

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Risti Safira G1F024029		18 September 2024
[Nomor Soal] Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variable Untuk nomor 1 kesalahan nya ada di dalam kelas manusia1 yang memasukkan nama dan warna rambut ke dalam deklarasi yang hanya menerima satu parameter Untuk nomor 2 kesalahan nya adalah nama konstruktor yang tidak sama dengan nama kelas dan juga constructor tidak menerima dua data tetapi di dalam method dituliskan untuk dua data Nomor 3 latihan 3.1 adalah nama constructor seharusnya menggunakan awalan huruf kapital, dan juga membuat constructor yang menerima dua data Nomor 4. Ada beberapa hal yang perlu di evaluasi dan mencatat waktu eksekusi</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage) Belajar Java OOP: Memahami Konsep Dasar OOP (untuk Pemula) (petanikode.com) https://www.duniailkom.com/tutorial-oop-java-pengertian-class-object-property-dan-method/</p> <p>3) Uraikan rancangan solusi yang diusulkan (jika ada). Untuk nomor 1 solusinya adalah membuat konstruktor yang dapat menerima dua parameter</p> <p>4) Analisis susunan solusi, parameter solusi (jika ada).</p>		
[Nomor Soal] Analisis dan Argumentasi		
<p>1) Uraikan rancangan solusi yang diusulkan. Untuk nomor 1 solusinya adalah membuat constructor yang dapat menerima dua parameter Untuk Solusi nomor 2.1 adalah nama constructor seharusnya menggunakan awalan huruf kapital, dan juga membuat constructor yang menerima dua data pada 2.2 atribut yang akan di turunkan adalah nama, warna rambut, kepribadian, dan kebiasaan Nomor 3 : Constructor dan nama kelas harus sama dan menggunakan huruf awalan kapital Method “sukaNonton” harus didefinisikan dengan parameter yang sesuai. Nomor 4 : Latihan 4.1 Kelas Anak mendefinisikan ulang method sukaMenonton(String a) dan sukaMembaca(String a), padahal method tersebut sudah ada di kelas Ortu. Seharusnya jangan di tulis ulang karena tidak ada perubahan pada kelas anak (Kesimpulan nya agar lebih efisien, method nya tidak perlu di tulis ulang)</p> <p>2) Analisis solusi, kaitkan dengan permasalahan. Solusinya benar, karena dengan constructor yang bisa menerima dua parameter itu contoh ke 1 tadi bisa dijalankan</p>		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<p>1) Rancang desain solusi atau algoritma</p> <ol style="list-style-type: none">1. Mencari permasalahan dalam soal2. Memperbaiki kode yang salah3. Merancang kode atribut4. Membuat kode program sesuai perintah di soal <p>2) Tuliskan kode program dan luaran</p> <ol style="list-style-type: none">a) Beri komentar pada kode		

Untuk nomor 1 kesalahan nya ada di dalam kelas manusia1 yang memasukkan nama dan warna rambut ke dalam deklarasi yang hanya menerima satu parameter
solusinya adalah membuat constructor yang dapat menerima dua parameter

Untuk nomor 2 kesalahan nya adalah nama konstruktor yang tidak sama dengan nama kelas dan juga constructor tidak menerima dua data tetapi di dalam method dituliskan untuk dua data

solusinya adalah nama constructor seharusnya menggunakan awalan huruf kapital, dan juga membuat constructor yang menerima dua data

pada 2.2 atribut yang akan di turunkan adalah nama, warna rambut, kepribadian, dan kebiasaan

nomor 3. 2

a) Method Overloading

Method overloading memungkinkan kita menggunakan nama yang sama untuk beberapa method dengan parameter yang berbeda.

b) Method dengan Return Value

Method dengan return value mengembalikan hasil tertentu ke pemanggil.

c) Method Tanpa Return Value

Method tanpa return value hanya melakukan suatu tindakan tanpa mengembalikan hasil.

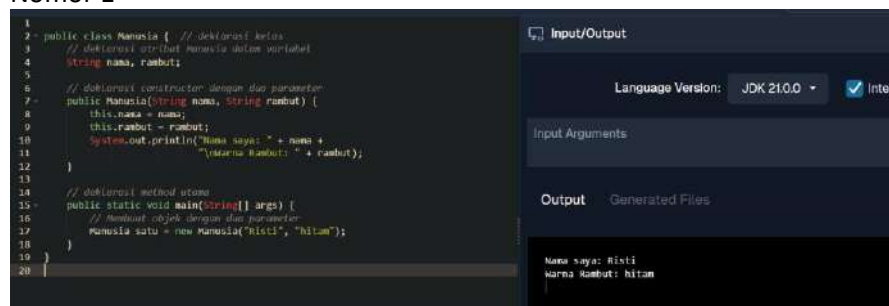
Nomor 4. Ada beberapa hal yang perlu di evaluasi

1. **Override yang Tidak Diperlukan:** Method yang sama persis di kelas Anak dan Ortu harus dihapus untuk efisiensi.
2. **Method Overloading:** Tetap efisien dan memberikan fleksibilitas tanpa konflik dengan method di kelas induk.
3. **Override yang Dibutuhkan:** Diperlukan ketika perilaku berbeda diinginkan, seperti pada method sukaMembaca.

b) Uraikan luaran yang dihasilkan

c) Screenshot/ Capture potongan kode dan hasil luaran

Nomor 1



```
1
2- public class Manusia { // deklarasi kelas
3- // deklarasi atribut manusia dalam variabel
4-     String nama, rambut;
5-
6- // deklarasi constructor dengan dua parameter
7-     public Manusia(String nama, String rambut) {
8-         this.nama = nama;
9-         this.rambut = rambut;
10-        System.out.println("Nama saya: " + nama +
11-        "Warna Rambut: " + rambut);
12-    }
13-
14- // deklarasi method utama
15-     public static void main(String[] args) {
16-         // membuat objek dengan dua parameter
17-         Manusia satu = new Manusia("Risti", "hitam");
18-     }
19- }
20
```

Input/Output

Language Version: JDK 21.0.0 ☒ Intera

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam

Nomor 2

Latihan 2.1

```

1 public class Ortu {
2     String nama, rambut;
3
4     // Deklarasi constructor yang benar
5     public Ortu(String nama, String rambut) {
6         this.nama = nama; // inisialisasi variabel nama
7         this.ambut = rambut; // inisialisasi variabel rambut
8         System.out.println("Nama saya: " + nama +
9             "\nWarna Rambut: " + rambut);
10    }
11
12    public static void main(String[] args) {
13        // Membuat objek dengan dua parameter
14        Ortu satu = new Ortu("Risti", "hitam");
15    }
16 }

```

Input/Output

Language Version:

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam

Latihan 2.3

```

1 public class Ortu {
2     String nama, rambut, sifat, hobi;
3
4     // Constructor 1: menerima nama dan rambut
5     public Ortu(String nama, String rambut) {
6         this.nama = nama;
7         this.ambut = rambut;
8         System.out.println("Nama saya: " + nama +
9             "\nWarna Rambut: " + rambut);
10    }
11
12    // Constructor 2: menerima nama, rambut, dan sifat
13    public Ortu(String nama, String rambut, String sifat) {
14        this.nama = nama;
15        this.ambut = rambut;
16        this.sifat = sifat;
17        System.out.println("Nama saya: " + nama +
18            "\nWarna Rambut: " + rambut +
19            "\nSifat: " + sifat);
20    }
21
22    // Constructor 3: menerima nama, rambut, sifat, dan hobi
23    public Ortu(String nama, String rambut, String sifat, String hobi) {
24        this.nama = nama;
25        this.ambut = rambut;
26        this.sifat = sifat;
27        this.hobi = hobi;
28        System.out.println("Nama saya: " + nama +
29            "\nWarna Rambut: " + rambut +
30            "\nSifat: " + sifat +
31            "\nHobi: " + hobi);
32    }
33 }

```

Input/Output

Language Version: JDK 21.0.0 ☒ Int

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam
Nama saya: Rista
Warna Rambut: hitam
Sifat: rajin
Nama saya: Siti
Warna Rambut: hitam
Sifat: rajin
Hobi: membaca

Compiled and executed in 1.501 sec(s)

```

33
34 public static void main(String[] args) {
35     // Memanggil constructor 1
36     Ortu ortu1 = new Ortu("Risti", "hitam");
37
38     // Memanggil constructor 2
39     Ortu ortu2 = new Ortu("Rista", "hitam", "rajin");
40
41     // Memanggil constructor 3
42     Ortu ortu3 = new Ortu("Siti", "hitam", "rajin", "membaca");
43 }
44 }
45

```

Nomor 3

Latihan 3.1

```

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         this.nama = nama;
8         this.ambut = rambut;
9         System.out.println("Nama saya: " + nama +
10             "\nWarna Rambut: " + rambut);
11    }
12
13    // method untuk menampilkan hobi menonton
14    void sukaNonton(String film) {
15        System.out.println("Hobi Menonton: " + film);
16    }
17
18    // method untuk menghitung jam nonton (overloading)
19    int sukaNonton(int episode, int durasi) {
20        return episode * durasi; // menghitung total jam menonton
21    }
22
23    // method utama
24    public static void main(String[] args) {
25        Manusia satu = new Manusia("Risti", "hitam");
26        satu.sukaNonton("Drakor"); // memanggil method dengan string
27        int jumlahJam = satu.sukaNonton(1, 2); // memanggil method dengan 2 parameter
28        System.out.println("Jam nonton = " + jumlahJam + " jam");
29    }
30 }
31

```

Input/Output

Language Version: JDK 21.0

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam
Hobi Menonton: Drakor
Jam nonton = 2 jam

Compiled and executed in 1.887 sec(s)

Latihan 3.2

```

1 public class Ortu {
2     String nama, rambut, sifat, hobi;
3
4     // constructor yang mempunyai semua atribut
5     public Ortu(String nama, String rambut, String sifat, String hobi) {
6         this.nama = nama;
7         this.rambut = rambut;
8         this.sifat = sifat;
9         this.hobi = hobi;
10        System.out.println("Nama saya: " + nama +
11        "Warna Rambut: " + rambut +
12        "Sifat: " + sifat +
13        "Hobi: " + hobi);
14    }
15
16    // Method tanpa return value. Menampilkan aktivitas
17    void aktivitas(String kegiatan) {
18        System.out.println(nama + " sedang melakukan aktivitas: " + kegiatan);
19    }
20
21    // method dengan return value. Menghitung waktu yang dihabiskan
22    int waktuAktivitas(int durasi, int frekuensi) {
23        return durasi * frekuensi; // menghitung total waktu aktivitas
24    }
25
26    // method overloading: versi yang menerima deskripsi tambahan
27    void aktivitas(String kegiatan, String tempat) {
28        System.out.println(nama + " sedang melakukan aktivitas: " + kegiatan + " di " + tempat);
29    }
30
31    public static void main(String[] args) {
32        // Membuat objek Ortu dengan data pribadi
33        Ortu ortu = new Ortu("Risti", "hitam", "rajin", "membaca");
34
35        // Memanggil method tanpa return value
36        ortu.aktivitas("membaca buku");
37
38        // Memanggil method dengan return value
39        int totalWaktu = ortu.waktuAktivitas(1, 3); // 3 jam per hari, 3 hari di
40        System.out.println("Total waktu aktivitas: " + totalWaktu + " jam");
41
42        // Memanggil method overloading
43        ortu.aktivitas("membaca buku", "perpustakaan");
44    }
45 }

```

Input/Output

Language Version: JDK 21.0.0 - ☒ Interactive

Input Arguments

Output Generated Files

```

Nama saya: Risti
Warna Rambut: hitam
Sifat: rajin
Hobi: membaca
Risti sedang melakukan aktivitas: membaca buku
Total waktu aktivitas: 3 jam
Risti sedang melakukan aktivitas: membaca buku di perpustakaan

```

● Compiled and executed in 1.884 sec(s)

Nomor 4

```

1 public class Ortu {
2     void sukaNonton(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        long startTime = System.currentTimeMillis(); // Catat waktu sebelum eksekusi
12
13        System.out.println("Sifat Orang Tua :");
14        Ortu ortu = new Ortu();
15        ortu.sukaNonton("youtube");
16        ortu.sukaMembaca("koran");
17
18        System.out.println("Sifat Anak :");
19        Anak objekA = new Anak();
20        objekA.sukaNonton("Film Drakor");
21        objekA.sukaMembaca("novel");
22
23        long endTime = System.currentTimeMillis(); // Catat waktu setelah eksekusi
24        System.out.println("Waktu eksekusi: " + (endTime - startTime) + " ms");
25    }
26 }
27
28 class Anak extends Ortu {
29     void sukaNonton(int a, String b) {
30         System.out.println("Nonton Jan " + a + " Malan " + b);
31     }
32 }
33

```

Input/Output

Language Version: JDK 21.0.0 - ☒ Interactive

Input Arguments

Output Generated Files

```

Sifat Orang Tua :
Nonton youtube
Suka Baca koran

Sifat Anak :
Nonton Jan 9 Malan Film Drakor
Suka Baca novel
Waktu eksekusi: 69 ms

```

● Compiled and executed in 1.906 sec(s)

[Nomor Soal] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
 Nomor 1 : Untuk memasukkan dua data sebaiknya menggunakan constructor yang dapat menerima dua data tersebut
 Nomor 2 : constructor seharusnya menggunakan awalan huruf kapital
 Nomor 3 : Constructor dan nama kelas harus sama dan menggunakan huruf awalan kapital
 Method "sukaNonton" harus didefinisikan dengan parameter yang sesuai.
 Nomor 4. Ada beberapa hal yang perlu di evaluasi
 - Override yang Tidak Diperlukan:** Method yang sama persis di kelas Anak dan Ortu harus dihapus untuk efisiensi.
 - Method Overloading:** Tetap efisien dan memberikan fleksibilitas tanpa konflik dengan method di kelas induk.
 - Override yang Dibutuhkan:** Diperlukan ketika perilaku berbeda diinginkan, seperti pada method sukaMembaca.
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

2) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Organisasi Kode: Kelas membantu mengorganisir kode menjadi unit yang lebih kecil dan lebih terkelola, membuatnya lebih mudah untuk dipahami dan dirawat.

Reusabilitas: Metode dalam kelas dapat digunakan kembali di berbagai bagian program, mengurangi duplikasi kode dan meningkatkan efisiensi.

b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)

3) Kreasi

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

Lebih bisa memahami materi mengenai class, method, objek, dan extends

b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Contoh Jawaban:

Nama & NPM	Topik:	Tanggal:
Risti Safira G1F024029	Class, method, object, dan extends	18 September 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Contoh:

Tuliskan kembali soal:

```
1. 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");
    } }
```

Luaran 1:

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. Cermati contoh 1. susun kode menggunakan constructor dengan parameter data pribadi anda!

```
2. 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 nanti Anda akan memiliki keturunan, analisa sifat (atribut) dan constructor sebagai Ortu apa yang akan diturunkan (gunakan data karakter pribadi anda) ?

2.3. Rancanglah kode program untuk sifat (atribut) dan constructor overloaded dari Latihan 2.2!

```
3. 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. Berdasarkan Latihan 2.2. Anda sudah punya kode program untuk atribut dan constructor sebagai Ortu.

Kembangkanlah kode program untuk method dari Ortu dengan data perilaku pribadi Anda yang menggunakan:

- a) method overloading,
- b) method dengan return value
- c) method tanpa return value

```
4. 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. Evaluasi method yang dimiliki Contoh 4 pada `class Anak extends Ortu` dengan method di `class Ortu`.

Simpulkan hasil evaluasi Anda agar method ini menjadi efisien!

4.2. Setelah dirunning di JDoodle, catat waktu eksekusinya.

Susun kembali kode program yang dapat mengefisienkan waktu eksekusi!

Pada soal masih ada pesan kesalahan

Untuk nomor 1 kesalahan nya ada di dalam kelas manusia1 yang memasukkan nama dan warna rambut ke dalam deklarasi yang hanya menerima satu parameter

Untuk nomor 2 kesalahan nya adalah nama konstruktor yang tidak sama dengan nama kelas dan juga constructor tidak menerima dua data tetapi di dalam method dituliskan untuk dua data

Nomor 3, Latihan 3.1 nama konstruktor berbeda dengan nama kelas, memanggil method `sukaNonton("Drakor")`, namun method tersebut tidak ada

Nomor 4. Ada beberapa hal yang perlu di evaluasi dan mencatat waktu eksekusi

Atau

Diketahui dari soal : variabel _____

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara _____
- 2) Alasan solusi ini karena _____
- 3) Perbaikan kode program dengan cara _____
- 4) Untuk nomor 1 solusinya adalah membuat constructor yang dapat menerima dua parameter
Untuk Solusi nomor 2.1 adalah nama constructor seharusnya menggunakan awalan huruf kapital, dan juga membuat constructor yang menerima dua data pada 2.2 atribut yang akan di turunkan adalah nama, warna rambut, kepribadian, dan kebiasaan

Nomor 3 : Constructor dan nama kelas harus sama dan menggunakan huruf awalan kapital

Method “sukaNonton” harus didefinisikan dengan parameter yang sesuai.

Nomor 4 : Latihan 4.1 Kelas Anak mendefinisikan ulang method sukaMenonton(String a) dan sukaMembaca(String a), padahal method tersebut sudah ada di kelas Ortu.

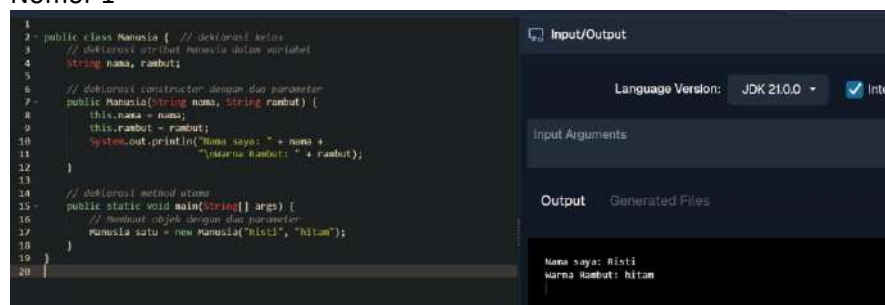
Seharusnya jangan di tulis ulang karena tidak ada perubahan pada kelas anak (Kesimpulan nya agar lebih efisien, method nya tidak perlu di tulis ulang)

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.:
 - a. Mencari permasalahan dalam soal
 - b. Memperbaiki kode yang salah
 - c. Merancang kode atribut
 - d. Membuat kode program sesuai perintah di soal
- 2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Nomor 1



```
1
2 public class Manusia { // deklarasi kelas
3     // deklarasi atribut manusia dalam variabel
4     String nama, rambut;
5
6     // deklarasi constructor dengan dua parameter
7     public Manusia(String nama, String rambut) {
8         this.nama = nama;
9         this.ambut = rambut;
10        System.out.println("Nama saya: " + nama +
11                           "\nWarna Rambut: " + rambut);
12    }
13
14    // deklarasi method utama
15    public static void main(String[] args) {
16        // Membuat objek dengan dua parameter
17        Manusia satu = new Manusia("Risti", "hitam");
18    }
19 }
20
```

Input/Output

Language Version: JDK 21.0.0 ☒ Intere

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam

Nomor 2
Latihan 2.1

```

1 public class Ortu {
2     String nama, rambut;
3
4     // Deklarasi constructor yang benar
5     public Ortu(String nama, String rambut) {
6         this.nama = nama; // inisialisasi variabel nama
7         this.ambut = rambut; // inisialisasi variabel rambut
8         System.out.println("Nama saya: " + nama +
9             "\nWarna Rambut: " + rambut);
10    }
11
12    public static void main(String[] args) {
13        // Membuat objek dengan dua parameter
14        Ortu satu = new Ortu("Risti", "hitam");
15    }
16 }

```

Input/Output

Language Version:

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam

Latihan 2.3

```

1 public class Ortu {
2     String nama, rambut, sifat, hobi;
3
4     // Constructor 1: menerima nama dan rambut
5     public Ortu(String nama, String rambut) {
6         this.nama = nama;
7         this.ambut = rambut;
8         System.out.println("Nama saya: " + nama +
9             "\nWarna Rambut: " + rambut);
10    }
11
12    // Constructor 2: menerima nama, rambut, dan sifat
13    public Ortu(String nama, String rambut, String sifat) {
14        this.nama = nama;
15        this.ambut = rambut;
16        this.sifat = sifat;
17        System.out.println("Nama saya: " + nama +
18            "\nWarna Rambut: " + rambut +
19            "\nsifat: " + sifat);
20    }
21
22    // Constructor 3: menerima nama, rambut, sifat, dan hobi
23    public Ortu(String nama, String rambut, String sifat, String hobi) {
24        this.nama = nama;
25        this.ambut = rambut;
26        this.sifat = sifat;
27        this.hobi = hobi;
28        System.out.println("Nama saya: " + nama +
29            "\nWarna Rambut: " + rambut +
30            "\nsifat: " + sifat +
31            "\nhobi: " + hobi);
32    }
33 }

```

Input/Output

Language Version: JDK 21.0.0 ☒ Int

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam
Nama saya: Rista
Warna Rambut: hitam
Sifat: rajin
Nama saya: Siti
Warna Rambut: hitam
Sifat: rajin
hobi: membaca

Compiled and executed in 1.501 sec(s)

```

33
34 public static void main(String[] args) {
35     // Memanggil constructor 1
36     Ortu ortu1 = new Ortu("Risti", "hitam");
37
38     // Memanggil constructor 2
39     Ortu ortu2 = new Ortu("Rista", "hitam", "rajin");
40
41     // Memanggil constructor 3
42     Ortu ortu3 = new Ortu("Siti", "hitam", "rajin", "membaca");
43 }
44 }
45

```

Nomor 3

Latihan 3.1

```

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         this.nama = nama;
8         this.ambut = rambut;
9         System.out.println("Nama saya: " + nama +
10             "\nWarna Rambut: " + rambut);
11    }
12
13    // method untuk menampilkan hobi menonton
14    void sukaMonton(String film) {
15        System.out.println("Hobi Menonton: " + film);
16    }
17
18    // method untuk menghitung jam nonton (overloading)
19    int sukaMonton(int episode, int durasi) {
20        return episode * durasi; // menghitung total jam menonton
21    }
22
23    // method utama
24    public static void main(String[] args) {
25        Manusia satu = new Manusia("Risti", "hitam");
26        satu.sukaMonton("Drakor"); // memanggil method dengan string
27        int jumlahJam = satu.sukaMonton(1, 2); // memanggil method dengan 2 parameter
28        System.out.println("Jam nonton = " + jumlahJam + " jam");
29    }
30 }
31

```

Input/Output

Language Version: JDK 21.0

Input Arguments

Output Generated Files

Nama saya: Risti
Warna Rambut: hitam
Hobi Menonton: Drakor
Jam nonton = 2 jam

Compiled and executed in 1.887 sec(s)

Latihan 3.2

```

1 public class Ortu {
2     String nama, rambut, sifat, hobi;
3
4     // constructor yang menerima semua atribut
5     public Ortu(String nama, String rambut, String sifat, String hobi) {
6         this.nama = nama;
7         this.rambut = rambut;
8         this.sifat = sifat;
9         this.hobi = hobi;
10        System.out.println("Nama saya: " + nama +
11        "Warna Rambut: " + rambut +
12        "Sifat: " + sifat +
13        "Hobi: " + hobi);
14    }
15
16    // Method tanpa return value: Menampilkan aktivitas
17    void aktivitas(String kegiatan) {
18        System.out.println(nama + " sedang melakukan aktivitas: " + kegiatan);
19    }
20
21    // Method dengan return value: Menghitung waktu yang dibutuhkan
22    int waktuAktivitas(int durasi, int frekuensi) {
23        return durasi * frekuensi; // menghitung total waktu aktivitas
24    }
25
26    // Method overloading: versi yang menerima deskripsi tambahan
27    void aktivitas(String kegiatan, String tempat) {
28        System.out.println(nama + " sedang melakukan aktivitas: " + kegiatan + " di " + tempat);
29    }
30
31    public static void main(String[] args) {
32        // Membuat objek Ortu dengan data pribadi
33        Ortu ortu = new Ortu("Risti", "hitam", "rajin", "membaca");
34
35        // Memanggil method tanpa return value
36        ortu.aktivitas("membaca buku");
37
38        // Memanggil method dengan return value
39        int totalWaktu = ortu.waktuAktivitas(1, 3); // 3 jam per hari, 3 kali di
40        System.out.println("Total waktu aktivitas: " + totalWaktu + " jam");
41
42        // Memanggil method overloading
43        ortu.aktivitas("membaca buku", "perpustakaan");
44    }
45 }

```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```

Nama saya: Risti
Warna Rambut: hitam
Sifat: rajin
Hobi: membaca
Risti sedang melakukan aktivitas: membaca buku
Total waktu aktivitas: 3 jam
Risti sedang melakukan aktivitas: membaca buku di perpustakaan

```

Compiled and executed in 1.884 sec(s)

```

31 public static void main(String[] args) {
32     // Membuat objek Ortu dengan data pribadi
33     Ortu ortu = new Ortu("Risti", "hitam", "rajin", "membaca");
34
35     // Memanggil method tanpa return value
36     ortu.aktivitas("membaca buku");
37
38     // Memanggil method dengan return value
39     int totalWaktu = ortu.waktuAktivitas(1, 3); // 3 jam per hari, 3 kali di
40     System.out.println("Total waktu aktivitas: " + totalWaktu + " jam");
41
42     // Memanggil method overloading
43     ortu.aktivitas("membaca buku", "perpustakaan");
44 }
45 }

```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```

Nama saya: Risti
Warna Rambut: hitam
Sifat: rajin
Hobi: membaca
Risti sedang melakukan aktivitas: membaca buku
Total waktu aktivitas: 3 jam
Risti sedang melakukan aktivitas: membaca buku di perpustakaan

```

Compiled and executed in 1.584 sec(s)

Nomor 4

```

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        long startTime = System.currentTimeMillis(); // Catat waktu sebelum eksekusi
12
13        System.out.println("Sifat Orang Tua :");
14        Ortu ortu = new Ortu();
15        ortu.sukaMenonton("youtube");
16        ortu.sukaMembaca("koran");
17
18        System.out.println("Sifat Anak :");
19        Anak anak = new Anak();
20        anak.sukaMenonton("Film Drakor");
21        anak.sukaMembaca("novel");
22
23        long endTime = System.currentTimeMillis(); // Catat waktu setelah eksekusi
24        System.out.println("Waktu eksekusi: " + (endTime - startTime) + " ms");
25    }
26 }
27
28 class Anak extends Ortu {
29     void sukaMenonton(int a, String b) {
30         System.out.println("Nonton Jam " + a + " Malam " + b);
31     }
32 }
33 }

```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```

Sifat Orang Tua :
Nonton youtube
Suka Baca koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca novel
Waktu eksekusi: 60 ms

```

Compiled and executed in 1.906 sec(s)

Beri komentar pada kode yang di Screenshot

Nomor 1:

Constructor Ortu(String, String): Constructor ini menerima dua argumen yaitu nama dan rambut. Kedua atribut tersebut diinisialisasi dan dicetak ke output.

Method main: Membuat objek Ortu dengan parameter "Risti" dan "hitam". Ini di gunakan untuk memanggil constructor dan menampilkan informasi ke layar.

Nomor 2.2:

Constructor Overloading: Terdapat tiga constructor di kelas Ortu, masing-masing menerima jumlah parameter yang berbeda:

Constructor 1: Hanya menerima nama dan rambut.

Constructor 2: Menambahkan parameter kepribadian dan memanggil constructor pertama.

Constructor 3: Menambahkan parameter hobi dan memanggil constructor kedua.

Method main: Membuat objek Ortu dengan berbagai kombinasi parameter dan menampilkan atribut mereka.

Nomor 3.2:

Method Overloading: Ada dua method sukaNonton. Satu menerima parameter String untuk menampilkan film yang ditonton, dan satu lagi menerima dua int untuk menghitung total waktu nonton berdasarkan jumlah episode dan durasi.

Return Value: Method sukaNonton(int, int) mengembalikan hasil perkalian antara jumlah episode dan durasi per episode.

Nomor 4:

pewarisan: Kelas Anak mewarisi method dari kelas Ortu.

Method Overriding: Method sukaMembaca di kelas Anak menimpa method sukaMembaca di kelas Ortu, memungkinkan kelas Anak memiliki perilaku berbeda untuk method tersebut.

Method Overloading: Method sukaMenonton di kelas Anak di-overload dengan menambahkan versi yang menerima dua parameter (int dan String), sementara versi lain tetap memanggil method dari induknya.

Untuk menghitung waktu eksekusi

Metode System.currentTimeMillis() adalah metode statis di kelas System. Ini mengembalikan waktu saat ini dalam milidetik sebagai nilai panjang. Metode ini biasanya digunakan untuk mengukur interval waktu, stempel waktu, dan eksekusi kode profil.

b) Analisa luaran yang dihasilkan

Contoh:

Luaran sudah sesuai dengan program yang disusun.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

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

Ya untuk kode dan luaran sudah sesuai

[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!

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

Contoh jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena _agar semua orang bisa mengakses kelas saya untuk melihat program yang dikerjakan_____

Perbaiki program dengan menambahkan _____ karena struktur java mengharuskan _____

(penjelasan analisa mengulangi kembali materi yang sudah diberikan)

(penjelasan mengkaitkan dengan materi yang ada)

2) Evaluasi

a) Apa konsekuensi/dampak dari kode program yang dibuat?

Jika ada kode yang salah maka akan eror

b) Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)

Contoh jawaban Evaluasi:

Pada program itu saya mengkonversi bentuk kelas public karena _____

Setelah dikonversi, saya mengevaluasi bahwa tipe data _____ lebih baik digunakan untuk bentuk data seperti _____

(penjelasan evaluasi mengulangi kembali materi yang sudah diberikan dan mengetahui kekurangan dari materi hasil eksperimen

(misal tipe data ____ ternyata tidak dapat dipakai untuk _____ karena _____)

3) Kreasi

- a) Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?
- b) Susunlah hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Contoh jawaban Kreasi:

Pada program itu saya mengkonversi bentuk kelas public karena _____

Setelah dikonversi, saya mengevaluasi bahwa tipe data _____ lebih baik digunakan untuk bentuk data seperti _____

Saya telah mencoba mengubah menjadi kelas private dan protected, ternyata menghasilkan _____

Berarti kelas private dan protected mempengaruhi _____

(sampaikan temuan Anda yang baru diketahui, eksperimen baru diluar materi yang diberikan)

(penjelasan kreasi mengulangi kembali materi yang sudah diberikan dan menambahkan pengetahuan baru dari pengalaman dari hasil eksperimen)

Lanjutkan ke soal nomor 2 – 3 – ... – dan seterusnya

Refleksi

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)