

Nama & NPM	Topik:	Tanggal:
Donna Zelvia Gizella Alvera Deadora G1F024018	Kelas Java	16 September 2024

Unit 1 Kelas (Class)

[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");
    }
}
```

Penjelasan :

Pada program tersebut, kode tidak memiliki metode main sehingga tidak dapat menjalankan program dan tidak ada luaran yang ditampilkan. Kode yang diperlukan untuk menampilkan sebuah luaran adalah metode main public static void main (String[] args) {manusia satu = new manusia ();}. Constructor ini tidak memiliki parameter, sehingga objek yang dibuat tidak memiliki informasi yang spesifik dan hanya mencetak pesan "Kelas Manusia tanpa nama".

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

<https://www.youtube.com/watch?v=60IdOc8m8Es>

Latihan 1:

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

Penjelasan :

Atribut variabel adalah data atau informasi yang disimpan pada objek tersebut. Ciri-ciri umum yang dapat menjadi atribut variabel adalah nama yang memberikan informasi berupa nama dari kelas tersebut dan rambut yang menjelaskan salah satu ciri-ciri fisik dari objek tersebut. Perilaku atau behavior adalah yang menggambarkan apa yang dilakukan oleh objek tersebut. Pada program tersebut, tidak ada metode yang digunakan untuk menampilkan pesan ini. Tetapi, ciri-ciri umum yang dapat menjadi perilaku atau behavior contohnya seperti makan, menonton, memasak, mendaki, dan lain-lain menggambarkan berbagai aktivitas yang bisa dilakukan oleh objek Manusia.

[No.1] Analisis dan Argumentasi

Pada contoh 1 merupakan kode sederhana dari kelas java yang menunjukkan cara membuat objek dan menggunakan constructor untuk mencetak pesan. Variabel nama dan rambut akan menyimpan informasi dari objek. Kode public manusia(): adalah constructor atau metode khusus yang dipanggil saat objek manusia dibuat. Pada program diatas constructor ini tidak memiliki parameter.

[No.1] Penyusunan Algoritma dan Kode Program

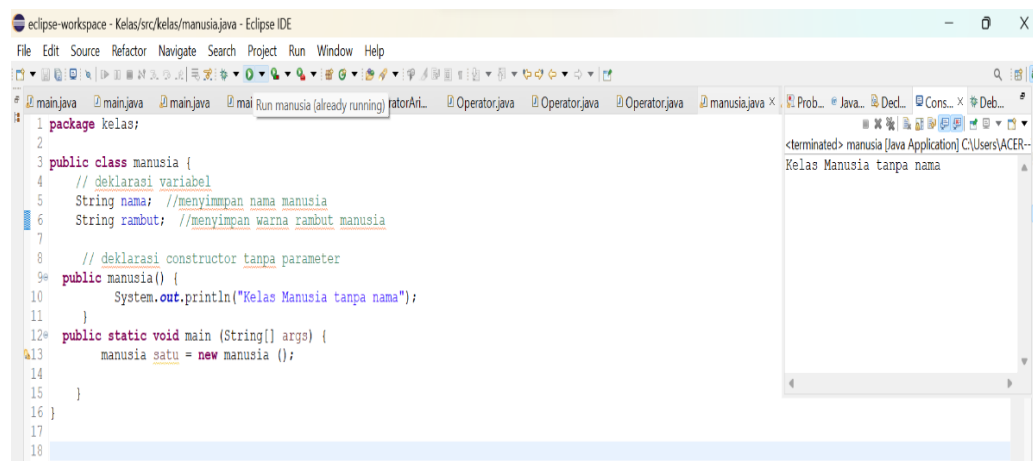
1) Algoritma

- (a) Mulai
- (b) Deklarasi kelas Manusia
- (c) Deklarasi atribut
- (d) Deklarasi constructor tanpa parameter
- (e) Membuat main method
- (f) Selesai

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



```
1 package kelas;
2
3 public class manusia {
4     // deklarasi variabel
5     String nama; //menyimpan nama manusia
6     String rambut; //menyimpan warna rambut manusia
7
8     // deklarasi constructor tanpa parameter
9     public manusia() {
10         System.out.println("Kelas Manusia tanpa nama");
11     }
12     public static void main (String[] args) {
13         manusia satu = new manusia ();
14     }
15 }
16
17
18
```

The screenshot shows the Eclipse IDE with a Java project named 'kelas'. The source code for 'manusia.java' is displayed, featuring a package declaration, class definition, attributes, a no-parameter constructor, and a main method. The console output on the right shows the message 'Kelas Manusia tanpa nama'.

- b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan yaitu 'Kelas Manusia tanpa nama' karena constructor tidak memiliki parameter.

[No.1] Kesimpulan

1) Analisa

Pada program tersebut, karena constructor tidak memiliki parameter, maka program hanya menampilkan luaran 'Kelas Manusia tanpa nama'. Atribut dari kelas tersebut meliputi nama untuk menyimpan nama dari objek manusia dan rambut untuk menyimpan warna rambut dari objek manusia.

Unit 2 Objek

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh 2: Salin dan tempel kode program berikut ke Eclipse atau JDoodle.

```
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

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

<https://www.youtube.com/watch?v=60IdOc8m8Es>

Penjelasan :

Pada program tersebut, kita diminta untuk menyusun kembali kode tersebut, lalu menambahkan data berupa ciri-ciri yang dan Saya menambahkan warna mata, warna rambut, tinggi badan, hobi, dan lain-lain.

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?

Penjelasan :

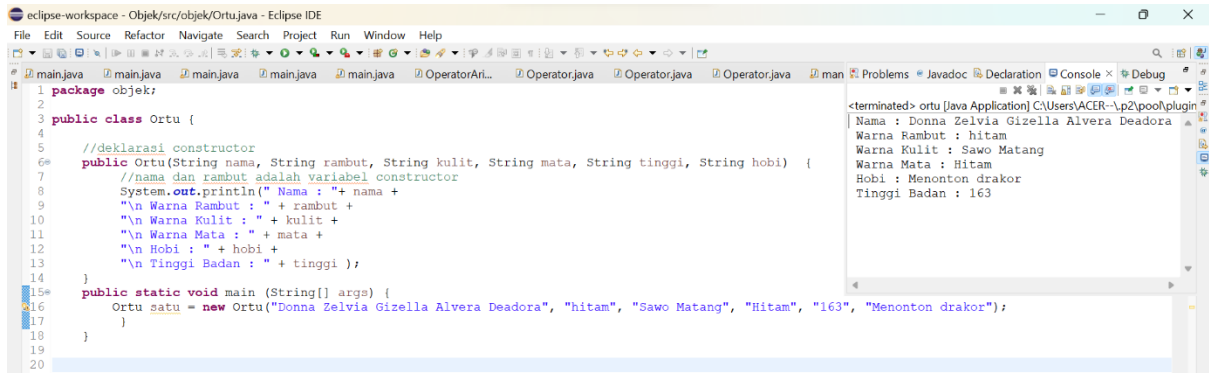
Apabila nanti memiliki keturunan, atribut yang dapat diturunkan berupa warna rambut, warna kulit, warna mata, tinggi, dan hobi. Kelas keturunan dapat memiliki constructor yang sama dengan constructor dari kelas ortu untuk menginisialisasi atribut yang diwarisi. Perilaku positif (behavior) yang diturunkan adalah menggambarkan aktivitas yang dilakukan oleh objek yang akan diturunkan, sehingga keturunan memiliki perilaku yang sama dengan orang tua. Seperti menonton drakor ataupun yang lainnya.

[No.2] Analisis dan Argumentasi

Untuk data berupa ciri-ciri, Saya mengusulkan untuk menambahkan data berupa warna mata, warna kulit, hobi, dan tinggi badan yang mana itu akan diturunkan pada kelas anak. Constructor pada kelas anak juga bisa menggunakan constructor yang sama dengan kelas ortu.

[No.2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas
 - (c) Deklarasi Atribut
 - (d) Deklarasi constructor
 - (e) Membuat main method
 - (f) Menampilkan hasil
 - (g) Selesai
 - 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
- Beri komentar pada kode yang di Screenshot



```
1 package objek;
2
3 public class Ortu {
4
5     //deklarasi constructor
6     public Ortu(String nama, String rambut, String kulit, String mata, String tinggi, String hobi) {
7         //nama dan rambut adalah variabel constructor
8         System.out.println(" Nama : " + nama +
9             "\n Warna Rambut : " + rambut +
10            "\n Warna Kulit : " + kulit +
11            "\n Warna Mata : " + mata +
12            "\n Hobi : " + hobi +
13            "\n Tinggi Badan : " + tinggi );
14     }
15     public static void main (String[] args) {
16         Ortu satu = new Ortu("Donna Zelvia Gizella Alvera Deadora", "hitam", "Sawo Matang", "Hitam", "163", "Menonton drakor");
17     }
18 }
19
20
```

<terminated> ortu [Java Application] C:\Users\ACER-Lp2\poo\plugin
Nama : Donna Zelvia Gizella Alvera Deadora
Warna Rambut : hitam
Warna Kulit : Sawo Matang
Warna Mata : Hitam
Hobi : Menonton drakor
Tinggi Badan : 163

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai, yaitu informasi dari kelas ortu yang meliputi nama, warna rambut, warna mata, warna kulit, hobi, dan tinggi badan.

[No.2] Kesimpulan

1) Analisa

Program telah berhasil diperbarui dengan menambahkan atribut baru yang meliputi mata, kulit, hobi, dan tinggi lalu menampilkan sebuah informasi. Constructor dari kelas ortu dapat digunakan juga pada kelas anak. Program ini memberikan pemahaman yang baik mengenai konsep pewarisan dalam pemrograman.

Unit 3 Method

[No. 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

Contoh 3: Salin dan tempel kode program berikut ke Eclipse atau JDoodle.

```
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

Penjelasan :

Pada program tersebut, atributnya meliputi nama dan rambut, void sukaNonton(String film) merupakan method atau kegiatan yang dilakukan objek, dan public static void main(String[] args) digunakan untuk mencetak atau memanggil method.

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

<https://www.youtube.com/watch?v=6qULMlcv-eg>

Latihan 3:

- 3.1. Analisa perbedaan deklarasi constructor, method, dan method utama!

Penjelasan :

- a) Constructor adalah method khusus yang didefinisikan kedalam kelas dan akan dipanggil secara otomatis. Deklarasi constructor diawali dengan membuat kelas dan nama pada constructor harus sesuai dengan nama kelasnya. Lalu menentukan parameter dari constructor dan parameter boleh ada atau tidak. Pada program tersebut, public Manusia1(String nama, String rambut) berfungsi untuk melakukan inisialisasi nilai atau parameter terhadap data-data yang terdapat pada kelas. Lalu akan dicetak dengan System.out.println. Menggunakan tipe data seperti String, tetapi tidak memerlukan void.
- b) Method adalah kegiatan, aksi, aktivitas, atau perilaku yang dilakukan oleh objek. Hal pertama yang dilakukan yaitu menentukan nama method dan pada kode program yaitu sukaNonton. Lalu, String film merupakan parameter yang diterima

oleh method. `System.out.println` akan mencetak hobi menonton dan jenis film yang ditonton. Menggunakan tipe data seperti `String`, `int`, `char` dan memerlukan `void`.

- c) Method utama atau main method adalah titik masuk program. Memiliki kelas `public` yang dapat diakses dari mana saja, membuat kelas baru, memanggil constructor kelas, mendeklarasikan variabel, dan mencetak informasi tentang objek.

3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?

Penjelasan :

- a) Constructor digunakan untuk membuat objek baru dari suatu kelas menginisialisasi atribut, biasanya dipanggil dengan kata kunci `"new"`, dan berjalan otomatis.
- b) Method digunakan untuk menentukan perilaku atau kegiatan dari suatu objek, memanggil objek yang sudah ada, dapat diakses berkali-kali, dan dapat mengembalikan nilai.
- c) Method digunakan untuk menentukan perilaku atau kegiatan dari suatu objek, memanggil objek yang sudah ada, dapat diakses berkali-kali, dan dapat mengembalikan nilai.

3.3. Uraikan perbedaan berikut :

- a) constructor overloading dan overriding

Penjelasan :

Constructor overloading adalah suatu cara membuat lebih dari constructor pada suatu class. Constructor overloading mengizinkan lebih dari satu konstruktor dalam sebuah kelas dengan parameter yang berbeda. Sedangkan constructor overriding tidak ada dalam Java. Constructor tidak dapat diwarisi dan tidak bisa di-overrides seperti metode. Namun, subclass dapat memiliki konstruktor sendiri yang berbeda dari superclass.

- b) method overloading, dan method overriding

Method overloading adalah suatu cara membuat lebih dari method pada suatu class. Supaya pengaksesan method tersebut lancar, maka sebagai pembedanya adalah tipe parameter dan atau jumlah parameternya. Method overriding adalah sebuah metode yang dipakai untuk kelas induk dan nantinya akan dipakai untuk

mendefinisikan ulang dengan kelas turunan menggunakan nama metode serta parameter-parameter yang sama.

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

Method yang mengembalikan nilai menggunakan pernyataan return untuk menunjukkan nilai tertentu yang akan dikembalikan ke pemanggil. Nilai yang dikembalikan dari suatu fungsi disebut return value. Bisa berupa tipe data primitif (seperti int, double, char, dll.) atau objek dari kelas. Method tidak mengembalikan nilai disebut void method, yaitu method yang bertipe void dan tidak mengembalikan nilai apapun. Method void tidak perlu berisi pernyataan return, tetapi boleh saja.

[No.3] Analisis dan Argumentasi

Constructor merupakan metode khusus untuk menginisialisasi sebuah objek dan pada program tersebut meliputi nama dan rambut. Method adalah fungsi yang melakukan tindakan atau mengembalikan nilai dan pada program yaitu void sukaNonton(String film), yang mencetak hobi menonton film. Method utama digunakan untuk memanggil objek dan kode programnya yaitu public static void main(String[] args).

[No.3] Penyusunan Algoritma dan Kode Program

1) Algoritma

- (a) Mulai
- (b) Membuat kelas
- (c) Deklarasi Atribut
- (d) Deklarasi constructor
- (e) Deklarasi method
- (f) Membuat main method
- (g) Menampilkan hasil
- (h) Selesai

2) Kode program dan luaran

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


```

1 package method;
2
3 public class Manusia {
4     //deklarasi atribut Manusia dalam variabel
5     String nama, rambut;
6
7     //deklarasi constructor
8     public Manusia(String nama, String rambut) {
9         System.out.println(" Nama saya : " + nama +
10            "\n Warna Rambut : " + rambut);
11     }
12
13     //deklarasi method
14     void sukaMenonton(String film) {
15         System.out.println(" Hobi Menonton : " + film);
16     }
17
18     //deklarasi method utama
19     public static void main( String[] args) {
20         Manusia satu = new Manusia("Putri", "hitam");
21         satu.sukaMenonton("Drakor");
22     }
23 }

```

Console Output:

```

<terminated> Manusia [Java Application] C:\Users\ACER\AppData\Local\Temp\p2pool\p2
Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor

```

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan yaitu menampilkan informasi dari objek yang meliputi nama, warna rambut, dan hobi menonton.

[No.3] Kesimpulan

1) Analisa

Program tersebut merupakan penerapan dari method yaitu perilaku yang dilakukan objek. Atribut atau variabel yang digunakan dalam sebuah kelas untuk menyimpan data atau informasi tentang objek yang dibuat dari kelas tersebut berupa nama dan rambut. Constructor berfungsi untuk menginisialisasi objek, method berfungsi untuk menerima parameter, dan method utama digunakan untuk memanggil method.

Unit 4 Extends

[No. 4] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

Contoh 4: Salin dan tempel kode program berikut ke JDoodle. Kemudian catat waktu eksekusinya.

```

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

Penjelasan :

Pada kode program tersebut kita diminta untuk menambahkan objek anak tetapi dengan method yang berbeda. Selain itu juga harus membandingkan method yang dimiliki kelas ortu dengan method kelas anak.

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

<https://www.youtube.com/watch?v=6qULMlcv-eg>

<https://frieypadie.web.id/constructor-pada-java/>

Latihan 4:

4.1. Bandingkan method yang dimiliki `class Anak extends Ortu` dengan method di `class Ortu`!

Penjelasan :

Method yang dimiliki oleh kelas anak dengan method yang dimiliki kelas ortu adalah sama tetapi yang membedakan yaitu parameternya. Di kelas induk menampilkan pesan tentang apa yang ditonton yaitu berita. Pada kelas anak, pesan yang ditampilkan sama, tetapi dapat mengoverride untuk menampilkan pesan yang sama dan juga memiliki overload method dengan parameter tambahan, misalnya `sukaMenonton(int a, String b)` yang menerima jam tayang dan jenis film. Method `sukaMembaca(String a)`, di Kelas Induk (Ortu) menampilkan pesan tentang bacaan yang disukai dan pada kelas anak method ini untuk menampilkan pesan yang sama.

4.2. Ubahlah Contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

[No.4] Analisis dan Argumentasi

Pada program tersebut, kelas anak mewarisi method dari kelas induk dan bisa mengubah atau menambah fungsinya. Method `sukaMenonton` di kelas anak memiliki parameter berbeda, sehingga bisa disesuaikan dengan kebutuhan, sementara `sukaMembaca` pada kelas anak dan kelas ortu tetap sama dan tidak memiliki tambahan, hanya apa yang dibaca saja yang membedakan. Selain itu, saya mengusulkan untuk menambahkan objek baru dan method baru yaitu `sukaOlahraga` yang hanya dimiliki kelas anak.

[No.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Membuat kelas ortu
 - (c) Membuat kelas anak extends ortu
 - (d) Deklarasi method
 - (e) Membuat main method
 - (f) Menampilkan hasil
 - (g) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
Berikan komentar pada kode yang di Screenshot

The screenshot shows the Eclipse IDE with a Java project named 'extends'. The main editor displays the source code for two classes: `Ortu` and `Anak`. `Ortu` is the superclass with methods `sukaMenonton` and `sukaMembaca`. `Anak` extends `Ortu` and overrides `sukaMenonton` while adding a new method `sukaOlahraga`. The `main` method in `Ortu` creates an `Ortu` object and calls its methods. The `main` method in `Anak` creates an `Anak` object, calls its methods, and then creates an `Anak` object with specific attributes and calls its methods. The console window on the right shows the output of the program, displaying the attributes of the objects and the results of the methods.

```
package extends;

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);
    }
    void sukaOlahraga(String a) { // method induk umum bisa diubah anak
        System.out.println("Suka Olahraga " + 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 Pertama :");
    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

    System.out.println("\n Sifat Anak kedua :");
    Anak objekB = new Anak(); // memanggil objek anak
    objekB.sukaMenonton(2, "Kartun Spongebob"); // memanggil sifat spesifik anak yang diturunkan induk
    objekB.sukaOlahraga("Bulutangkis"); // memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
}
```

Console Output:

```
<terminated> Anak [Java Application] C:\Users\ACER\AppData\Local\Temp\plugin
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

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

Sifat Anak kedua :
Nonton Jam 2 Malam Kartun Spongebob
Suka Olahraga Bulutangkis
```

b) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan menampilkan informasi sifat orang tua yang diturunkan ke anak, objek baru juga sudah ditambahkan sesuai dengan perintah pada soal.

[No.4] Kesimpulan

1) Analisa

Program ini berhasil menunjukkan pewarisan method dari kelas induk ke kelas anak. Tidak semua method induk diwariskan ke anak. Penambahan objek dan method baru di kelas anak, seperti `sukaOlahraga`.

Refleksi

Pada tugas kali ini, Saya mendapatkan pengalaman dan pengetahuan baru. Mulai dari pemahaman tentang kelas, method, objek dan extends. Bukan hanya itu, Saya juga jadi memiliki pemahaman mengenai constructor. Dalam mengerjakan tugas ini, sama seperti sebelumnya, terdapat beberapa kesulitan yang Saya alami karena kurangnya pemahaman mengenai materi tersebut. Tetapi, berkat usaha Saya untuk memahami materinya, baik melalui

jurnal, ebook, atau youtube dan juga bantuan dari teman-teman, Saya bisa menyelesaikan tugas ini. Untuk kedepannya, Saya akan berusaha lebih keras lagi dalam memahami setiap materi yang ada.