

**Template Lembar Kerja Individu dan Kelompok**

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>[Nomor Soal] Identifikasi Masalah:</b>		
<ol style="list-style-type: none"><li>1) Uraikan permasalahan dan variabel</li><li>2) Rincikan sumber informasi yang relevan (buku / webpage)</li><li>3) Uraikan rancangan solusi yang diusulkan (jika ada).</li><li>4) Analisis susunan solusi, parameter solusi (jika ada).</li></ol>		
<b>[Nomor Soal] Analisis dan Argumentasi</b>		
<ol style="list-style-type: none"><li>1) Uraikan rancangan solusi yang diusulkan.</li><li>2) Analisis solusi, kaitkan dengan permasalahan.</li></ol>		
<b>[Nomor Soal] Penyusunan Algoritma dan Kode Program</b>		
<ol style="list-style-type: none"><li>1) Rancang desain solusi atau algoritma</li><li>2) Tuliskan kode program dan luaran<ol style="list-style-type: none"><li>a) Beri komentar pada kode</li><li>b) Uraikan luaran yang dihasilkan</li><li>c) Screenshot/ Capture potongan kode dan hasil luaran</li></ol></li></ol>		
<b>[Nomor Soal] Kesimpulan</b>		
<ol style="list-style-type: none"><li>1) Analisa<ol style="list-style-type: none"><li>a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!</li><li>b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?</li></ol></li><li>2) Evaluasi<ol style="list-style-type: none"><li>a) Apa konsekuensi dari skenario pemrograman ini?</li><li>b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)</li></ol></li><li>3) Kreasi<ol style="list-style-type: none"><li>a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?</li><li>b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)</li></ol></li></ol>		

Luaran 1: Contoh Jawaban:

Nama & NPM	Topik:	Tanggal:
IVANALY DEES TUMANGGER G1F024033	CLASS	16 SEPTEMBER 2024

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

1) Uraikan permasalahan dan variabel

Contoh: 

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

//deklarasi constructor
public Manusia1 (String nama) {
    System.out.println(" Nama saya : "+ nama +
    "\n Warna Rambut : " + rambut);
}

//deklarasi method utama
public static void main( String[] args) {
    Manusia1 satu = new Manusia1("Putri", "hitam");
}
}
```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
The constructor Manusia1(String, String) is undefined  
at Manusia1.main(Manusia1.java:13)

**Latihan 1:**

1.1. Perbaiki pesan kesalahan Contoh 1!

1.2. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi

- atribut variabel, dan
- perilaku/ behavior

1.1. Perbaiki pesan kesalahan Contoh 1!

Pada soal masih ada pesan kesalahan Pada nama constructor pada kelas manusia1 yang tidak sesuai .karena seharusnya constructor di dalam kelas memiliki nama yang sama persis dengan nama kelas itu sendiri.

Atau

Diketahui dari soal : variabel nya yaitu Nama dan Rambut

1.2. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi

a. atribut variabel;

\*Umur

\*Nama

\*Jenis kelamin

\*Warna rambut

\*Tinggi badan

\*Berat badan

b. perilaku/ behavior

\*Tidur \*Berjalan \*Bekerja

\*Makan \*Berbicara

## [No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara public karena dapat diakses oleh kelas lain atau yang bukan anggota dari kelas tersebut.
- 2) Alasan solusi ini karena atribut digunakan berdasarkan variabel karakteristik yang sesuai dalam menggambarkan objek manusia.  
Perilaku digunakan berdasarkan aktivitas yang sering dilakukan oleh manusia
- 3) Perbaikan kode program dengan cara menggunakan dua parameter seperti string nama dan string rambut dan sesuaikan jumlah parameter yang diberikan serta menghapus manusia1 dengan hanya menggunakan manusia saja sehingga menampilkan output yang sesuai dengan kode program.

## [No.3 ] Penyusunan Algoritma dan Kode Program

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.  
Misalkan algoritma memasak mi instan:
  - (a) Mulai program
  - (b) Mendefinisikan kelas dengan nama yang sesuai
  - (c) Membuat constructor pada kelas
  - (d) Membuat method main dengan deklarasi objek dan eksekusi constructor
  - (e) Program selesai .

- 2) Kode program dan luaran

Main.java	Output
<pre>1- public class Manusia { // deklarasi kelas 2   //deklarasi atribut Manusia dalam variabel 3   String nama, rambut; 4 5   //deklarasi constructor 6- public Manusia (String nama, String rambut) { 7-     System.out.println(" Nama saya : " + nama + 8       "\n Warna Rambut : " + rambut); 9   } 10 11   //deklarasi method utama 12- public static void main( String[] args) { 13     Manusia satu = new Manusia("Putri", "hitam"); 14   } 15 } 16 17</pre>	<pre>java -cp /tmp/dMNjUk8iDs/Manusia Nama saya : Putri Warna Rambut : hitam  === Code Execution Successful ===</pre>

- a) Screenshot/ Capture potongan kode dan hasil luaran  
Beri komentar pada kode yang di Screenshot  
Komentar: Kelas manusia memiliki dua variabel yaitu nama dan rambut  
Constructor berfungsi mengatur objek manusia dengan parameter yang digunakan.
- b) Analisa luaran yang dihasilkan  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sehingga menampilkan output dari program yang di buat.

## [No.4] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

### 1) Evaluasi

- a) Apa konsekuensi/dampak dari kode program yang dibuat?
  - b) Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)
- jawaban Evaluasi: Pada program itu saya mengkonversi bentuk kelas public karena

class publik bersifat publik Karena dapat diakses oleh kelas lain yang bukan dari anggota kelas tersebut Setelah dikonversi, saya mengevaluasi bahwa tipe data string lebih baik digunakan untuk bentuk data seperti nama dan warna rambut.

Penjelasan yaitu: Tipe data numerik ternyata tidak dapat digunakan untuk kode program yang menggunakan teks karena tipe data numerik menggunakan angka yang tidak diperlukan dalam kode pemrograman diatas.Evaluasi materi yang diberikan sudah mencakup atribut,constructor,dan method utama . kekurangan eksprimennya yaitu seharusnya untuk memperluas fungsionalitas dan memastikan program lebih fleksibel ,penambahan input dan tipe data numerik yang lebih mendetail sangat dianjurkan, dimana berguna untuk meningkatkan kemampuan program dalam menangani data yang lebih kompleks .

Nama & NPM	Topik:	Tanggal:
IVANALY DEES TUMANGGER G1F024033	OBJEK	16 SEPTEMBER 2024

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

- 1) Uraikan permasalahan dan variabel

```
public class Ortu {  
    //deklarasi constructor (variabel constructor)  
    public ortu {  
        //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:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
The constructor Ortu(String, String) is undefined  
at Ortu.main(Ortu.java:9)
```

#### Latihan 2:

- 2.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!
- 2.2. Apabila Ortu memiliki data variabel umur = 25 dan jenis kelamin = P (untuk Perempuan), rekomendasikan constructor dengan parameter yang baru untuk ditambahkan dalam program!

- 2.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

Pada soal masih ada pesan kesalahan constructor tidak didefinisikan dalam kelas ortu dan kesalahan pada definisi constructor public ortu salah karena nama constructor harus sama dengan nama Ortu .

Atau

Diketahui dari soal : variabel yang digunakan Nama, warna rambut, umur dan jenis kelamin

2.2. Apabila Ortu memiliki data variabel umur = 25 dan jenis kelamin = P (untuk Perempuan), rekomendasikan constructor dengan parameter yang baru untuk ditambahkan dalam program!

Rekomendasi constructor dengan parameter yang baru adalah untuk menambahkan data variabel umur dan jenis kelamin maka perlu mendeklarasikan atribut baru, memperbarui constructor untuk menerima parameter tambahan dan memperbarui constructor untuk memasukkan atribut baru.

### [No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara public karena dapat diakses oleh kelas lain atau yang bukan anggota dari kelas tersebut.
- 2) Alasan solusi ini karena menambahkan atribut umur dan jenis kelamin memungkinkan untuk menyimpan informasi yang lebih lengkap tentang objek Ortu.
- 3) Perbaiki kode program dengan cara menggunakan constructor dengan parameter tambahan diperlukan untuk menambahkan semua variabel baru saat objek dibuat, dimana ini memastikan bahwa objek ortu memiliki semua informasi yang dibutuhkan dan siap digunakan setelah dibuat.

### [No.3 ] Penyusunan Algoritma dan Kode Program

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.  
Misalkan algoritma memasak mi instan:
  - (a) Mulai program
  - (b) Deklarasi kelas ortu
  - (c) Deklarasi atribut
  - (d) Deklarasikan constructor dan parameter tambahan
  - (e) Membuat method main seperti buat objek ortu
  - (f) Akhiri program

- 2) Kode program dan luaran

Main.java	Output
<pre>1- public class Ortu { 2-     String nama, rambut, umur, jeniskelamin; 3- 4-     public Ortu (String nama, String rambut, String umur, String         jeniskelamin) { 5-         System.out.println(" Nama saya : "+ nama + 6-         "\n Umur: " + umur + 7-         "\n Jenis Kelamin: " + jeniskelamin + 8-         "\n Warna Rambut : " + rambut); 9-     } 10-     public static void main (String[] args) { 11-         Ortu satu = new Ortu("Putri", "hitam", "25", "perempuan"); 12-     } 13- }</pre>	<pre>java -cp /tmp/2ozHByJy6Y/Ortu Nama saya : Putri Umur: 25 Jenis Kelamin: perempuan Warna Rambut : hitam  === Code Execution Successful ===</pre>

- a) Screenshot/ Capture potongan kode dan hasil luaran  
Beri komentar pada kode yang di Screenshot  
Komentar yaitu output dari kode ini adalah nama saya : putri dan warna rambut : hitam
- b) Analisa luaran yang dihasilkan  
Luaran sudah sesuai dengan program yang disusun.  
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sehingga menampilkan output dari program yang di buat.

#### [No.4] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

##### 1) Analisa

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

b)Apakah dasar dari pengambilan keputusan anda untuk kasus ini ?

jawaban Analisis: Pada program itu saya mengkonversi bentuk kelas publik Karena dapat diakses oleh kelas lain yang bukan dari anggota kelas tersebut. Perbaiki program dengan menambahkan nama variabel ortu harus sama-sama menggunakan huruf kapital agar tidak melanggar konvensi penamaan java dan constructor dengan parameter karena struktur java mengharuskan setiap kelas memiliki method main dan deklarasi variabel sebelum digunakan .

Nama & NPM	Topik:	Tanggal:
IVANALY DEES TUMANGGER G1F024033	UNIT TIGA METHOD	16 SEPTEMBER 2024

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

Uraikan permasalahan dan variabel

Contoh: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 {

System.out.println(" Hobi Menonton : " + film);

}

int sukaNonton {

episode\*durasi;

```

    }

    //deklarasi method utama
    public static void main( String[] args) {
        Manusia satu = new Manusia("Putri", "hitam");
        satu.sukaNonton("Drakor");
        int jumlahJam = satu.sukaNonton(2, 2);
        System.out.println("Jam nonton = " +jumlahJam + " jam");
    }
}

```

Luaran 3:

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

The method sukaNonton(String) is undefined for the type Manusia1

The method sukaNonton(int, int) is undefined for the type Manusia1

at Manusia1.main(Manusia1.java:23)

Latihan 3:

3.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

3.2. Ubahlah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior anda

3.3. Berdasarkan Contoh 3 dan Latihan 3.2. simpulkan perbedaan:

- a) constructor overloading dan overriding
- b) method overloading, dan method overriding
- c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

3.1 Kelas yang dideklarasikan adalah Manusia, tapi konstruktor yang ditulis adalah public Manusia1, yang seharusnya hanya menggunakan public Manusia. Metode sukaNonton tidak memiliki parameter yang dideklarasikan dengan benar. Metode sukaNonton kedua yang menghitung jumlah episode diketik sebagai variabel atau atribut, padahal harus berupa metode dan Nama variabel film tidak dideklarasikan dalam metode .

Perbaiki kode yaitu Nama kelas dan konstruktor harus disamakan menjadi Manusia. Metode sukaNonton didefinisikan dengan parameter yang sesuai. Atribut film dideklarasikan dalam metode sukaNonton. Metode sukaNonton kedua dideklarasikan dengan tipe pengembalian int dan menghitung durasi menonton.

3.2 Saya mengubah metode sukaNonton menjadi sukaBaca karena saya suka membaca , dan menambahkan metode waktuBaca untuk menghitung waktu yang dihabiskan baca berdasarkan jumlah halaman dan waktu membaca per bab.

3.3 Constructor Overloading: Adalah ketika ada beberapa constructor dalam satu kelas dengan jumlah parameter atau tipe parameter yang berbeda. Ini memungkinkan pembuatan objek dengan berbagai cara Constructor Overriding: Tidak ada konsep constructor overriding dalam Java karena constructor tidak diwarisi.

Method Overloading: yaitu ketika ada beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda (baik jumlah atau tipe).

Method Overriding: Adalah ketika sebuah metode pada kelas anak menimpa metode dengan nama dan parameter yang sama dari kelas induk.

c) Method yang Mengembalikan Nilai dan Method Tidak Mengembalikan Nilai

Method yang Mengembalikan Nilai: Adalah metode yang memiliki tipe pengembalian (selain void) dan menggunakan pernyataan return untuk mengembalikan nilai.

Method Tidak Mengembalikan Nilai: Adalah metode yang dideklarasikan dengan void dan tidak mengembalikan nilai apapun.

## **[No.2] Analisis dan Argumentasi**

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara public karena dapat diakses oleh kelas lain atau yang bukan anggota dari kelas tersebut. Kelas ini adalah contoh dari kelas sederhana dengan method overloading dimana atribut atau variabel nya menerima parameter String, dan yang lainnya menerima dua parameter int. Ini adalah contoh dari method overloading di mana beberapa method bisa memiliki nama yang sama tetapi dengan parameter yang berbeda.

2) Alasan solusi ini karena Nama konstruktor salah: Harus sama dengan nama kelas, yaitu Manusia, bukan Manusia1. Deklarasi metode sukaNonton salah: Metode harus memiliki tanda kurung parameter yang sesuai, dan variabel film dalam metode tidak dideklarasikan. Metode sukaNonton yang mengembalikan nilai harus berupa fungsi dengan parameter dan logika perhitungan, bukan deklarasi variabel.

3) Perbaiki kode program dengan cara Nama konstruktor diubah menjadi Manusia sesuai nama kelas. Metode sukaNonton(String film) diperbaiki dengan penambahan parameter film untuk menghindari error. Metode sukaNonton(int episode, int durasi) diperbaiki agar menghitung jumlah jam menonton berdasarkan episode dan durasi per episode.

## **[No.3 ] Penyusunan Algoritma dan Kode Program**

Algoritma

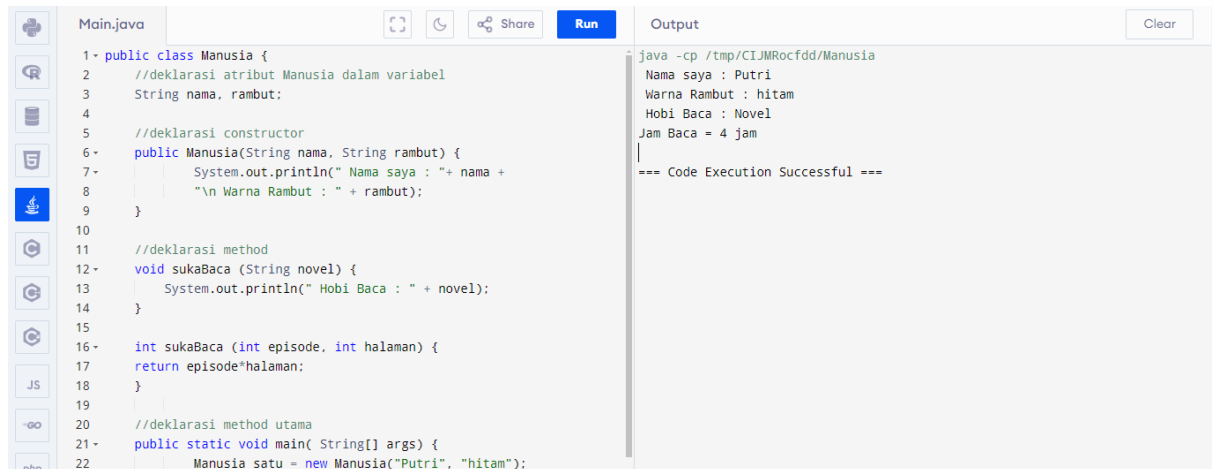
Algoritma adalah langkah-langkah penyelesaian masalah.

Misalkan algoritma memasak mi instan:

- (a). Mulai Program
- (b) Deklarasikan kelas Manusia dengan atribut nama dan rambut
- (c). Buat konstruktor untuk menginisialisasi atribut dan mencetak nama serta warna rambut
- (d) Ubah sukaNonton menjadi perilaku diri sendiri seperti suka baca.
- (e) Deklarasikan metode sukaBaca yang menerima parameter Buku dan mencetak hobinya.
- (f) Deklarasikan metode sukaBaca yang mengembalikan nilai dengan parameter episode dan durasi
- (g) Dalam main, buat objek Manusia dengan nama "Putri" dan warna rambut "Hitam".
- (h) . Panggil metode sukaBonton dengan string Novel.
- (i) . Hitung total jam membaca berdasarkan jumlah episode dan durasi, lalu cetak hasilnya.
- (j) . Akhiri Program



## 2) Kode program dan luaran



```
1- public class Manusia {
2-     //deklarasi atribut Manusia dalam variabel
3-     String nama, rambut;
4-
5-     //deklarasi constructor
6-     public Manusia(String nama, String rambut) {
7-         System.out.println(" Nama saya : " + nama +
8-             "\n Warna Rambut : " + rambut);
9-     }
10-
11-     //deklarasi method
12-     void sukaBaca (String novel) {
13-         System.out.println(" Hobi Baca : " + novel);
14-     }
15-
16-     int sukaBaca (int episode, int halaman) {
17-         return episode*halaman;
18-     }
19-
20-     //deklarasi method utama
21-     public static void main( String[] args) {
22-         Manusia satu = new Manusia("Putri", "hitam");
```

```
java -cp /tmp/CIJMRocfdd/Manusia
Nama saya : Putri
Warna Rambut : hitam
Hobi Baca : Novel
Jam Baca = 4 jam
=== Code Execution Successful ===
```



```
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 sukaBaca (String novel) {
13-         System.out.println(" Hobi Baca : " + novel);
14-     }
15-
16-     int sukaBaca (int episode, int halaman) {
17-         return episode*halaman;
18-     }
19-
20-     //deklarasi method utama
21-     public static void main( String[] args) {
22-         Manusia satu = new Manusia("Putri", "hitam");
23-         satu.sukaBaca("Novel");
24-         int jumlahJam = satu.sukaBaca(2, 2);
25-         System.out.println("Jam Baca = " + jumlahJam + " jam");
```

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

Komentar yaitu output dari kode ini adalah nama saya : putri dan warna rambut : hitam

b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sehingga menampilkan output dari program yang di buat.

## [No.4] Kesimpulan

### 1) Analisa

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

b) Apakah dasar dari pengambilan keputusan anda untuk kasus ini ?

jawaban Analisis: Pada program itu saya mengkonversi bentuk kelas publik Karena dapat diakses oleh kelas lain yang bukan dari anggota kelas tersebut. Constructor Overloading: Terjadi ketika ada

beberapa constructor dengan nama yang sama namun memiliki parameter yang berbeda. Constructor Overriding: Tidak terdapat konsep overriding untuk constructor karena constructor tidak diturunkan. Method Overloading: Terjadi ketika terdapat beberapa method dengan nama yang sama tetapi menggunakan parameter yang berbeda. Method Overriding: Terjadi ketika class mendefinisikan ulang method yang sudah ada di dalam class. Dasar pengambilan keputusan yaitu Berdasarkan analisis kesalahan pada constructor dan method, masalah ini disebabkan oleh ketidaksesuaian parameter dengan deklarasi. sehingga dengan memperbaiki deklarasi constructor dan method agar sesuai dengan parameter yang diperlukan, sehingga kesalahan dapat diatasi dan menampilkan hasil output.

Nama & NPM	Topik:	Tanggal:
IVANALY DEES TUMANGGER G1F024033	EXTENDS	16 SEPTEMBER 2024

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

Uraikan permasalahan dan variabel

```

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
    } }

```

#### 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. Evaluasi method yang dimiliki class Anak extends Ortu dengan method di class Ortu!  
Apakah penulisan method ini sudah efisien?
- 4.2. Setelah dirunning di JDoodle, catat waktu eksekusinya.  
Rekomendasikan perbaikan penulisan kode method untuk dapat mengefisienkan waktu eksekusi!

Jawaban:

1.Evaluasi Method di kelas Anak dan Ortu. Kelas Anak mewarisi dari kelas Ortu. Ada 2method utama yang dievaluasi: 1. Method sukaMenonton, di kelas Ortu, method ini menerima satu parameter dengan tipe String dan di kelas Anak, method ini di-overload, yaitu ada dua versi: Versi pertama menerima satu parameter String, yang sebenarnya sama dengan versi di kelas Ortu. Versi kedua menerima dua parameter (int dan String), yang merupakan penambahan fungsi di kelas Anak.

2. Method sukaMembaca, di kelas Ortu, method ini menerima satu parameter tipe data String.

4.2. Peningkatan Efisiensi Kode dan Waktu Eksekusi. Beberapa rekomendasi untuk meningkatkan efisiensi kode dan waktu eksekusi: Hapus Method sukaMenonton(String a) di Kelas Anak: Karena method sukaMenonton(String a) di kelas Anak sama dengan yang ada di kelas Ortu, tidak perlu menyimpannya. Hal ini akan mengurangi duplikasi kode dan meningkatkan efisiensi dalam hal maintainability.Pertahankan Method sukaMenonton(int a, String b): Penambahan method ini melalui overloading adalah valid dan bermanfaat karena memperluas fungsi di kelas Anak. Tidak perlu perubahan pada bagian ini.

#### [No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara tipe kelas yang turunan atau kelas anak yang menggunakan konsep pewarisan yang diturunkan dari ortu ke anak.
- 2) Alasan solusi ini karena untuk mengatasi permasalahan ini adalah dengan menghilangkan duplikat method yang tidak perlu di kelas Anak, karena method tersebut sudah tersedia di kelas Ortu dan diwariskan secara otomatis.sehingga dengan cara ini lebih efisien dan mudah dipahami
- 3)Perbaikan kode program dengan cara menghilangkan duplikasi method sukaMenonton(String a) di kelas Anak dan mempertahankan penimpanan method yang diperlukan saja.

#### [No.3 ] Penyusunan Algoritma dan Kode Program

## Algoritma

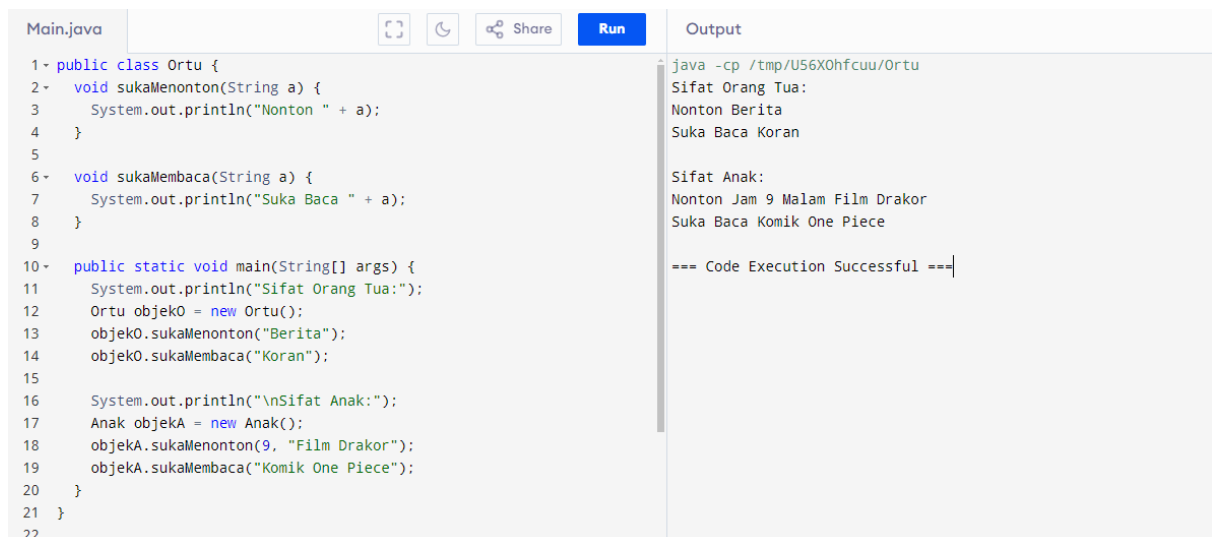
Algoritma adalah langkah-langkah penyelesaian masalah.

Misalkan algoritma memasak mi instan:

Algoritma Program:

1. Membuat kelas Ortu yang memiliki dua method sukaMenonton(String a) dan sukaMembaca(String a).
2. Membuat kelas Anak yang mewarisi kelas Ortu.
3. Mengeksekusi program agar program lebih singkat.
4. Jalankan program
5. Akhiri program

## 2) Kode program dan luaran



The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'Ortu' with two methods: 'sukaMenonton(String a)' and 'sukaMembaca(String a)'. The 'main' method creates an 'Ortu' object and calls these methods. The 'Output' pane shows the results of the execution.

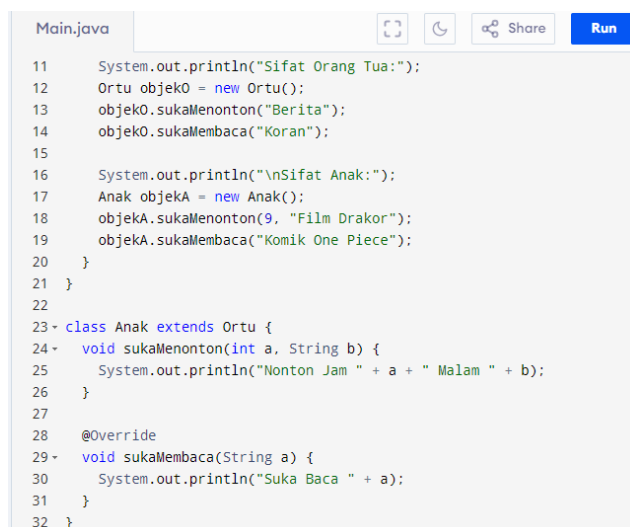
```
1 public class Ortu {
2     void sukaMenonton(String a) {
3         System.out.println("Nonton " + a);
4     }
5
6     void sukaMembaca(String a) {
7         System.out.println("Suka Baca " + a);
8     }
9
10    public static void main(String[] args) {
11        System.out.println("Sifat Orang Tua:");
12        Ortu objek0 = new Ortu();
13        objek0.sukaMenonton("Berita");
14        objek0.sukaMembaca("Koran");
15
16        System.out.println("\nSifat Anak:");
17        Anak objekA = new Anak();
18        objekA.sukaMenonton(9, "Film Drakor");
19        objekA.sukaMembaca("Komik One Piece");
20    }
21 }
22
```

Output

```
java -cp /tmp/U56X0hfcuu/Ortu
Sifat Orang Tua:
Nonton Berita
Suka Baca Koran

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

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



The screenshot shows the same Java IDE with 'Main.java'. The code now includes a second class, 'Anak', which extends 'Ortu'. The 'Anak' class overrides the 'sukaMenonton' and 'sukaMembaca' methods. The 'main' method remains the same, but it now creates an 'Anak' object and calls its methods.

```
11    System.out.println("Sifat Orang Tua:");
12    Ortu objek0 = new Ortu();
13    objek0.sukaMenonton("Berita");
14    objek0.sukaMembaca("Koran");
15
16    System.out.println("\nSifat Anak:");
17    Anak objekA = new Anak();
18    objekA.sukaMenonton(9, "Film Drakor");
19    objekA.sukaMembaca("Komik One Piece");
20 }
21 }
22
23 class Anak extends Ortu {
24     void sukaMenonton(int a, String b) {
25         System.out.println("Nonton Jam " + a + " Malam " + b);
26     }
27
28     @Override
29     void sukaMembaca(String a) {
30         System.out.println("Suka Baca " + a);
31     }
32 }
```

a) Screenshot/ Capture potongan kode dan hasil luaran

b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data sehingga menampilkan output dari program yang di buat.

#### **[No.4] Kesimpulan**

##### **1) Analisa**

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

b)Apakah dasar dari pengambilan keputusan anda untuk kasus ini ?

jawab:

- a) Dapat disimpulkan bahwa menghindari penulisan method yang tidak perlu di kelas anak akan membuat kode lebih efisien dan mudah dipelihara. Overriding atau overloading method hanya dilakukan jika memang ada perubahan logika atau penambahan fungsi yang diperlukan pada kelas anak. Dengan mengurangi duplikasi kode, tidak hanya efisiensi waktu eksekusi yang sedikit meningkat, tetapi juga readability dan maintainability kode menjadi lebih baik.
- b) Untuk mengatasi permasalahan ini adalah prinsip pewarisan pada OOP, di mana kelas anak secara otomatis mewarisi method dan atribut dari kelas induk. Jika tidak ada perubahan pada method yang diwariskan, maka tidak perlu menulis ulang method tersebut di kelas anak.

#### **Refleksi**

Pada minggu ini saya mempelajari materi tentang konsep class, object, serta extends dalam pemrograman berorientasi objek (OOP). Secara umum, materi ini memperdalam pemahaman saya mengenai bagaimana class dan object bekerja dalam sebuah program yang dijalankan serta bagaimana kita bisa memanfaatkan extends untuk memperluas fungsionalitas dari sebuah class. Tantangan yang saya hadapi adalah saya kurang paham bagaimana extends bekerja ketika ada metode yang sama di superclass dan subclass, dan cara mengatasi kondisi kesalahan yang mungkin muncul pada saat program sedang dijalankan. Dimana materi dengan materi yang ada pada minggu ini saya sedikit lebih paham i dasar-dasarr materi yang diberikan.