Nama & NPM	Topik:	Tanggal:
Shalfa Maharani Fikrian	Kelas Java	19 September 2024
G1F024016		

## [No. 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

#### Contoh 1:

#### 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. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi
  - a. atribut variabel, dan
  - b. perilaku/ behavior!

## [No.1] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara nama kelas pada kode di ubah dan di samakan menjadi Manusia1
- 2) Pada method tipe data String menambahkan string warna kulit dan hobi
- 3) Pada kode public Manusia1 (String nama) menambahkan di tipe data string rambut,kulit,hobi
- 4) Pada kode System.out.println(" Nama saya : "+ nama + "\n Warna Rambut : " + rambut); menambahkan untuk warna kulit dan hobi
- 5) Pada kode Manusia1 satu = new Manusia1("Putri", "hitam"); menambahkan warna kulit dan hobi

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

- 1) Algoritma
  - a) Salin kode ke jdoodle/eclipse/progamiz
  - b) Samakan nama method manusia menjadi Manusia1
  - c) Menambahkan warna kulit dan hobi
  - d) Menambahkan kode untuk menghasilkan warna kulit dan hobi
- 2) Kode program dan luaran

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



b) Analisa luaran yang dihasilkanLuaran sudah sesuai dengan program yang disusun.

## [No.1] Kesimpulan

#### **Analisa**

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Jawaban :

Berdasarkan kode progam yang telah saya susun output yang di hasilkan sudah sesuai dengan kode dan pengubahan nama method manusia menjadi manusia1 untuk menyamakan agar program tidak error.

## [No. 2] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

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

### 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 Ortu memiliki data variabel umur = 25 dan jenis kelamin = P (untuk Perempuan), rekomendasikan constructor dengan parameter yang baru untuk ditambahkan dalam program!

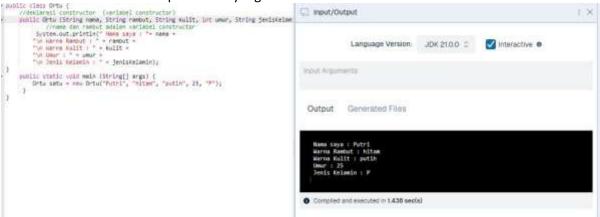
## [No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan untuk menambahkan tipe data string pada kode public Ortu
- 2) Menambahkan umur, jenis kelamin dan warna kulit

- 4) Pada kode Ortu satu = new Ortu("Putri", "hitam"); tambahkan juga untuk umur, jenis kelamin dan warna kulit ortu

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

- 1) Algoritma
  - a) Salin kode ke jdoodle/eclipse/progamiz
  - b) Tambahkan umur, jenis kelamin, dan warna kullt pada tipe data string
  - c) Pada system.out.println tambahkan kode untuk umur,jenis kelamin, dan warna kullt
  - d) Pada kode Ortu satu tambahkan umur, jenis kelamin dan warna kulit ortu
- 2) Kode program dan luaran
  - a) Screenshot/ Capture potongan kode dan hasil luaran Beri komentar pada kode yang di Screenshot



b) Analisa luaran yang dihasilkanLuaran sudah sesuai dengan program yang disusun.

#### [No.2] Kesimpulan

## Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Jawaban:

Kode program yang disusun telah sesuai dengan program yang diminta soal pengambilan keputusan untuk solusi pada program diambil dari materi yang telah diajarkan pada praktikum

### [No. 3] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

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

```
//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. Ubahlah method dan constructor Contoh 3 sesuai dengan perilaku/ behavior anda
- 3.3. Berdasarkan Contoh 3 dan Latihan 3.2. simpulkan perbedaan:
  - a) constructor overloading dan overriding
  - b) method overloading, dan method overriding
  - c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

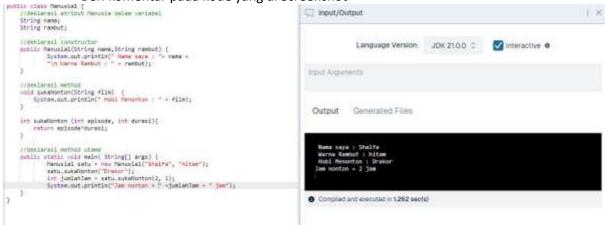
## [No.3] Analisis dan Argumentasi

- 1) Pada kode void sukaNonton { saya menambahkan tipe data String + nama variabel
- 2) Pada kode int sukaNonton { saya menambahkan tipe data int + nama variabel
- 3) Pada kode episode\*durasi saya menambahkan return di depan nya agar program bisa berjalan
- 4) Pada nama method saya ubah dari Manusia menjadi Manusia1
- 5) Perbedaan constructor overloading dan overriding yaitu constructor overloading terjadi ketika sebuah kelas memiliki lebih dari satu constructor dengan nama yang sama tetapi dengan parameter yang berbeda sedangkan constructor verriding Setiap kelas harus memiliki constructor-nya sendiri, dan meskipun subclass dapat memiliki constructor dengan nama yang sama, itu bukanlah overriding tetapi lebih kepada pembuatan constructor baru.
- 6) Perbedaaan method overloading dan method overriding, method overloading adalah konsep di mana beberapa method dalam satu kelas memiliki nama yang sama tetapi parameter (jumlah atau tipe) yang berbeda sedangkan Method overriding terjadi ketika subclass mendefinisikan ulang method dari superclass dengan nama, parameter, dan tipe pengembalian yang sama tetapi memberikan implementasi yang berbeda. Ini memungkinkan subclass untuk memberikan perilaku spesifik dari method yang diwarisi dari superclass.
- 7) Perbedaan method yang mengembalikan nilai dan method yang tidak mengembalikan nilai yaitu method yang mengembalikan nilai memiliki tipe pengembalian tertentu sedangkan Method yang tidak mengembalikan nilai dideklarasikan dengan tipe

pengembalian void. Method ini menjalankan aksi tertentu tetapi tidak menghasilkan nilai kembali.

# [No.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - a) Salin kode program
  - b) Ubah nama method dari Manusia menjadi Manusia1 agar sam semua dan program tidak error
  - c) Tambahkan tipe data String + nma variabel pada kode void sukaNonton {
  - d) Tambahkan tipe data int + nama variabel pada kode int sukaNonton {
  - e) Tambahkan return di depan kode episode\*durasi
- 2) Kode program dan luaran
  - a) Screenshot/ Capture potongan kode dan hasil luaran Beri komentar pada kode yang di Screenshot



b) Analisa luaran yang dihasilkan Luaran sudah sesuai dengan program yang disusun dan permintaan soal

#### [No.3] Kesimpulan

#### **Analisa**

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Jawaban:

Untuk kode program sudah disusu sesuai dengan permintaan soal dan kode program telah diperbaiki. Kesalahan terjadi pada nama method, kurang nya tipe data dan nama variabel pada kode,.

## [No. 4] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

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

```
Ortu objekO = new Ortu(); // memanggil objek induk
   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);
 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
   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
Latihan 4:
4.1. Evaluasi method yang dimiliki class Anak extends Ortu dengan method
di class Ortu!
   Apakah penulisan method ini sudah efisien?
```

4.2. Setelah dirunning di JDoodle, catat waktu eksekusinya.

Rekomendasikan perbaikan penulisan kode method untuk dapat mengefisienkan waktu eksekusi!

#### [No.4] Analisis dan Argumentasi

- 1) Penulisan pada method sudah efisien karena yang ditampilkan data anak dan ortu sesuai .
- 2) Pada saaat running di jdoodle waktu eksekusinya selama 3.585 sec(s)
- 3) Pada method class Anak extends Ortu memiliki isi sifat anak
- 4) Pada method class Ortu memiliki isi sifat ortu
- 5) Menghapus kode

untuk mempersingkat kode program karena pada kode diatas sudah dijabarkan sifat anak dan ortu sehingga tidak perlu menambahkan kode tersebut.

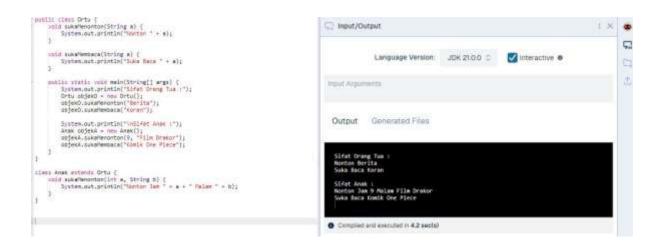
## [No.4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
  - a) Salin kode program ke jdoodle
  - b) Run untuk melihat berapa lama waktu eksekusinya

```
c) Menghapus kode System.out.println("Nonton Jam " + a + " Malam " + b);
     }
     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
      objekO.sukaMenonton("Berita"); // memanggil sifat spesifik
      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
   karena sifat anak dan ortu sudah dijabarkan pada kode sebelumnya.
```

# 2) Kode program dan luaran

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



b) Analisa luaran yang dihasilkan Luaran sudah sesuai dengan program yang disusun.

# [No.4] Kesimpulan

## **Analisa**

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Jawaban:

Kode progam outputnya sudah sesuai dengan soal dan kode program telah dipersingkat agar tidak bingung saat melihat program