

Nama & NPM	Topik:	Tanggal:
Muhammad Hovid Arman G1F024034	Kelas, Objek, Method	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
 - a. Atribut variabel.
 - b. Perilaku/ behavior untuk method.

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun atribut variabel dan perilaku method dari kelas Manusia untuk Manusia dengan jelas.
- 2) Alasan solusi ini karena
 - a. Penyusunan variabel atribut berfungsi untuk merepresentasikan sifat dari suatu objek pada suatu kelas.
 - b. Penyusunan method pada program berfungsi untuk merepresentasikan perilaku objek pada suatu kelas.
- 3) Perbaiki kode program dengan cara
 - a. Penyusunan program untuk variabel nama dan variabel rambut

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - (a) Mulai di JDoodle online Java compiler
 - (b) Deklarasikan kelas
 - (c) Definisikan metode
 - (d) Susun variabel nama dan rambut
 - (e) Jalankan dan akhiri program
- 2) Kode program dan luaran
 - 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)
 - 9) `System.out.println("Kelas Manusia tanpa nama" + nama + rambut);`

```

10)    }
11)    public static void main(String[] args) {
12)        Manusia tes = new Manusia();
13)
14)        System.out.println("Kelas Manusia dengan nama" + tes.nama +
        tes.rambut);
15)
16)    }
17) }

```

Luaran:

Kelas Manusia tanpa namanullnull

Kelas Manusia dengan namanullnull

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

The screenshot shows an online Java compiler interface. On the left, the code editor contains the following Java code:

```

1 public class Manusia { // deklarasi kelas Manusia
2     // deklarasi variabel/atribut kelas
3     String nama; // variabel untuk menyimpan nama manusia
4     String rambut; // variabel untuk menyimpan jenis rambut manusia
5
6     // deklarasi constructor tanpa parameter
7     public Manusia() {
8         // karena nama dan rambut tidak diinisialisasi, nilainya default (null)
9         System.out.println("Kelas Manusia tanpa nama" + nama + rambut);
10        // mencetak "Kelas Manusia tanpa nama" diikuti nilai null untuk nama dan rambut
11    }
12
13    public static void main(String[] args) {
14        // main method untuk menjalankan program
15        Manusia tes = new Manusia(); // membuat objek baru dari kelas Manusia
16
17        // mencetak nilai dari variabel nama dan rambut objek "tes"
18        System.out.println("Kelas Manusia dengan nama" + tes.nama + tes.rambut);
19        // karena nama dan rambut tidak diberi nilai, hasilnya null untuk keduanya
20    }
21 }
22

```

On the right side, the 'Input/Output' panel shows the output of the program:

```

Kelas Manusia tanpa namanullnull
Kelas Manusia dengan namanullnull

```

At the bottom of the output panel, it states: 'Compiled and executed in 1.867 sec(s)'.

- b) Analisa luaran yang dihasilkan

Luaran menghasilkan Kelas Manusia tanpa nama dan Kelas manusia dengan nama karena tidak ada pendeklarasian nilai dari variabel.

[No.1] Kesimpulan

1) Analisa

Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi

a. Atribut variabel.

1. Variabel nama = untuk merepresentasikan nama dari objek di kelas Manusia.
2. Variabel rambut = untuk merepresentasikan warna rambut dari objek di kelas Manusia.

b. Perilaku/ behavior untuk method.

1. Perilaku dari objek di representasikan oleh fungsi Method, aksi atau Tindakan yang bisa dilakukan oleh objek manusia.

[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

```

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

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode pada contoh 2 lalu menambahkan ciri-ciri data.
- 2) Alasan solusi ini karena atribut variabel dalam constructor berfungsi untuk merepresentasikan ciri-ciri Objek.
- 3) Perbaiki kode program dengan cara penyusunan menambahkan ciri-ciri seperti tinggi dan warna kulit objek.

[No.2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - (a) Mulai di JDoodle online Java compiler
 - (b) Deklarasikan kelas
 - (c) Deklarasi constructor
 - (d) Definisikan method
 - (e) Susun nilai variabel
 - (f) Jalankan dan akhiri program
- 2) Kode program dan luaran
 - 1) `public class Ortu {`
 - 2) `// Deklarasi constructor, method khusus yang akan dipanggil saat objek kelas dibuat`
 - 3) `public Ortu(String nama, String rambut, String warnakulit, String warnamata, String bentukrambut) {`
 - 4) `// Variabel nama, rambut, warnakulit, warnamata, dan bentukrambut adalah parameter yang diterima oleh constructor`
 - 5) `// Constructor ini akan mencetak informasi tentang nama dan karakteristik fisik lainnya`
 - 6) `System.out.println("Nama saya: " + nama +`
 - 7) `"\nWarna Rambut: " + rambut +`
 - 8) `"\nWarna Kulit: " + warnakulit +`
 - 9) `"\nWarna Mata: " + warnamata +`
 - 10) `"\nBentuk Rambut: " + bentukrambut);`
 - 11) `}`
 - 12)
 - 13) `// Method main, sebagai titik awal program untuk dieksekusi`

```

14)    public static void main(String[] args) {
15)        // Membuat objek baru dari kelas Ortu dengan menggunakan
        constructor yang telah didefinisikan
16)        Ortu satu = new Ortu("Muhammad Hovid Armmman", "hitam", "sawo
        matang", "hitam", "ikal");
17)    }
18) }

```

Luaran:

Nama saya : Muhammad Hovid Armmman

Warna Rambut : hitam

Warna Kulit : sawo matang

Warna Mata : hitam

Bentuk Rambut : ikal

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

```

1- public class Ortu {
2-     // Deklarasi constructor, method khusus yang akan dipanggil saat objek kelas dibuat
3-     public Ortu(String nama, String rambut, String warnakulit, String warnamata, String bentukrambut) {
4-         // Variabel nama, rambut, warnakulit, warnamata, dan bentukrambut adalah parameter yang diterima oleh constructor
5-         // Constructor ini akan mencetak informasi tentang nama dan karakteristik fisik lainnya
6-         System.out.println("Nama saya: " + nama +
7-             "\nWarna Rambut: " + rambut +
8-             "\nWarna Kulit: " + warnakulit +
9-             "\nWarna Mata: " + warnamata +
10-            "\nBentuk Rambut: " + bentukrambut);
11-     }
12-
13-     // Method main, sebagai titik awal program untuk dieksekusi
14-     public static void main(String[] args) {
15-         // Membuat objek baru dari kelas Ortu dengan menggunakan constructor yang telah didefinisikan
16-         Ortu satu = new Ortu("Muhammad Hovid Armmman", "hitam", "sawo matang", "hitam", "ikal");
17-     }
18- }
19-
Output    Generated Files

Nama saya: Muhammad Hovid Armmman
Warna Rambut: hitam
Warna Kulit: sawo matang
Warna Mata: hitam
Bentuk Rambut: ikal

Compiled and executed in 1.922 sec(s)

```

- b) Analisa luaran yang dihasilkan

Luaran menghasilkan ciri ciri dari Muhammad Hovid Arman seperti memiliki nama, warna rambutnya yang hitam, warna kulit nya sawo matang, warna mata Hitam, dan bentuk rambutnya yang ikal.

[No.2] Kesimpulan

1) Analisa

Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan?

Jika seorang induk atau orang tua memiliki keturunan, maka:

- Keturunan tersebut akan mewariskan sifat-sifat dari induknya.
- Constructor yang akan diturunkan seperti warna rambut, warna kulit, warna mata, dan bentuk rambut.

- Induk atau orang tua akan menjadi teladan sekaligus contoh bagi sang keturunan, maka dari itu penting bagi orang tua untuk membiasakan penerapan perilaku positif agar dapat diturunkan kepada sang anak seperti rajin, berbudi pekerti, bersikap baik, beradab dan lain-lain.

[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
```

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

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode seperti pada contoh 3 dan mengganti nama constructor menjadi sama dengan nama kelas.
- 2) Alasan solusi ini karena karena constructor bertugas untuk menginisialisasi objek dari kelas tersebut.
- 3) Perbaiki kode program dengan cara.
 - Mengubah constructor Manusia1 menjadi sesuai dengan nama kelas, jika kelas Manusia maka ubah jadi Manusia.

[No.3] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Mulai di JDoodle online Java compiler
- (b) Deklarasikan kelas
- (c) Deklarasi constructor
- (d) Definisikan method
- (e) Definisikan main method
- (f) Susun nilai parameter
- (g) Jalankan dan 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 sukaNonton(String film) {
13)        System.out.println(" Hobi Menonton : " + film);
14)    }
15)
16)    //deklarasi method utama
17)    public static void main( String[] args) {
18)        Manusia satu = new Manusia("Muhammad Hovid Armman","Hitam");
19)        satu.sukaNonton("Anime");
20)    }
21) }
```

Luaran:

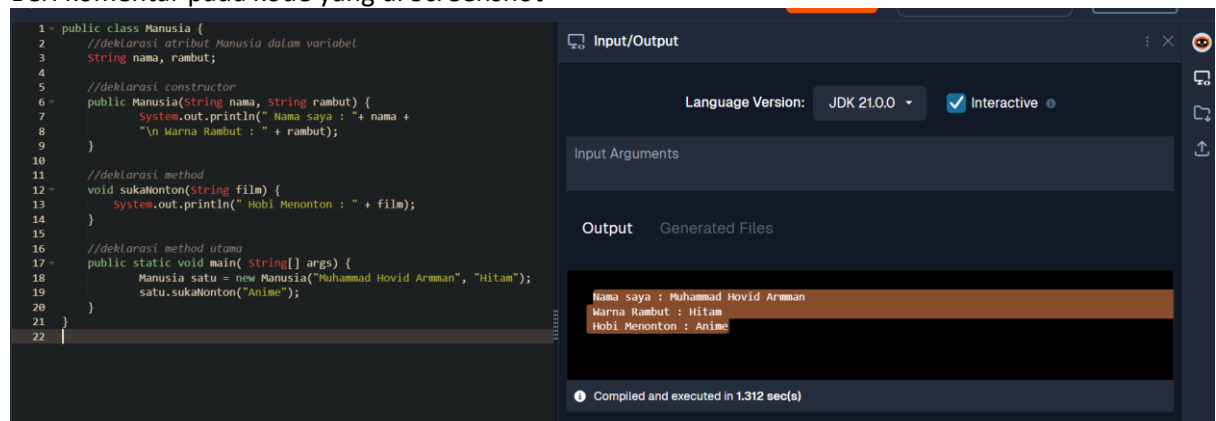
Nama saya : Muhammad Hovid Armman

Warna Rambut : Hitam

Hobi Menonton : Anime

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



- b) Analisa luaran yang dihasilkan
Luaran menghasilkan tampilan
Nama saya : Muhammad Hovid Armman
Warna Rambut : Hitam
Hobi Menonton : Anime
Menampilkan luaran yang sesuai dengan kode program yang disusun.

[No.3] Kesimpulan

1) Analisa

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

- Constructor adalah metod khusus yang berfungsi untuk melakukan penginisialisasian objek, nama constructor harus sama dengan nama kelas, tidak memiliki pengembalian nilai/return.
- Method adalah kode blok yang bisa dipanggil berkali-kali dan memiliki pengembalian nilai, seperti function.
- Method utama atau main method adalah titik masuk, titik awal, entry point dari suatu program, dimana objek diinisialisasi dan method dipanggil.

3.2. Tentukan kapan Anda perlu menggunakan constructor dan method?

- Constructor digunakan ketika kita ingin menciptakan objek dan Method digunakan Ketika kita ingin membuat objek tersebut melakukan sesuatu

3.3. Uraikan perbedaan berikut:

a) constructor overloading dan overriding

- Constructor overloading adalah kondisi ketika ada pendeklarasian constructor yang lebih dari satu dengan parameter berbeda
- Constructor overriding adalah kondisi ketika melakukan pendefinisian ulang dengan cara yang berbeda (tidak berlaku pada constructor).

b) method overloading, dan method overriding

- Method overloading adalah pendeklarasian beberapa method yang sama dengan parameter berbeda.
- Method overriding adalah pendeklarasian ulang method yang sudah ada di kelas induk menuju kelas turunan dengan cara yang berbeda, kelas turunan memiliki method yang sama dengan kelas induk.

c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

- Method yang mengembalikan nilai memiliki tipe selain void, seperti integer atau string.
- Method yang tidak mengembalikan nilai memiliki tipe void.

[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 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

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyusun kode seperti pada contoh 4 dan menambahkan kelas turunan ortu yaitu kelas anak untuk objek anak1 dan anak2.
- 2) Alasan solusi ini agar kita bisa lebih memahami tentang cara kerja dari kelas, objek, dan method, untuk menambah pengalaman sesuai dengan yang diminta soal.
- 3) Perbaiki kode program dengan cara.
 - Mengubah objek anak menjadi A1 dan menambahkan A2.
 - Menambahkan method `sukaBermainGame`.

[No.4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - (a) Mulai di JDoodle online Java compiler
 - (b) Deklarasikan kelas induk
 - (c) Definisikan method induk spesifik
 - (d) Definisikan method induk umum
 - (e) Definisikan main method
 - (f) Inisialisasi objek
 - (g) Susun nilai parameter untuk method objek
 - (h) Jalankan dan akhiri program

- 2) Kode program dan luaran

```
public class OrtuLagi {           // 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 :");
        OrtuLagi objekO = new OrtuLagi(); // memanggil objek induk
        objekO.sukaMenonton("Berita");    // memanggil sifat spesifik induk
        objekO.sukaMembaca("Koran");      // memanggil method dengan variabel
        dapat diubah

        System.out.println("\nSifat Anak1 :");
        Anak objekA1 = new Anak(); // memanggil objek anak untuk anak1
        objekA1.sukaMenonton(9, "Film Drakor"); // memanggil sifat spesifik
        anak yang diturunkan induk
        objekA1.sukaMembaca("Komik One Piece"); // memanggil method ke induk
        tanpa deklarasi ulang di anak
        objekA1.sukaBermainGame("Free Fire Max"); // memanggil method
        sukaBermainGame
    }
}
```

```

        System.out.println("\nSifat Anak2 :");
        Anak objekA2 = new Anak(); //memanggil objek anak untuk anak2
        objekA1.sukaBermainGame("Candy Crush"); //memanggil method
sukaBermainGame
    }

    static class Anak extends OrtuLagi {
        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 sukaBermainGame(String a) { // method sukaBermainGame
            System.out.println("Suka Bermain " + a);
        }
    }
}

```

Luaran:

Sifat Orang Tua :

Nonton Berita

Suka Baca Koran

Sifat Anak1 :

Nonton Jam 9 Malam Film Drakor

Suka Baca Komik One Piece

Suka Bermain Free Fire Max

Sifat Anak2 :

Suka Bermain Candy Crush

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

The screenshot shows an IDE with a Java file named 'OrtuLagi.java'. The code defines a base class 'OrtuLagi' and a subclass 'Anak'. The 'OrtuLagi' class has methods 'sukaMenonton' (with an overloaded version for specific input), 'sukaMembaca', and 'sukaBermainGame'. The 'Anak' class inherits from 'OrtuLagi' and overrides the 'sukaMenonton' method. The 'main' method creates instances of 'OrtuLagi' and 'Anak', and calls their methods to produce the output shown in the 'Output' window.

```

1- public class OrtuLagi { // 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-     OrtuLagi objek0 = new OrtuLagi(); // memanggil objek induk
12-     objek0.sukaMenonton("Berita"); // memanggil sifat spesifik induk
13-     objek0.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
14- }
15- System.out.println("\nSifat Anak1 :");
16- Anak objekA1 = new Anak(); //memanggil objek anak untuk anak1
17- objekA1.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
18- objekA1.sukaMembaca("Komik One Piece"); //memanggil method ke induk tanpa deklarasi ulang di anak
19- objekA1.sukaBermainGame("Free Fire Max"); //memanggil method sukaBermainGame
20- }
21- System.out.println("\nSifat Anak2 :");
22- Anak objekA2 = new Anak(); //memanggil objek anak untuk anak2
23- objekA1.sukaBermainGame("Candy Crush"); //memanggil method sukaBermainGame
24- }
25- }
26- static class Anak extends OrtuLagi {
27-     void sukaMenonton(int a, String b) {
28-         System.out.println("Nonton Jam " + a + " Malam " + b);
29-     }
30-     void sukaMenonton(String a) { // method induk spesifik
31-         System.out.println("Nonton " + a);
32-     }
33-     void sukaMembaca(String a) { // method induk umum bisa diubah anak
34-         System.out.println("Suka Baca " + a);
35-     }
36-     void sukaBermainGame(String a) { // method sukaBermainGame
37-         System.out.println("Suka Bermain " + a);
38-     }
39- }
40- }

```

The 'Output' window shows the following results:

```

Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak1 :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
Suka Bermain Free Fire Max

Sifat Anak2 :
Suka Bermain Candy Crush

```

Compiled and executed in 1.877 sec(s)

- b) Analisa luaran yang dihasilkan
Luaran menghasilkan tampilan

Sifat Orang Tua : yang merupakan objek kelas induk
Nonton Berita : spesifik
Suka Baca Koran : umum

Sifat Anak1 : kelas turunan dari kelas induk
Nonton Jam 9 Malam Film Drakor : method overloading
Suka Baca Komik One Piece : method overriding
Suka Bermain Free Fire Max : method baru yang tidak ada di class ortu

Sifat Anak2 :
Suka Bermain Candy Crush

Menampilkan luaran yang sesuai dengan kode program yang disusun.

[No.4] Kesimpulan

1) Analisa

- a) Bandingkan method yang dimiliki class Anak extends Ortu dengan method di class Ortu!
- Method suka Menonton pada class OrtuLagi memiliki parameter (String a), sedangkan methodnya pada class Anak extends ortu memiliki parameter berupa (int a , String b)
 - Method sukaMembaca pada classAnak extends Ortu melakukan overriding untuk menyesuaikan teks yang ditampilkan

Refleksi

Izin sebelumnya, mohon maaf, bagi saya materi cukup sulit, karena ada banyak istilah atau kata-kata yang sangat asing di telinga saya, seperti constructor, overloading, overriding, class induk umum dan spesifik. Saya masih sangat kesulitan untuk memahami hal-hal itu, namun perlahan saya mencari tahu untuk mempelajarinya. Saya juga masih kesulitan untuk membedakan constructor dengan yang lainnya.

Saya menemukan banyak kendala pada materi kali ini, namun saya juga menemukan banyak pengetahuan baru yang didapatkan.

Pada materi ini membahas empat macam hal, yaitu :

- Class : Kumpulan data-data dari semua objek pada pemrograman, tempat pendefinisian variabel dan method.
- Objek : hasil produk dari suatu kelas.
- Method : fungsi yang dipanggil untuk melakukan tugas tertentu yang bisa dipanggil berulang kali
- Constructor : Mekanisme proses yang digunakan untuk menginisialisasi objek dari class.