

Nama & NPM	Topik:	Tanggal:
Hedy Rafian Firdaus G1F024027	Class (Kelas)	17 September 2024

[No. 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

Contoh 1:

```
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
- atribut variabel, dan
 - perilaku/ behavior untuk method!

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

Kelas Java Bagian 1 - Kelas, Objek, New, Constructor

Source : <https://youtu.be/60ldOc8m8Es>

[No. 1] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Untuk mengetahui ciri – ciri umum dari kelas Manusia yang ada pada contoh 1, kita bisa menambahkan 2 kategori utama, yaitu atribut variabel dan perilaku/behavior (method). Untuk penjelasan lebih lanjutnya, akan saya jabarkan sebagai berikut :

a) Atribut Variabel

Atribut variabel adalah data atau state yang disimpan dalam objek dari kelas tersebut. Dalam kode yang diberikan, Manusia memiliki dua atribut variabel:

- String nama: Variabel ini menyimpan nama dari objek Manusia.
- String rambut: Variabel ini menyimpan informasi tentang warna rambut dari objek Manusia.

b) Perilaku/Behavior (Method)

Perilaku atau method dalam kelas adalah fungsi atau tindakan yang dapat dilakukan oleh objek dari kelas tersebut. Dalam kode yang diberikan:

- Constructor Manusia(): Constructor adalah metode khusus yang dipanggil ketika sebuah objek baru dari kelas Manusia dibuat. Constructor ini tanpa parameter dan hanya mencetak pesan "Kelas Manusia tanpa nama" ke konsol.

[No. 1] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

- Mulai
- Membuat nama kelas dan nama file Manusia
- Melakukan deklarasi variabel
- Melakukan deklarasi constructor tanpa parameter
- Melakukan deklarasi method utama
- Menghasilkan keluaran
- Selesai

2) Tuliskan kode program dan keluaran

```

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 bowo = new Manusia ();
12    }
13 }

```

- Beri komentar pada kode
Untuk kode yang saya berikan telah sesuai dengan syarat yang berlaku untuk kode tersebut. Saya menambahkan kode deklarasi method utama supaya keluaran kode dapat dihasilkan.
- Uraikan keluaran yang dihasilkan

```

java -cp /tmp/mtqcesYQ71/Manusia
Kelas Manusia tanpa nama

=== Code Execution Successful ===

```

Keluaran yang dihasilkan telah sesuai dengan kode program yang dijalankan, Dimana hasil yang muncul yaitu :

Kelas Manusia tanpa nama

[No. 1] Kesimpulan

Analisa

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

Untuk soal ini, saya memakai objek nama dan rambut dengan menggunakan tipe data String untuk menyatakan kalimat. Lalu, untuk method yang dipakai yaitu pada kode Manusia() yang tidak diisi parameter.

Untuk Atribut variabel adalah data atau state yang disimpan dalam objek dari kelas tersebut, sedangkan perilaku atau method dalam kelas adalah fungsi atau tindakan yang dapat dilakukan oleh objek dari kelas tersebut

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

Untuk Latihan 1 ini, saya telah menganalisa mengenai atribut variabel dan method yang dimana saya bisa menegrti sistem kerjanya. Dan untuk kode program yang diberikan, saya hanya menambahkan deklarasi method utama sehingga kode program dapat berjalan.

Nama & NPM	Topik:	Tanggal:
Hedy Rafian Firdaus G1F024027	Objek	17 September 2024

[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?

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

Kelas Java Bagian 1 - Kelas, Objek, New, Constructor

Source : <https://youtu.be/60ldOc8m8Es>

[No. 2] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Untuk soal Latihan 2 ini, saya akan menambahkan beberapa ciri - ciri saya di dalam kode yang telah disediakan, yaitu warna kulit dan tinggi badan.

Ketika berbicara tentang "keturunan" dalam konteks pemrograman berorientasi objek, biasanya kita merujuk pada konsep pewarisan (inheritance). Berikut adalah analisis tentang sifat, constructor, dan perilaku positif yang akan diturunkan jika Anda membuat subclass dari Ortu:

- a) Sifat(atribut)

Untuk sifat sendiri, yang diturunkan kepada kelas Anak yang berasal dari kelas induknya adalah ciri – ciri kelas Ortu. Misalkan kelas Ortu memiliki rambut berwarna hitam, maka kelas Anak akan memiliki rambut berwarna hitam juga.
- b) Constructor

Constructor akan digunakan apabila ingin membuat objek dari kelas anak untuk menginisialisasi atribut yang akan diwariskan.
- c) Perilaku (behavior)

Untuk perilaku, kelas Anak akan mewarisi perilaku positif yang sama dengan kelas Ortu. Misalnya kelas Ortu memiliki hobi memancing, maka kelas Anak akan mewarisi kegiatan memancing dari kelas Ortu yang bersangkutan.

[No. 2] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

- a) Mulai
- b) Membuat nama kelas dan nama file Ortu
- c) Deklarasi constructor

- d) Deklarasi variabel
- e) Deklarasi method utama
- f) Menghasilkan luaran
- g) Selesai

2) **Tuliskan kode program dan luaran**

```

1 public class Ortu {
2     //deklarasi constructor
3     public Ortu(String nama, String rambut, String kulit, int tinggiBadan) {
4
5         //nama dan rambut adalah variabel constructor
6         System.out.println(" Nama saya : " + nama +
7             "\n Warna Rambut : " + rambut + "\n Warna Kulit : " + kulit + "\n
8             Tinggi Badan : " + tinggiBadan);
9     }
10    public static void main (String[] args) {
11        Ortu satu = new Ortu("Hedy Rafian Firdaus", "Hitam", "Putih", 177);
12    }
13 }

```

- c) Beri komentar pada kode
Kode yang diberikan telah sesuai dengan syarat kode yang berlaku, sehingga bisa menghasilkan luaran yang diperintahkan.
- d) Uraikan luaran yang dihasilkan

```

java -cp /tmp/tjjbq68rZ0/Ortu
Nama saya : Hedy Rafian Firdaus
Warna Rambut : Hitam
Warna Kulit : Putih
Tinggi Badan : 177

=== Code Execution Successful ===

```

Luaran yang dihasilkan telah sesuai dengan kode yang diprogram. Yang dimana luaran yang dihasilkan telah didapat, yaitu :

Nama saya: Hedy Rafian Firdaus
Warna Rambut : Hitam
Warna Kulit : Putih
Tinggi Badan : 177

[No. 2] Kesimpulan

Kreasi

- a) **Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?**
Untuk Latihan ini, saya dapat mengetahui mengenai metode pewarisan objek variabel dari kelas Ortu yang akan diwariskan ke anaknya. Saya bisa mewariskan berbagai macam ciri – ciri, seperti warna rambut, warna kulit, hobi, dan lain sebagainya.

Nama & NPM	Topik:	Tanggal:
Hedy Rafian Firdaus G1F024027	Method	17 September 2024

[No. 3] Identifikasi Masalah:

1) 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

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

Kelas Java Bagian 2 - Method dan Extends

Source : <https://youtu.be/6qULMlcv-eg>

[No. 3] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

1. Berikut adalah analisis singkat tentang perbedaan antara deklarasi constructor, method, dan method utama (main method) dalam Java:
 - a) Constructor

Constructor adalah metode khusus yang dipanggil saat sebuah objek dari kelas dibuat.

 - Nama: Harus sama dengan nama kelas.
 - Tipe Kembalikan: Tidak memiliki tipe kembalikan (tidak ada void atau tipe data lainnya).
 - Fungsi: Digunakan untuk menginisialisasi objek dengan nilai awal atau setup awal.
 - Pemanggilan: Dipanggil secara otomatis saat objek dibuat dengan new.

b) Method

Method adalah blok kode yang melakukan tugas tertentu dan dapat dipanggil untuk menjalankan fungsionalitas.

- Nama: Nama bebas, tetapi tidak boleh sama dengan nama constructor.
- Tipe Kembalian: Dapat memiliki tipe kembalian seperti int, String, void, dll.
- Fungsi: Digunakan untuk melakukan operasi atau manipulasi data dalam objek.
- Pemanggilan: Dipanggil secara eksplisit melalui objek atau kelas (untuk metode statis).

c) Method Utama (main method)

Method utama adalah titik masuk utama untuk program Java.

- Nama: Harus main.
- Tipe Kembalian: void.
- Parameter: Harus memiliki parameter String[] args.
- Fungsi: Digunakan oleh JVM untuk memulai eksekusi program Java.
- Pemanggilan: Dipanggil oleh JVM ketika program dijalankan; tidak dipanggil secara langsung oleh objek.

2. Kapan Menggunakan Constructor dan Method?

- Constructor : Digunakan pada saat menginisialisasi objek pada saat pertama kali dibuat. Ini adalah tempat di mana Anda menetapkan nilai awal atribut objek atau melakukan setup yang diperlukan.
- Method : Digunakan pada saat objek ingin diberikan suatu tindakan atau perilaku.

3. Uraikan perbedaan :

a) constructor overloading dan overriding

- Constructor Overloading : Constructor overloading terjadi ketika ada lebih dari satu constructor dalam sebuah kelas dengan nama yang sama tetapi parameter yang berbeda. Ini memungkinkan Anda untuk membuat objek dengan cara yang berbeda tergantung pada parameter yang disediakan.
- Constructor Overriding : Constructor overriding tidak ada dalam Java. Konsep overriding hanya berlaku untuk metode, bukan untuk constructor. Dengan kata lain, Anda tidak dapat mengoverride constructor di subclass. Constructor dari superclass tidak bisa diubah di subclass.

b) method overloading dan method overriding

- Method Overloading : kemampuan untuk mendefinisikan lebih dari satu metode dengan nama yang sama di dalam sebuah kelas, tetapi dengan parameter yang berbeda (jumlah, tipe, atau urutan parameter).
- Method Overriding : kemampuan ketika subclass mendefinisikan ulang metode yang sudah ada di superclass dengan nama, tipe kembalian, dan parameter yang sama. Ini memungkinkan subclass untuk memberikan implementasi khusus dari metode yang diwarisi.

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

- Method yang Mengembalikan Nilai, metode ini mengembalikan sebuah nilai dari tipe data tertentu setelah eksekusi. Tipe data yang dikembalikan ditentukan dalam deklarasi metode.
- Method yang tidak Mengembalikan Nilai, Metode ini tidak mengembalikan nilai apapun. Sebaliknya, metode ini dideklarasikan dengan tipe kembalian void.

- d) Untuk Latihan 3 ini, saya akan mengubah variabel nama, warna rambut, dan hobi menonton sesuai dengan yang saya inginkan di dalam kode yang telah disediakan.

[No. 3] Menyusunan Algoritma dan Kode Program

- 3) Rancang desain solusi atau algoritma
- Mulai
 - Membuat nama kelas dan nama file Manusia
 - Deklarasi atribut Manusia
 - Deklarasi constructor
 - Deklarasi method
 - Deklarasi method utama
 - Menghasilkan keluaran
 - Selesai
- 4) Tuliskan kode program dan keluaran

```
public class Manusia {  
    //deklarasi atribut Manusia dalam variabel  
    String nama, rambut;  
  
    //deklarasi constructor  
    public Manusia(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("Hedy", "Hitam");  
        satu.sukaNonton("Anime");  
    }  
}
```

- e) Beri komentar pada kode
Kode yang dibuat telah sesuai dengan syarat kode program yang dilakukan, sehingga dapat memunculkan hasil keluarannya.
- f) Uraikan keluaran yang dihasilkan

```
java -cp /tmp/rwLHQGkNrI/Manusia  
Nama saya : Hedy  
Warna Rambut : Hitam  
Hobi Menonton : Anime  
  
=== Code Execution Successful ===
```


Luaran yang dihasilkan telah sesuai dengan kode yang telah deprogram. Hasil yang muncul adalah :

Nama saya : Hedy

Warna Rambut : Hitam

Hobi Menonton : Anime

[No. 3] Kesimpulan

Kreasi

- a) **Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?**

Saya telah mempelajari lebih dalam mengenai constructor dan method secara lebih detail. Hal ini menjadikan saya lebih tau lagi mengenai fungsi dan macam – macam method dan constructor.

Nama & NPM	Topik:	Tanggal:
Hedy Rafian Firdaus G1F024027	Extends	17 September 2024

[No. 4] Identifikasi Masalah:

1) 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!

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

Kelas Java Bagian 2 - Method dan Extends

Source : <https://youtu.be/6qULMlcv-eg>

[No. 4] Analisis dan Argumentasi**1) Uraikan rancangan solusi yang diusulkan.**

1. Bandingkan method yang dimiliki class Anak extends Ortu dengan method di class Ortu!
 - a) Metode yang Ada di Kelas Ortu:
 - metodeOrtu(): Metode ini hanya ada di Ortu. Hanya dapat digunakan oleh objek Ortu atau objek Anak.
 - metodeOverride(): Metode ini ada di Ortu dan bisa diubah di Anak menggunakan method overriding.
 - b) Metode yang Ada di Kelas Anak:
 - metodeAnak(): Metode ini hanya ada di Anak. Tidak ada di Ortu, jadi hanya dapat digunakan oleh objek Anak.
 - metodeOverride(): Metode ini di-override dari Ortu, sehingga ketika dipanggil pada objek Anak, implementasi yang baru (di Anak) yang akan dijalankan.
2. Untuk kode program kali ini, saya akan menambahkan extends untuk anak ke-2 dan membuat method yang berbeda juga. Untuk anak ke-2 ini, saya akan menambahkan method sukaMakan dan sukaMain.

[No. 4] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 - a) Mulai
 - b) Membuat nama kelas dan nama file Ortu
 - c) Membuat method spesifik
 - d) Membuat method umum yang bisa diubah anak
 - e) Membuat class Anak extends Ortu
 - f) Membuat Overloaded untuk mengisi variabel
 - g) Membuat Overriden untuk mengisi variabel
 - h) Lakukan hal yang sama untuk class Anak2 extends Ortu (dari e – g)
 - i) Menghasilkan luaran
 - j) Selesai

2) Tuliskan kode program dan luaran

```
public class Ortu {
    // Method spesifik
    void sukaMenonton(String a) {
        System.out.println("Nonton " + a);
    }

    // Method umum yang bisa diubah anak
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }

    public static void main(String[] args) {
        System.out.println("Sifat Orang Tua :");
        Ortu objekO = new Ortu();
        objekO.sukaMenonton("Berita");
        objekO.sukaMembaca("Koran");

        System.out.println("\nSifat Anak :");
        Anak objekA = new Anak();
        objekA.sukaMenonton(9, "Film Drakor"); // Overloaded method in Anak
        objekA.sukaMembaca("Komik One Piece");

        System.out.println("\nSifat Anak2 :");
        Anak2 objekB = new Anak2();
        objekB.sukaMakan(10, "Nasi Padang campur Caviar"); // Overloaded method in Anak2
        objekB.sukaMain("Genshin Impact");
    }
}

class Anak extends Ortu {
    // Overloaded method
    void sukaMenonton(int a, String b) {
        System.out.println("Nonton Jam " + a + " Malam " + b);
    }

    // Overridden method
    @Override
    void sukaMenonton(String a) {
        System.out.println("Nonton " + a);
    }

    // Overridden method
    @Override
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }
}

class Anak2 extends Ortu {
    // Overloaded method
    void sukaMakan(int a, String b) {
        System.out.println("Makan " + a + " piring " + b);
    }

    // Overridden method
    @Override
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }

    // New method specific to Anak2
    void sukaMain(String a) {
        System.out.println("Suka main game " + a);
    }
}
```

a) Beri komentar pada kode

Kode yang dibuat telah sesuai dengan syarat kode yang digunakan sehingga akan menghasilkan luaran yang sesuai dengan yang kita program.

b) Uraikan luaran yang dihasilkan

```
java -cp /tmp/5Mq0hBwWy0/Ortu
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

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

Sifat Anak2 :
Makan 10 piring Nasi Padang campur Caviar
Suka main game Genshin Impact

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

Luaran yang dihasilkan telah sesuai dengan kode program yang dilakukan. Hasil luaran yang didapat untuk Anak2 adalah :

Sifat Anak2 :
Makan 10 piring Nasi Padang campur Caviar
Suka main game Genshin Impact

[No. 4] Kesimpulan

Analisa

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

Untuk pemrograman kali ini, saya membuat method yang berbeda untuk anak2, yang dimana method yang Anak2 punya berbeda dengan yang Ortu dan Anak punya, sehingga hasilnya pun menjadi berbeda satu sama lain.

Refleksi :

Setelah saya mengerjakan tugas ini, saya jadi belajar mengenai class, objek, method, dan extends. Jujur saja, ini lumayan membingungkan awalnya, bahkan saya pun mengerjakannya hingga tenggat waktu yang mepet. Tetapi, saya mencoba berusaha semaksimal mungkin sehingga saya pun bisa mengerjakannya hingga selesai.