Nama & NPM	Topik:	Tanggal:
Khairul Aji Pangestu G1F024042	**Kelas Java**	**15 September 2024**

[Nomor Soal] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel
2) Rincikan sumber informasi yang relevan (buku / webpage)
3) Uraikan rancangan solusi yang diusulkan (jika ada).
4) Analisis susunan solusi, parameter solusi (jika ada).

[Nomor Soal] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.
2) Analisis solusi, kaitkan dengan permasalahan.

[Nomor Soal] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma
2) Tuliskan kode program dan luaran
    a) Beri komentar pada kode
    b) Uraikan luaran yang dihasilkan
    c) Screenshot/ Capture potongan kode dan hasil luaran

[Nomor Soal] Kesimpulan

1) Analisa
    a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
    b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
2) Evaluasi
    a) Apa konsekuensi dari skenario pemprograman ini?
    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?
    b) Konstruksikan hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Nama & NPM	Topik:	Tanggal:
Khairul Aji Pangestu G1F024042	**Kelas Java**	**15 September 2024**

[No. 1] Identifikasi Masalah:
Unit 1 Kelas (Class)
Contoh 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. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi
   a. atribut variabel, dan
   b. perilaku/ behavior!

1) Uraikan permasalahan dan variabel

Kode Program dibuat untuk untuk membuat Kelas. Kesalahan kode program tersebut karena kurangnya beberapa *syntax*.

[No.1] Analisis dan Argumentasi

1) Kesalahan pada kode program tersebut dapat diatasi dengan cara mengganti nama konstruktor yang cocok dengan nama kelas yaitu Manusia, selanjutnya menambahkan parameter String rambut pada konstruktor, dan inisialisasi atribut rambut.
2) Ciri-ciri lain Kelas Manusia yang dapat menjadi
   a. atribut variable menggambarkan karakteristik yang dimiliki oleh objek tersebut. Misalnya dalam Kelas Manusia yaitu umur, jenis kelamin dan tinggi badan
   b. perilaku/ behavior menggambarkan apa yang dilakukan oleh objek tersebut. Misalnya dalam Kelas Manusia yaitu Membaca, Makan, Tidur, dan Berjalan.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma
   (a) Mulai
   (b) Install aplikasi java compiler
   (c) Buat project Manusia
   (d) Deklarasi class Manusia
   (e) Deklarasi variabel
   (f) Deklarasi parameter
   (g) Print program
   (h) Run
   (i) Selesai

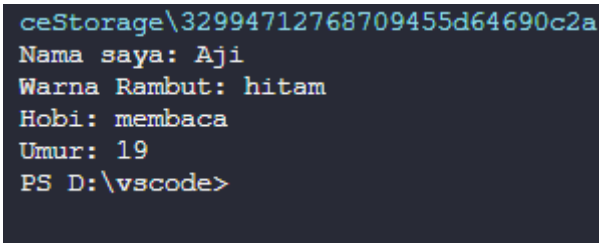
2) Kode program dan luaran
   a) Kode Program

```
java > E-Leanrning > Manusia.java > Manusia > main(String[])
Codeium: Refactor | Explain
public class Manusia {
    // Deklarasi atribut Manusia dalam variabel
    String nama, rambut, hobi;
    int umur;

    // Deklarasi constructor
    public Manusia(String nama, String rambut, String hobi, int umur) {
        this.nama = nama;          // Menginisialisasi atribut nama
        this.rambut = rambut;      // Menginisialisasi atribut rambut
        this.hobi = hobi;          // Menginisialisasi atribut hobi
        this.umur = umur;          // Menginisialisasi atribut umur
        System.out.println("Nama saya: " + nama + "\nWarna Rambut: " + rambut + "\nHobi: " + hobi + "\nUmur: " + umur); ;
    }

    // Deklarasi method utama
    Run | Debug | Codeium: Refactor | Explain | ×
    public static void main(String[] args) {
        // Membuat objek dari kelas Manusia dan menginisialisasi dengan nilai
        Manusia satu = new Manusia(nama:"Aji", rambut:"hitam", hobi:"membaca",  umur:19);
    }
}
```

   b) Hasil Luaran



```
ceStorage\32994712768709455d64690c2a
Nama saya: Aji
Warna Rambut: hitam
Hobi: membaca
Umur: 19
PS D:\vscode>
```

   a) Analisa luaran yang dihasilkan
   Luaran sudah sesuai dengan program yang disusun.
   Kode program dibuat untuk membuat Kelas Manusia dan mendefinisikan atirbut serta behaviornya. Kode sudah benar, dan luaran juga sudah benar.

[No.1] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
   Kesimpulan kode tersebut, kode program dibuat untuk membuat class Manusia dan mendeklarasikan atribut dan behaviornya serta objek dari class tersebut. Keaslahan dapat diatasi dengan menambahkan dan mengganti beberapa syntax.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Alasan mengganti nama konstruktor, menambah atribut rambut, serta inisialisasi atribut rambut dilalukan untuk memperbaiki kesalahan kode program.

[No. 2] Identifikasi Masalah:
Unit 2 Objek
Contoh 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 Ortu memiliki data variabel umur = 25 dan jenis kelamin = P (untuk Perempuan), rekomendasikan constructor dengan parameter yang baru untuk ditambahkan dalam program!

1) Uraikan permasalahan dan variabel

Kode Program tersebut dibuat untuk membuat Kelas Ortu. Kesalahan kode program tersebut karena kurangnya beberapa syntax.

[No.2] Analisis dan Argumentasi

1) Kesalahan pada kode program tersebut dapat diatasi dengan cara mengganti nama konstruktor yang cocok dengan nama kelas, mendeklarasi variabel String nama dan rambut dalam Kelas Ortu, selanjutnya menambahkan parameter String rambut pada konstruktor, dan inisialisasi atribut nama dan rambut.
2) Saya merekomendasikan pada kode progam tersebut menambahkan variabel int umur dan char JenisKelamin, memperbarui konstruktor agar menerima parameter umur dan JenisKelamin, dan mencetak data baru di dalam konstruktor.

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma
   (a) Mulai
   (b) Install aplikasi java compiler
   (c) Buat project Ortu
   (d) Deklarasi class Ortu
   (e) Deklarasi variabel
   (f) Deklarasi parameter
   (g) Print program
   (h) Run
   (i) Selesai

2) Kode program dan luaran

c) Kode Program

```
java > E-Leanrning > Ortu.java > ...
/************  ✨ Codeium Command 🌟  ************/
// Deklarasi kelas Ortu
Codeium: Refactor | Explain
public class Ortu {
    String nama, rambut;     // Deklarasi variabel nama dan rambut
    int umur;                // Deklarasi variabel umur
    char JenisKelamin;        // Deklarasi variabel jenis kelamin

    //deklarasi constructor  (variabel constructor)
    // Constructor untuk inisialisasi nilai
    public Ortu (String nama, String rambut, int umur, char JenisKelamin) {
            this.nama=nama;
            this.rambut=rambut;
            this.umur=umur;
            this.JenisKelamin=JenisKelamin;
        System.out.println(" Nama saya : "+ nama +
                           "\n Warna Rambut : " + rambut+
                           "\n Umur : " + umur +
                           "\n Jenis Kelamin : " + (JenisKelamin == 'P' ? "Perempuan" : "Laki-laki")
        );
    }
    // Method utama
    Run | Debug | Codeium: Refactor | Explain |
    public static void main (String[] args) {
        // Membuat objek Ortu
        Ortu satu = new Ortu(nama:"Sri", rambut:"hitam", umur:44, JenisKelamin:'P');
     }
}
```

d) Hasil Luaran

```
\redhat.java\jdt_ws\vscode_71559de3\bin'
 Nama saya : Sri
 Warna Rambut : hitam
 Umur : 44
 Jenis Kelamin : Perempuan
PS D:\vscode>
```

b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.

Kode program dibuat untuk membuat Kelas Ortu dan mendefinisikan atirbut serta behaviornya. Kode sudah benar, dan luaran juga sudah benar.

[No.2] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Kesimpulannya kode program tersebut dibuat untuk membuat class Ortu dengan atribut dan behaviornya, dan juga membuat objeknya. Kesalahan dapat diatasi dengan mengganti nama konstruktor, menambah parameter String rambut pada kontruktor, dan inisialisasi atribut rambut dan nama.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Alasan saya menambah dan mengganti beberapa kode program untuk memperbaiki kesalahan pada kode. Alasan saya merekomendasikan menambah atribut umur dan JenisKelamin dan memperbarui konstruktor agar menerima atribut tersebut.

[No. 3Identifikasi Masalah:

Unit 3 Method

Contoh 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. 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 dengan method, constructor dan atribut serta behaviornya. Masih ada kesalahan dalam penulisan Kode Program.

1,794 detik to 1

1) Uraikan permasalahan dan variabel

Kode program tersebut dibuat untuk membuat Kelas Manusia

[No.3] Analisis dan Argumentasi

1) Kesalahan pada kode program tersebut dapat diatasi dengan cara mengganti nama konstruktor yang cocok dengan nama kelas yaitu Manusia, selanjutnya menambahkan parameter String film dalam sukaMenonton, dan menambahkan syntax return untuk perhitungan episode*durasi

2) Pada latihan 3.2 perbedannya yaitu:
   a) Constructor overloading dan overriding
      - Constructor Overloading: Terjadi ketika sebuah kelas memiliki beberapa constructor dengan parameter yang berbeda (baik jumlah atau tipe parameternya). Ini memungkinkan kita untuk membuat objek dengan cara yang berbeda tergantung pada argument yang diterima oleh constructor.
      - Constructor Overriding: Tidak mungkin dilakukan di Java, karena constructor tidak diwariskan dari kelas induk ke kelas turunan, sehingga tidak bisa di-override.
   b) Method Overloading dan Method Overriding
      - Method Overloading: Terjadi ketika sebuah kelas memiliki beberapa metode dengan nama yang sama, tetapi dengan parameter yang berbeda (baik jumlah, tipe, atau keduanya). Ini memberikan fleksibilitas dalam cara metode tersebut dipanggil berdasarkan argument yang diberikan.
      - Method Overriding: Terjadi ketika sebuah metode dalam subclass memiliki nama, return type, dan parameter yang sama persis dengan metode di superclass. Metode di subclass menggantikan metode di superclass.
   c) Nilai yang mengembalikan nilai dan Method yang tidak memngembalikan nilai
      - Method yang Mengembalikan Nilai: Adalah metode yang memiliki return type selain void dan mengembalikan suatu nilai ketika dipanggil. Nilai yang dikembalikan dapat berupa tipe data apa pun (int, String, objek, dll).
      - Method yang Tidak Mengembalikan Nilai: Adalah metode dengan return type void, yang berarti tidak mengembalikan nilai apa pun setelah dijalankan.

[No.3] Penyusunan Algoritma dan Kode Program

3) Algoritma
   (a) Mulai
   (b) Install aplikasi java compiler
   (c) Buat project Manusia2
   (d) Deklarasi class Manusia2
   (e) Deklarasi variabel
   (f) Deklarasi parameter
   (g) Print program
   (h) Run
   (i) Selesai

4) Kode program dan luaran
   a) Kode Program

```java
java > E-Leanrning > Manusia2.java > ...
Codeium: Refactor | Explain
public class Manusia2 {
  //deklarasi atribut Manusia dalam variabel
  String nama, npm;
  int umur;

  //deklarasi constructor
  public Manusia2(String nama, String npm, int umur) {
        System.out.println(" Nama saya : "+ nama +
        "\n Warna NPM : " + npm + "\n Umur : " + umur);
  }

  //deklarasi method
  Codeium: Refactor | Explain |
  void sukaBaca (String bacaan) {
      System.out.println(" Hobi Membaca : " + bacaan);
  }

  Codeium: Refactor | Explain | Generate Javadoc |
  int sukaBaca (int chapter, int durasi) {
      return chapter*durasi;
  }

  //deklarasi method utama
  Run | Debug | Codeium: Refactor | Explain |
  public static void main( String[] args) {
        Manusia2 satu = new Manusia2 (nama:"Khairul", npm:"G1F024024", umur:19);
        satu.sukaBaca(bacaan:"Komik");
        int jumlahmenit = satu.sukaBaca(chapter:20, durasi:5);
        System.out.println("Durasi Membaca = " +jumlahmenit + " menit");
  }
}
```

b) Hasil Luaran

```
 Nama saya : Khairul
 Warna NPM : G1F024024
 Umur : 19
 Hobi Membaca : Komik
Durasi Membaca = 100 menit
PS D:\vscode>
```

c) Analisa luaran yang dihasilkan
Luaran sudah sesuai dengan program yang disusun.
Kode program dibuat untuk membuat Kelas Manusia1 dan mendefinisikan atirbut serta behaviornya. Kode sudah benar, dan luaran juga sudah benar.

[No.2] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Kesimpulannya kode program tersebut untuk membuat class Manusi dengan atribut, behavior serta methodnya. Kesalahan dapat diatasi dengan mengganti nama konstruktor, menambahkan parameter String film dalam sukaMenonton, dan menambahkan syntax return. Overridng dan Overloading tidak sama.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Alasan mengganti dan menambah beberapa kode untuk mengatasi kesalahan pada kode program.

[No. 4] Identifikasi Masalah:

Unit 4 Extends

Contoh 1:

```
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
}  }
```

Rekomendasikan perbaikan penulisan kode method untuk dapat mengefisienkan waktu eksekusi!

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!

1) Uraikan permasalahan dan variabel
   Kode Program dibuat untuk untuk membuat Kelas induk dan Kelas anak serta atribut, method, konstruktor, dan objeknya. Kode tersebut belum efisien.

[No.4] Analisis dan Argumentasi

1) Kode yang diberikan sudah benar tetapi belum efisien, cara mengefisienkannya ialah dengan menghapus deklarasi berlebihan pada class Anak, mengganti motode dengan metode Overriding, mendeklarsi main cukup satu kali.
2) Waktu eksekusi sebelum kode diperbaiki adalah 1.76 sec dan ketika sudah di perbaiki adalah 1.24 sec.

[No.4] Penyusunan Algoritma dan Kode Program

1) Algoritma
   (a) Mulai
   (b) Buka Jdoodle(online java compiler)
   (c) Deklarasi class Ortu
   (d) Deklarasi method
   (e) Buat atribut dan behavior
   (f) Deklarasi class Anak extends Ortu
   (g) Buat objek
   (h) Print program
   (i) Run
   (j) Selesai

2) Kode program dan luaran
   a) Kode Program

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

    // Method that can be overridden by the Anak class
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }

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

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

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

    // Overriding the method from Ortu class
    @Override
    void sukaMembaca(String a) {
        System.out.println("Suka Baca " + a);
    }
}
```

b) Hasil Luaran

```
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

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

Compiled and executed in 1.24 sec(s)
```

c) Analisa luaran yang dihasilkan
Luaran sudah sesuai dengan program yang disusun.
Kode program dibuat untuk membuat Kelas Ortu dan Anak menentukan methodnya, dan membuat atribut, behavior, dan juga objeknya. Kode sudah benar, dan luaran juga sudah benar.

[No.4] Kesimpulan

2) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Kesimpulannya Kode program ini dibuat untuk membuat class Ortu dan Anak serta method, atribut, behavior, dan objeknya. Untuk mengefisienkan kode kita perlu mengganti dan menghapus beberapa kode, salah satunya mengganti methodnya.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Alasannya karena Anakkelas menggantikan metode Ortu(sukaMenonton(String a)dan sukaMembaca(String a)), deklarasi metode duplikat tambahan di kelas anak akan menambah redundansi. Menghapus satu main karena program harus memiliki satu mainmetode untuk eksekusi.

Refleksi

Materi Class java cukup dapat dipahami tetapi juga ada kesulitan pada unit Method, overall masih bisa mengikuti materi walaupun sedikit terlambat daripada teman-teman yang lain karena kurangnya manajemen waktu.