

**Template Lembar Kerja Individu dan Kelompok**

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>IQBAL FERDINAND PUTRA G1F024039</b>	<b>Kelas, Objek, Method</b>	<b>17 September 2024</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>		

### [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

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:
  - a. Pada kode ini nama dan rambut adalah atribut dari kelas Manusia. Pada objek Manusia memiliki nilai nama dan rambut yang berbeda beda.
  - b. Perilaku menunjukkan apa yang dapat dilakukan oleh suatu objek berdasarkan atributnya. Pada kelas Manusia ini mencerminkan kehidupan kita di dunia nyata ini.
- 2) Alasan solusi ini karena: Solusi ini paling relevan karena secara langsung mengatasi akar permasalahannya.
- 3) Perbaiki kode program dengan cara:

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

- 1) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - (a) Mulai
  - (b) Deklarasi kelas Manusia
  - (c) Deklarasi variabel
  - (d) Deklarasi constructor
  - (e) Metode main
  - (f) Run
  - (g) Selesai
- 2) Kode program

```

1 public class Manusia { // deklarasi kelas
2     // deklarasi variabel
3     String nama;
4     String rambut;
5
6     // deklarasi constructor tanpa parameter
7     public Manusia() {
8         System.out.println("Kelas Manusia tanpa nama");
9     }
10    public static void main(String[] args) {
11        Manusia aksesclass = new Manusia();
12        aksesclass.nama = "Iqbal Ferdinand Putra";
13        aksesclass.ambut = "Hitam";
14        System.out.println(aksesclass.nama);
15        System.out.println(aksesclass.ambut);
16    }
17 }

```

a) Luaran

Output      Generated Files

```

Kelas Manusia tanpa nama
Iqbal Ferdinand Putra
Hitam
|

```

Compiled and executed in 1.3 sec(s)

b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Ketika kode program dijalankan luaran sudah sesuai dengan permintaan dari kode program tersebut.

### [No.1] Kesimpulan

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

#### 1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Membuat kode program sederhana dari seorang manusia dalam bentuk objek. Adapun algoritma yang dilakukan mendeklarasikan kelas Manusia dengan atribut nama dan rambut lalu memberi nilai pada atribut nama dan rambut dan diakhiri mencetak nilai dari atribut.

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

Contoh jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena ingin membuat kelas tersebut dapat diakses dari kelas lain dalam program ataupun berbeda. Perbaiki program dengan menambahkan metode main dan aksesclass objek dari kelas Manusia. Karena struktur java mengharuskan pengaturan Tingkat akses yang jelas untuk tiap anggota agar kode program menjadi lebih terstruktur dan aman.

### [No. 2] Identifikasi Masalah:

- 2) 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

- 4) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:
- 2.1 Menambahkan variabel constructor serta data pribadi seperti nama , warna rambut, warna kulit, dan jenis kelamin
  - 2.2 Sifat (atribot): saya akan menurunkan seperti fisik dari warna mata, tinggi badan, warna rambut, golongan darah kalo dari kepribadian seperti kecerdasan, kreativitas. Dalam contoh ini hanya atribut nama dan rambut digunakan.  
Constructor: Kombinasi gen dari orang tua yang menentukan sifat fisik dan potensi keturunan. Kelas keturunan dapat memanggil constructor dari kelas ortu dan juga dapat menambahkan parameter tambahan.  
Perilaku positif (behavior): Dari perilaku positif bisa menurunkan nilai jujur, disiplin, tanggung jawab. Kelas keturunan mewarisi method dari kelas ortu.
- 5) Alasan solusi ini karena:  
Memastikan bahwa kelas dan keturunannya nanti dirancang dengan baik, mendukung pemeliharaan kode dan menciptakan struktur program yang fleksibel dan efisien.
- 6) Perbaikan kode program dengan cara:  
Penambahan variabel constructor dan melengkapi data pribadi

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

- 3) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
- a) Mulai
  - b) Buka aplikasi

- c) Buat project
- d) Buat kelas Ortu
- e) Deklarasi kelas Ortu
- f) Deklarasi Constructor
- g) Method main
- h) Run
- i) Selesai

4) Kode program

```
Main.java ⌵ ☀ 🔗 Share Run  
1- public class Ortu {  
2    //deklarasi constructor  
3- public Ortu(String nama, String rambut, String kulit, String  
    jeniskel) {  
4    //nama dan rambut adalah variabel constructor  
5-    System.out.println(" Nama saya : " + nama +  
6    "\n Warna Rambut : " + rambut + "\n Warna Kulit = " + kulit +  
        "\n Jenis Kelamin = " + jeniskel);  
7 }  
8- public static void main (String[] args) {  
9    Ortu satu = new Ortu("Iqbal", "Hitam", "Kuning", "L");  
10 }  
11 }
```

c) Luaran

```
Output  
java -cp /tmp/5dQiCZIomq/Ortu  
Nama saya : Iqbal  
Warna Rambut : Hitam  
Warna Kulit = Kuning  
Jenis Kelamin = L  
  
=== Code Execution Successful ===
```

- d) Analisa luaran yang dihasilkan
- Konstruktor berfungsi dengan benar untuk mencetak informasi yang diberikan saat objek dibuat dan output memberikan konfirmasi bahwa constructor berfungsi dengan baik serta tampilan yang benar.

(Tuliskan penjelasan dari program yang dibuat, apakah kode dan luaran sudah benar?)

## [No.2] Kesimpulan

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

### 2) Analisa

- c) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
1. Permasalahan  
Program ini bertujuan untuk membuat objek ortu dengan metode main sebuah objek ortu dengan atribut nama, warna rambut, warna kulit, dan jenis kelamin.
  2. Algoritma
    - Deklarasi kelas
    - Konstruktor
    - Metode utama
  3. Kode program  
Kode program telah diberi dan sesuai dengan algoritma yang jelaskan.

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

Pada program itu saya menggunakan bentuk kelas public karena member dapat diakses dari mana saja, terbuka public. Perbaiki program dengan menambahkan variabel constructor serta data pribadi.

## [No. 3] Identifikasi Masalah:

- 3) Uraikan permasalahan dan variabel

```
public class Manusia {  
    //deklarasi atribut Manusia dalam variabel  
    String nama, rambut;  
  
    //deklarasi constructor  
    public Manusia1(String nama, String rambut) {  
        System.out.println(" Nama saya : " + nama +  
            "\n Warna Rambut : " + rambut);  
    }  
  
    //deklarasi method  
    void sukaNonton(String film) {  
        System.out.println(" Hobi Menonton : " + film);  
    }  
  
    //deklarasi method utama  
    public static void main( String[] args) {  
        Manusia satu = new Manusia("Putri", "hitam");  
        satu.sukaNonton("Drakor");  
    }  
}
```

### Luaran 3:

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

### Latihan 3:

- 3.1. Analisa perbedaan deklarasi constructor, method, dan method utama!
- 3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?
- 3.3. Uraikan perbedaan berikut:
  - a) constructor overloading dan overriding

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

### **[No.3] Analisis dan Argumentasi**

- 7) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:
  - 3.1 konstruktor adalah dipakai untuk menganalisis objek saat objek dibuat contoh public Manusia(String nama, String rambut), method merupakan fungsi yang didefinisi dalam suatu kelas contoh void sukaNonton(String film), method utama merupakan titik awal eksekusi program java.
  - 3.2 Konstruktor saat membuat objek baru, perlu menganalisis nilai atribut objek tersebut dan method saat ingin melakukan perhitungan atau manipulasi data
  - 3.3 overloading membuat beberapa constructor dengan nama yang sama tetapi parameter yang berbeda. overloading tidak berlaku untuk constructor karena constructor karena constructor tidak diwariskan. (Constructor)  
overloading memuat beberapa method dengan nama yang sama tetapi parameter yang berbeda dalam satu kelas. Overriding memuat method dengan nama dan parameter yang sama pada kelas turunan untuk mengganti implementasi method pada kelas induk. (Method)  
method yang mengembalikan nilai memiliki tipe kembalian tertentu sedangkan method yang tidak mengembalikan nilai tidak memiliki tipe kembalian.
- 8) Alasan solusi ini karena:  
Menyesuaikan prinsip prinsip yang ada
- 9) Perbaikan kode program dengan cara:  
Pergantian data diri seperti nama, warna rambut, hobi dan juga menambahkan waktu menonton

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

- 5) Algoritma  
Algoritma adalah langkah-langkah penyelesaian masalah.
  - a. Mulai
  - b. Pembuatan objek
  - c. Panggilan metode
  - d. Perhitungan
  - e. Run
  - f. Selesai
- 6) Kode program

```
Main.java [Full Screen] [Theme] [Share] [Run]
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     int sukaNonton (int episode , int durasi ) {
17         return episode*durasi;
18     }
19
20     //deklarasi method utama
21     public static void main( String[] args) {
22         Manusia satu = new Manusia("Iqbal", "hitam");
23         satu.sukaNonton("Ultramen");
24         int jumlahjam = satu.sukaNonton(2, 2);
25         System.out.println("jam nonton = " +jumlahjam + "jam");
26     }
```

e) Luaran

```
Output
java -cp /tmp/2zA24w18Xf/Manusia
Nama saya : Iqbal
Warna Rambut : hitam
Hobi Menonton : Ultramen
jam nonton = 4jam

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

f) Analisa luaran yang dihasilkan program ini membuat sebuah objek manusia dengan nama Iqbal dan warna rambut hitam dan juga memberi informasi hobi dan menghitung total durasi menonton. Luaran sudah sesuai dengan program yang disusun.

### [No.3] Kesimpulan

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



### 3) Analisa

- e) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program:  
Program ini dirancang untuk presentasi seorang manusia dengan atribut nama dan warna rambut serta aktivitas menonton dan juga waktu menonton. Dalam algoritma yaitu pembuatan objek, panggilan metode, perhitungan, dan output.
- f) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Pada program itu saya menggunakan bentuk kelas public karena agar bisa diakses oleh umum. Perbaiki program dengan menambahkan atribut data diri seperti nama, warna rambut serta aktivitas hobi dengan durasinya juga.

### [No. 4] Identifikasi Masalah:

- 4) 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 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

10) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:

4.1 Overriding: Kelas Anak melakukan overriding terhadap metode `sukaMenonton` dan `sukaMembaca` dari kelas `Ortu`. Ini mungkin metode tersebut diubah di kelas anak meskipun tetap memiliki nama dan parameter sama.

Overloading: Kelas Anak juga melakukan overloading pada metode `sukaMenonton`, menambahkan versi yang menerima dua parameter (int dan String).

Metode baru: Kelas Anak menambahkan metode baru `sukaOlahraga`, yang tidak ada di kelas `Ortu`

4.2 Disini kelas Anak mewarisi semua method dari kelas induk (`Ortu`), tetapi beberapa method diubah atau ditambah (override dan overload) seperti menambah pada objek anak dengan method `sukaOlahraga` yaitu basket (method baru yang tidak ada di kelas induk, dan hanya di definisi di kelas anak)

11) Alasan solusi ini karena:

Alasan Solusi ini didasari pada prinsip dasar pewarisan dalam pemrograman berorientasi objek, terutama konsep overriding dan overloading.

12) Perbaiki kode program dengan cara:

Menambahkan objek pada Anak dengan method `sukaOlahraga` yaitu basket

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

7) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- a. Mulai
- b. Buat kelas `Ortu`
- c. Deklarasi kelas
- d. Deklarasi metode
- e. Main method
- f. Run

g. Selesai

## 8) Kode program

```
KelasKujava  module-info.java  module-info.java  Ortu.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 }
9 public static void main(String [] args) {
10     System.out.println("Sifat Orang Tua :");
11     Ortu objekO = new Ortu(); // memanggil objek induk
12     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
13     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
14 }
15 System.out.println("\n Sifat Anak :");
16 Anak objekA = new Anak(); //memanggil objek anak
17 objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
18 objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
19 objekA.sukaOlahraga("Basket"); // memanggil method baru di anak
20 }
21 }
22 class Anak extends Ortu {
23     void sukaMenonton(int a, String b) {
24         System.out.println("Nonton Jam " + a + " Malam " + b);
25     }
26     void sukaMenonton(String a) { // method induk spesifik
27         System.out.println("Nonton " + a);
28     }
29     void sukaMembaca(String a) { // method induk umum bisa diubah anak
30         System.out.println("Suka Baca " + a);
31     }
32     void sukaOlahraga(String c) { //method baru di class Anak
33         System.out.println("Suka Olahraga " + c);
34     }
35 }
36 public static void main(String [] args) {
37     System.out.println("Sifat Orang Tua :");
38     Ortu objekO = new Ortu(); // memanggil objek induk
39     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
40     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
41 }
42 System.out.println("\n Sifat Anak :");
43 Anak objekA = new Anak(); //memanggil objek anak
44 objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
45 objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
46 objekA.sukaOlahraga("Basket"); // memanggil method baru di anak
47 }
48 }
```

## g) Luaran

```
@ Javadoc Declaration Console ×
<terminated> Anak [Java Application] C:\Users\Acer\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

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

## h) Analisa luaran yang dihasilkan

Contoh:

Tipe data yang tampil adalah string dan integer, dan metode yang di overriding maupun method baru dipanggil sesuai dengan napa yang diharapkan. Luaran sudah sesuai dengan program yang disusun.

## [No.4] Kesimpulan

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

### 4) Analisa

- g) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  
Pada program ini terdapat dua kelas yaitu Ortu (kelas induk) dan Anak (kelas anak), metode yang dari kelas Ortu di wariskan ke Anak melalui pewarisan. Kelas anak dapat mewarisi override atau overload perilaku dari kelas induk. Dan juga kelas anak dapat menambahkan metode baru yang tidak ada dikelas Ortu.

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

Contoh jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena agar dapat diakses oleh kelas lain. Perbaiki program dengan menambahkan metode baru sukaOlahraga hal ini menunjukkan anak memiliki kemampuan yang tidak dimiliki orang tua seperti suka olahraga basket.

### **Refleksi**

Selama seminggu ini saya mempelajari konsep penting dalam pemrograman seperti Kelas, Objek, Method, dan Extends.