernakan karakter manusia yang diinginkan.

[Nomor 3] Penyusunan Algoritma dan Kode Program

- 3) Rancang desain solusi atau algoritma
 - a. Mulai
 - b. Buat kelas 'manusia'
 - c. Deklarasikan atribut nama dan rambut dengan tipe data String
 - d. Buat constructor yang menerima parameter nama dan rambut
 - e. Cetak informasi nama dan rambut dengan System out println
 - f. Deklarasikan method hobi menonton dengan menerima satu parameter yaitu (film)
 - g. Deklarasikan method utama dengan nama "Lina Rosalinda" dan rambut "hitam".
 - h. Memanggil method hobi menonton dengan parameter "Dracin dan Drakor".
 - i. Akhiri program dengan kurung kurawal
 - j. Akhiri program.
- 4) Tuliskan kode program dan luaran
 1. Kode program



```
1 public class Manusia {
2     //deklarasi atribut Manusia dalam variabel
3     String nama, rambut;
4     //deklarasi constructor
5     public Manusia(String nama, String rambut) {
6         System.out.println(" Nama saya : " + nama +
7             "\n Warna Rambut : " + rambut);
8     }
9     //deklarasi method
10    void sukaNonton(String film) {
11        System.out.println(" Hobi Menonton : " + film);
12    }
13    //deklarasi method utama
14    public static void main( String[] args) {
15        Manusia satu = new Manusia("Lina Rosalinda",
16            "hitam");
17        satu.sukaNonton("Drakor dan Dracin");
18    }
19 }
```

2. Luaran

Output

```
java -cp /tmp/dPa7aXm0Jl/Manusia  
Nama saya : Lina Rosalinda  
Warna Rambut : hitam  
Hobi Menonton : Drakor dan Dracin  
  
=== Code Execution Successful ===
```

Luaran sudah sesuai dengan program yang disusun karena hasil yang dihasilkan oleh program telah sesuai dengan apa yang diharapkan. Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sesuai dengan spesifikasi yang diminta

[Nomor 3] Kesimpulan

3) Analisa

- e) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Pada materi ini saya dapat menyimpulkan bahwa perbaikan kode program dengan memperbaiki nama constructor dan menerapkan algoritma yang terstruktur memungkinkan objek manusia berfungsi dengan baik untuk mencetak informasi dan hobi sehingga mencapai apa yang diinginkan.
- f) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Saya memilih keputusan ini karena pengambilan keputusan dalam hal ini adalah untuk memastikan kode berfungsi dengan baik dan bebas error serta menerapkan praktik pemrograman yang mengutamakan struktur dan konsistensi.

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Lina Rosalinda G1F024063	Kelas Java	11 september 2024

[Nomor 4] Identifikasi Masalah:

1Uraikan 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
} }
}
```

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!

[Nomor 4] Analisis dan Argumentasi

1. Uraikan rancangan solusi yang diusulkan.

Rancangan solusi yang saya usulkan yaitu konsep pewarisan dalam pemrograman berorientasi objek, di mana anak mewarisi sifat dari induk sambil menambahkan atau mengubah perilaku tertentu. Ini memberikan fleksibilitas dalam pengembangan aplikasi dan memudahkan pengelolaan kode dengan struktur yang jelas. Dengan demikian, solusi ini menjawab kebutuhan untuk menggambarkan hubungan antara objek secara efektif dan efisien.

2. Analisis solusi, kaitkan dengan permasalahan.

Saya memilih solusi ini karena solusi yang saya usulkan secara efektif menjawab permasalahan dengan menerapkan prinsip-prinsip OOP seperti pewarisan, polimorfisme, dan modularitas. Dengan cara ini, program dapat menciptakan representasi yang jelas dan terstruktur tentang hubungan antara orang tua dan anak, serta memungkinkan pengembangan lebih lanjut dengan cara yang terorganisir.

[Nomor 4] Penyusunan Algoritma dan Kode Program

5) Rancang desain solusi atau algoritma

1. Algoritma

- a. Mulai
- b. Buat kelas dengan nama 'ortu'
- c. Deklarasi method ortu suka nonton dan suka baca dengan tipe data string
- d. Deklarasikan kelas anak yang mewarisi kelas dari ortu
- e. Deklarasikan method utama dalam kelas ortu
- f. Panggil duka nonton dan suka baca dengan parameter "Drakor dan Novel"
- g. Panggil method anak
- h. Akhiri program dengan kurung kurawal
- i. Akhiri program

6) Tuliskan kode program dan luaran

1. Kode program

```
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     void sukaMembaca(String a) { // method induk umum bisa
        diubah anak
6         System.out.println("Suka Baca " + a);
7     }
8     public static void main(String [] args) {
9         System.out.println("Sifat Orang Tua :");
10        Ortu objekO = new Ortu(); // memanggil objek induk
11        objekO.sukaMenonton("Drakor"); // memanggil sifat
            spesifik induk
12        objekO.sukaMembaca("Novel"); // memanggil method dengan
            variabel dapat diubah
13
14        System.out.println("\n Sifat Anak :");
15        Anak objekA = new Anak(); //memanggil objek anak
16        objekA.sukaMenonton(9, "Film Drakor"); //memanggil
            sifat spesifik anak yang diturunkan induk
17        objekA.sukaMembaca("Novel Online"); //memanggil method ke
            induk yang otomatis diturunkan tanpa deklarasi ulang di
            anak
18    } }
19 class Anak extends Ortu {
20     void sukaMenonton(int a, String b) {
21         System.out.println("Nonton Jam " + a + " Malam " + b);
22     }
23     void sukaMenonton(String a) { // method induk
            spesifik
24         System.out.println("Nonton " + a);
25     }
26     void sukaMembaca(String a) { // method induk umum bisa
            diubah anak
27         System.out.println("Suka Baca " + a);
28     }
29
30     public static void main(String [] args) {
31         System.out.println("Sifat Orang Tua :");
32         Ortu objekO = new Ortu(); // memanggil objek induk
33         objekO.sukaMenonton("Berita"); // memanggil sifat
            spesifik induk
34         objekO.sukaMembaca("Koran"); // memanggil method dengan
            variabel dapat diubah
35
36         System.out.println("\n Sifat Anak :");
37         Anak objekA = new Anak(); //memanggil objek anak
38         objekA.sukaMenonton(9, "Film Drakor"); //memanggil
            sifat spesifik anak yang diturunkan induk
39         objekA.sukaMembaca("Komik One Piece"); //memanggil method
            ke induk yang otomatis diturunkan tanpa deklarasi ulang
            di anak
40     }
```

2. Luaran

Output

```
java -cp /tmp/IIRyHtAbDY/Ortu
Sifat Orang Tua :
Nonton Drakor
Suka Baca Novel

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Novel Online

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

Luaran sudah sesuai dengan program yang disusun karena hasil yang dihasilkan oleh program telah sesuai dengan apa yang diharapkan. Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sesuai dengan spesifikasi yang diminta

[Nomor 4] Kesimpulan

4) Analisa

- g) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Kesimpulan dari permasalahan yang dihadapi yaitu dengan menggunakan konsep pewarisan dalam pemrograman berorientasi objek, kita dapat menciptakan struktur yang jelas dan terorganisir untuk menggambarkan hubungan antara orang tua dan anak. Algoritma yang disusun menggambarkan langkah-langkah logis dalam mendefinisikan kelas induk (Ortu) dan kelas turunan (Anak), serta bagaimana metode dapat diwariskan dan diubah. Kode program yang diimplementasikan membuktikan bahwa sifat dan perilaku orang tua dapat diakses dan dimodifikasi oleh anak, menciptakan fleksibilitas dalam pengembangan perangkat lunak.
- h) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Saya memilih keputusan ini karena pengambilan keputusan untuk menggunakan pewarisan dalam kasus ini adalah untuk mengurangi duplikasi kode dan meningkatkan keterbacaan serta pemeliharaan. Dengan mendefinisikan sifat umum dalam kelas induk dan memperbolehkan kelas turunan untuk mengoverride atau menambahkan metode baru, kita dapat menciptakan sistem yang lebih efisien dan modular. Hal ini juga memungkinkan penggunaan kembali kode yang sudah ada, sehingga mempercepat pengembangan dan mengurangi potensi kesalahan.

Refleksi

Pada minggu ini, saya belajar tentang konsep pemrograman berorientasi objek (OOP) dalam Java, yaitu mengenai kelas, objek, constructor, dan pewarisan. Melalui beberapa contoh, saya memahami cara mendeklarasikan kelas dengan atribut dan method, serta bagaimana objek dapat berinteraksi dengan informasi tersebut. Tantangan utama yang saya hadapi adalah memastikan kesesuaian antara nama kelas dan constructor, yang berpotensi menyebabkan error saat pembuatan objek. Namun, dengan mempraktikkan berbagai konsep OOP dan mulai berdiskusi dengan teman untuk

membantu saya , saya dapat lebih memahami struktur dan logika pemrograman yang baik, serta pentingnya menghindari duplikasi kode. Pengalaman ini sangat berharga dalam membangun dasar yang kuat untuk pengembangan pembelajaran saya.