

Contoh Jawaban:

| Nama & NPM | Topik: | Tanggal: |
|-----------------|--------------------|-------------------|
| M. Bagas Arjuna | Kelas,Objek,Method | 19 September 2024 |

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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");
    }
}
```

Pada soal masih ada pesan kesalahan yaitu yang terpadat pada constructor yaitu namanya berbeda dengan nama class kode yang salah:

- Public **Manusia1** tetapi nama class manusia tidak pake **(1)**
- Kesalahan selanjutnya tidak ada penambahan perintah String di constructor untuk memanggil rambut.
- Pada method utama juga terdapat kesalahan yaitu hanya pada penamaan Manusia1 yang berbeda dengan nama kelasnya.

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini karena kode tersebut telah di coba atau di jalankan dan mendapatkan hasil output yaitu menampilkan nama dan warna rambut. _____
- 2) Alasan solusi ini karena kode tersebut telah di analisis dan sudah di jalankan
- 3) Perbaikan kode program dengan cara perubahan nama constructor,method utama pada perubahan tersebut di ubah menjadi sesuai dengan nama class yaitu manusia, selanjutnya penambahan tipe data String untuk memanggil variable rambut.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
langkah-langkah penyelesaian masalah.
 1. Start
 2. Membaca kesalahan yang tampil pada output
 3. Melakukan perubahan constructor dan method
 4. Melakukan Penambahan tipe data pada konstruktor
 5. Jalankan kode program
 6. Selesai

- 2) Kode program:

```
public class Manusia { // deklarasi kelas
    //deklarasi atribut Manusia dalam variabel
    String nama, rambut;
```

```

//deklarasi constructor
public Manusia (String nama, String rambut) {
    System.out.println(" Nama saya : "+ nama +
        "\n Warna Rambut : " + rambut);
}

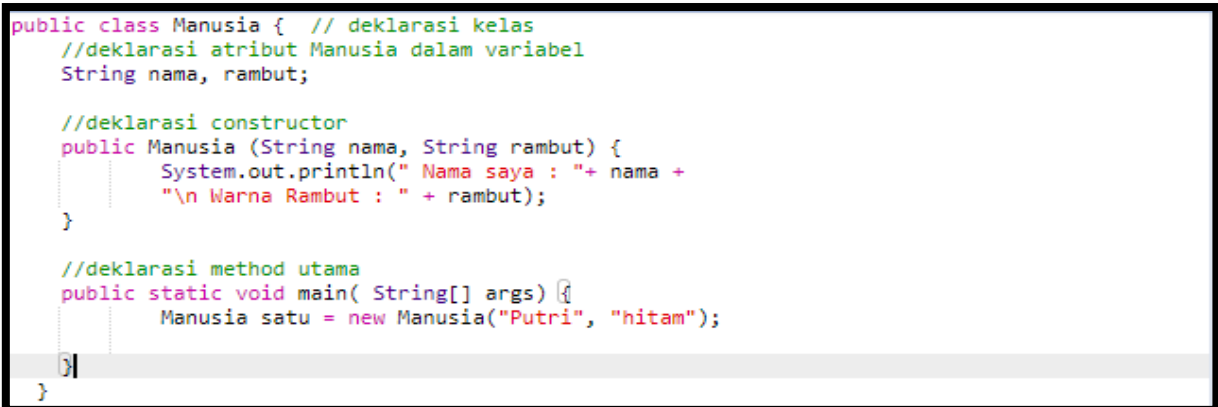
//deklarasi method utama
public static void main( String[] args) {
    Manusia satu = new Manusia("Putri", "hitam");
}
}

```

luaran :

Nama saya : Putri
Warna Rambut : hitam

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



```

public class Manusia { // deklarasi kelas
    //deklarasi atribut Manusia dalam variabel
    String nama, rambut;

    //deklarasi constructor
    public Manusia (String nama, String rambut) {
        System.out.println(" Nama saya : "+ nama +
            "\n Warna Rambut : " + rambut);
    }

    //deklarasi method utama
    public static void main( String[] args) {
        Manusia satu = new Manusia("Putri", "hitam");
    }
}

```

(Gambar 1.1 perbaikan kode class)

Luaran:



```

Nama saya : Putri
Warna Rambut : hitam

```

i CPU Time: 0.04 sec(s) | Memory: 39148 kilobyte(s) | Compiled and executed in 1.362 sec(s)

(Gambar hasil output 1.1 perbaikan kode class)

B) Analisa luaran yang dihasilkan :

Luaran sudah sesuai dengan program yang disusun.

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

Kode program yang di buat bertujuan untuk menyelesaikan masalah yang ada pada kode program yang Dimana kesalahan terdapat pada nama class yang tdaik sama dengan constructor dan method, dan juga melakukan penambahan tipe data untuk memanggil variable.

1.2. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi

- a. atribut variabel, dan
- b. perilaku/ behavior!

Kode Program:

```
public class Manusia { // deklarasi kelas
    // Deklarasi atribut Manusia dalam variabel
    String nama;
    String rambut;
    int umur;
    String hobi;

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

    // Method untuk tidur
    void tidur() {
        System.out.println(nama + " sedang tidur.");
    }

    // Method utama
    public static void main(String[] args) {
        Manusia satu = new Manusia("Putri", "hitam", 18, "Basket");
        satu.tidur(); // Memanggil method tidur
    }
}
```

(Gambar 1.2 atribut dan perilaku)

Luaran:

```
Nama saya : Putri
Warna Rambut : hitam
Umur : 18
Hobi : Basket
Putri sedang tidur.
```

Compiled and executed in 1.376 sec(s)

(Gambar hasil output 1.2 atribut dan perilaku)

[No.1] Kesimpulan

1) Analisa

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

Pada program ini saya menganalisis permasalahan yang terdapat pada kode programan yang telah di berikan, setelah di analisis terdapat permasalahan-permasalahan atau ada kode program yang kurang dan tidak cocok pada permasalahan pertama saya jumpai di penamaan untuk pemanggilan nama class yaitu pada constructor dan method jadi untuk penyelesaian nya hanya melakukan perubahan nama agar nama tersebut sesuai dengan nama class, pada permasalahan ke 2 yaitu di constructor tidak ada pemanggilan variabel rambut hanya ada pemanggilan variable nama dengan itu cara mengatasinya dengan menambahkan tipe data String di dalam constructor. Alasan pengambilan Keputusan ini adalah karena telah di analisis terlebih dahulu dan sudah di praktekan dan ternyata hasil nya berhasil atau kode nya berjalan.

[No. 2] Identifikasi Masalah:

- Uraikan permasalahan dan variabel
Tuliskan kembali soal:

```

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");
    }
}

```

Pada soal masih ada pesan kesalahan yaitu yang terpadat pada constructor yaitu namanya berbeda dengan nama class kode yang salah:

- Public **ortu** tetapi nama class Ortu (O) nya menggunakan huruf kapital
- Kesalahan selanjutnya tidak ada penambahan perintah String di constructor untuk memanggil nama dan rambut.

[No.2] Analisis dan Argumentasi

1. Saya mengusulkan permasalahan ini karena kode tersebut telah di coba atau di jalankan dan mendapatkan hasil output yaitu menampilkan nama dan warna rambut.
2. Alasan solusi ini karena kode tersebut telah di analisis dan sudah di jalankan
3. Perbaikan kode program dengan cara melakukan perubahan nama constructor, pada perubahan tersebut di ubah menjadi sesuai dengan nama class yaitu Ortu dimulai dengan huruf besar atau kapital, selanjutnya penambahan tipe data String untuk memanggil variable nama dan rambut.

[No.2] Penyusunan Algoritma dan Kode Program

- 3) Algoritma
langkah-langkah penyelesaian masalah.
 7. Start
 8. Membaca kesalahan yang tampil pada output
 9. Melakukan perubahan constructor
 10. Melakukan Penambahan tipe data String pada konstruktur
 11. Jalankan kode program
 12. Selesai

- 4) Kode program:

```

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 :

Nama saya : Putri

Warna Rambut : hitam

- C) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

```
public class Ortu {  
    //deklarasi constructor (variabel 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");  
    }  
}
```

(Gambar 2.1 perbaikan kode class)

Luaran:

```
Nama saya : Putri  
Warna Rambut : hitam
```

Compiled and executed in 1.805 sec(s)

(Gambar hasil output 2.1 perbaikan kode class)

D) Analisa luaran yang dihasilkan :

Luaran sudah sesuai dengan program yang disusun.

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

Kode program yang di buat bertujuan untuk menyelesaikan masalah yang ada pada kode program yang Dimana kesalahan terdapat pada nama class yang tidak sama dengan constructor dan method, dan juga melakukan penambahan tipe data untuk memanggil variable.

1.2. Analisa ciri-ciri lain Kelas Manusia yang dapat menjadi

- a. atribut variabel, dan
- b. perilaku/ behavior!

Kode Program:

```
public class Ortu {  
    String nama;  
    String rambut;  
    int umur;  
    String jeniskelamin;  
  
    //deklarasi constructor (variabel constructor)  
    public Ortu(String nama, String rambut, int umur, String jeniskelamin){  
        //nama dan rambut adalah variabel constructor  
        System.out.println(" Nama saya : " + nama +  
            "\n Warna Rambut : " + rambut + "\n umur : " + umur + "\n jeniskelamin : " + jeniskelamin);  
    }  
    public static void main (String[] args) {  
        Ortu satu = new Ortu("Putri", "hitam", 25, "Perempuan");  
    }  
}
```

(Gambar 2.2 penambahan umur dan jenis kelamin)

Luaran:

```
Nama saya : Putri  
Warna Rambut : hitam  
umur : 25  
jenisKelamin : Perempuan
```

 Compiled and executed in 1.299 sec(s)

(Gambar hasil output 2.2 penambahan umur dan jenis kelamin)

[No.2] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

1) Analisa

- c) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- d) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Pada program ini saya menganalisis permasalahan yang terdapat pada kode programan yang telah di berikan, setelah di analisis terdapat permasalahan-permasalahan atau ada kode program yang kurang dan tidak cocok pada permasalahan pertama saya jumpai di penamaan untuk pemanggilan nama class yaitu pada constructor dan method di dalam method terdapat kekurangan tipe data untuk pemanggilan variabel ke output jadi untuk penyelesaian nya hanya melakukan perubahan nama agar nama tersebut sesuai dengan nama class, pada permasalahan ke 2 yaitu di constructor tidak ada pemanggilan variabel rambut hanya ada pemanggilan variable nama dengan itu cara mengatasinya dengan menambahkan tipe data String di dalam constructor.

Alasan pengambilan Keputusan ini adalah karena telah di analisis terlebih dahulu dan sudah di praktekan dan ternyata hasil nya berhasil atau kode nya berjalan.

[No. 3] Identifikasi Masalah:

3) Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
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");  
    }  
}
```

Pada soal masih ada pesan kesalahan yaitu yang terdapat pada construsktor dan method berbeda dengan nama class kode:

1. Penamaan deklarasi konstruktur di constructor penamaan nya Manusia1 sedangkan nama kelas nya Manusia tidak menggunakan angka **(1)**.
2. Pada permasalahan kedua adalah hanya pengurangan tipe data di method yaitu tipe data String untuk memanggil perintah dari hobi menonton.
3. Pada permasalahan ketiga kurang tipe data juga untuk memanggil episode dan durasi pada permasalahan tersebut menggunakan tipe data int karena episode dan durasi menggunakan variable angka dan di method int sukaNonton juga kurang menambahkan return pada method kedua.

[No.3] Analisis dan Argumentasi

1. Saya mengusulkan permasalahan ini karena kode tersebut telah di coba atau di jalankan dan mendapatkan hasil output yaitu menampilkan nama dan warna rambut.
2. Pada kode tersebut terdapat beberapa kesalahan/kekurangan tipe data untuk memanggil kode tersebut ke output seperti pada method pertama dan kedua, di setiap method terdapat kurang tipe data pada method pertama kekurangan tipe data String, dan pada method kedua kurang tipe data int untuk memanggil episode dan durasi.
3. Selanjutnya pada tugas 3.2 mengubah kode tersebut sesuai dengan perilaku pribadi

Alasan solusi ini karena kode tersebut telah di analisis dan sudah di jalankan dan kode tersebut berhasil menampilkan output yang diinginkan.

[No.3] Penyusunan Algoritma dan Kode Program

5) Algoritma

langkah-langkah penyelesaian masalah.

1. Start
2. Membaca kesalahan yang tampil pada output
3. Melakukan perubahan constructor
4. Melakukan Penambahan tipe data pada constructor dan method
5. Jalankan kode program
6. Selesai

6) Kode program:

```
public class Manusia {  
    //deklarasi atribut Manusia dalam variabel  
    String nama, rambut;  
  
    //deklarasi constructor  
    public Manusia(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);  
    }  
  
    int sukaNonton(int episode, int durasi){  
        return 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 :

Nama saya : Putri

Warna Rambut : hitam

Hobi menonton : Drakor

Jam nonton : 4

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

```
public class Manusia {  
    //deklarasi atribut Manusia dalam variabel  
    String nama, rambut;  
  
    //deklarasi constructor  
    public Manusia(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);  
    }  
  
    int sukaNonton(int episode, int durasi){  
        return 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");  
    }  
}
```

(Gambar 3.1 perbaikan kode program)

Luaran:

```
Nama saya : Putri  
Warna Rambut : hitam  
Hobi Menonton : Drakor  
Jam nonton = 4 jam
```

Compiled and executed in 1.809 sec(s)

(Gambar hasil output 3.1 perbaikan kode class)

```
public class Manusia {  
    //deklarasi atribut Manusia dalam variabel  
    String nama, rambut;  
  
    //deklarasi constructor  
    public Manusia(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);  
    }  
  
    int sukaNonton(int episode, int durasi) {  
        return episode*durasi;  
    }  
  
    //deklarasi method utama  
    public static void main( String[] args) {  
        Manusia satu = new Manusia("M. Bagas Arjuna", "hitam");  
        satu.sukaNonton("Film action");  
        int jumlahJam = satu.sukaNonton(1, 2);  
        System.out.println("Jam nonton = " +jumlahJam + " jam");  
    }  
}
```

(Gambar 3.2 kode program)

```
Nama saya : M. Bagas Arjuna
Warna Rambut : hitam
Hobi Menonton : Film action
Jam nonton = 2 jam
```

Compiled and executed in 1.852 sec(s)

(Hasil output 3.2 kode program)

F) Analisa luaran yang dihasilkan :

Luaran sudah sesuai dengan program yang disusun.

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

Kode program yang di buat bertujuan untuk menyelesaikan masalah yang ada pada kode program dan menggantikan data tersebut dengan sesuai perilaku diri sendiri, selanjutnya kesalahan terdapat pada nama class yang tidak sama dengan constructor dan method, dan juga pada method satu dan dua tidak ada tipe data untuk memanggil perintah ke output.

[No.3] Kesimpulan

2) Analisa

e) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

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

Pada program ini saya menganalisis permasalahan yang terdapat pada kode programan yang telah di berikan, setelah di analisis terdapat permasalahan-permasalahan atau ada kode program yang kurang dan tidak cocok pada permasalahan pertama saya jumpai di penamaan untuk pemanggilan nama class yaitu pada constructor dan method jadi untuk penyelesaian nya hanya melakukan perubahan nama agar nama tersebut sesuai dengan nama class.

Berikut Analisa perbedaan dari

- a) constructor overloading dan overriding
- b) method overloading, dan method overriding
- c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

• **Constructor Overloading:**

- Constructor overloading terjadi ketika sebuah kelas memiliki lebih dari satu konstruktor dengan nama yang sama tetapi dengan parameter yang berbeda (jumlah atau jenis).
- Tujuannya adalah untuk memberikan fleksibilitas dalam menciptakan objek dengan cara yang berbeda, misalnya, dengan memberikan nilai default atau nilai yang berbeda saat objek dibuat.

• **Constructor Overriding:**

- Constructor overriding tidak berlaku karena konstruktor tidak dapat diwarisi seperti metode. Setiap kelas memiliki konstruktor yang harus didefinisikan sendiri.
- Namun, jika Anda menggunakan pewarisan, kelas anak dapat memanggil konstruktor induk menggunakan `super()`, tetapi tidak mengoverride-nya.

Method Overloading dan Method Overriding

• **Method Overloading:**

- Method overloading terjadi ketika dua atau lebih metode dalam kelas yang sama memiliki nama yang sama tetapi dengan parameter yang berbeda (jumlah atau jenis).
- Ini memungkinkan pengembang untuk menggunakan nama metode yang sama untuk berbagai jenis input, meningkatkan keterbacaan kode.

• **Method Overriding:**

- Method overriding terjadi ketika subclass (kelas anak) memiliki metode yang sama dengan nama dan parameter yang sama dengan superclass (kelas induk).

- Tujuannya adalah untuk memberikan implementasi yang berbeda dari metode yang telah didefinisikan dalam kelas induk, memungkinkan polimorfisme.

c) Method yang Mengembalikan Nilai dan Method Tidak Mengembalikan Nilai

- **Method yang Mengembalikan Nilai:**

- Method ini memiliki tipe pengembalian yang didefinisikan, dan menggunakan keyword return untuk mengembalikan nilai kepada pemanggil.
- Contoh: `int sukaNonton(int episode, int durasi)` di kelas Manusia yang mengembalikan hasil perhitungan.

- **Method Tidak Mengembalikan Nilai:**

- Method ini didefinisikan dengan tipe pengembalian void, yang menunjukkan bahwa metode tersebut tidak mengembalikan nilai.
- Contoh: `void sukaNonton(String film)` di kelas Manusia yang hanya mencetak informasi tanpa mengembalikan nilai.

Alasan pengambilan Keputusan ini adalah karena telah di analisis terlebih dahulu dan sudah di praktekan dan ternyata hasil nya berhasil atau kode nya berjalan.

[No. 4] Identifikasi Masalah:

- 4) Uraikan permasalahan dan variabel

Tuliskan kembali soal:

```
public class Ortu {    // membuat kelas induk
    void sukaMenonton(String a) { // method induk spesifik
        System.out.println("Nonton " + a);
    }
    void sukaMembaca(String a) { // method induk umum bisa diubah anak
        System.out.println("Suka Baca " + a);
    }
    public static void main(String [] args) {
        System.out.println("Sifat Orang Tua :");
        Ortu objekO = new Ortu(); // memanggil objek induk
        objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
        objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah

        System.out.println("\n Sifat Anak :");
        Anak objekA = new Anak(); //memanggil objek anak
        objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang
        diturunkan induk
        objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
        diturunkan tanpa deklarasi ulang di anak
    } }

class Anak extends Ortu {
    void sukaMenonton(int a, String b) {
        System.out.println("Nonton Jam " + a + " Malam " + b);
    }
    void sukaMenonton(String a) { // method induk spesifik
        System.out.println("Nonton " + a);
    }
    void sukaMembaca(String a) { // method induk umum bisa diubah anak
        System.out.println("Suka Baca " + a);
    }
    public static void main(String [] args) {
        System.out.println("Sifat Orang Tua :");
        Ortu objekO = new Ortu(); // memanggil objek induk
        objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
        objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah

        System.out.println("\n Sifat Anak :");
        Anak objekA = new Anak(); //memanggil objek anak
        objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang
        diturunkan induk
        objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis
        diturunkan tanpa deklarasi ulang di anak
    } }
```

luaran yang dihasilkan:

Sifat Orang Tua :

Nonton Berita

Suka Baca Koran

Sifat Anak :

Nonton Jam 9 Malam Film Drakor

Suka Baca Komik One Piece

Pada soal melakukan evaluasi terhadap method yang dimiliki yaitu pada class anak extends Ortu dengan method di class Ortu apakah method tersebut sudah efisien.

Kemudian melakukan running kode program tersebut ke dalam compiler Jdoodle dan catat waktu eksekusinya.

[No.4] Analisis dan Argumentasi

Analisis kode:

1. Struktur Kelas dan Pewarisan

- **Kelas Induk (Ortu):**

Kelas ini memiliki dua metode: `sukaMenonton(String a)` dan `sukaMembaca(String a)`.

Keduanya dirancang untuk memberikan perilaku umum yang dapat digunakan oleh kelas anak.

- **Kelas Anak (Anak):**

Kelas ini mewarisi dari Ortu dan mendefinisikan metode baru serta mengoverride metode yang sudah ada. Ini menunjukkan penggunaan pewarisan dengan baik, tetapi ada beberapa masalah.

2. Redundansi Metode

- **Redefinisi Metode:**

Kelas Anak mendefinisikan kembali metode `sukaMenonton(String a)` dan `sukaMembaca(String a)`, padahal metode tersebut sudah ada di kelas Ortu.

Hal ini menambah kompleksitas dan bisa membingungkan pembaca kode. Jika tidak ada perubahan logika dalam metode tersebut, tidak perlu mengulang definisi.

3. Penggunaan Overloading dan Overriding

- **Overloading:**

Metode `sukaMenonton(int a, String b)` di kelas Anak adalah contoh baik dari overloading, karena menambahkan variasi dalam penggunaan metode dengan parameter berbeda.

- **Overriding:**

Penggunaan overriding seharusnya lebih terfokus. Jika Anda mengubah perilaku dari `sukaMenonton(String a)`, lebih baik untuk memanggil metode induk menggunakan `super` untuk menghindari duplikasi logika.

4. Pengulangan Kode

- **Dua main Method:**

Keduanya memiliki blok kode yang sangat mirip untuk menampilkan sifat orang tua dan anak. Ini menunjukkan adanya pengulangan yang dapat dihindari dengan mendefinisikan satu main method di kelas terpisah (misalnya, kelas Main).

[No.4] Penyusunan Algoritma dan Kode Program

7) Algoritma

langkah-langkah penyelesaian masalah.

1. Start

2. Menjalankan kode tersebut ke dalam compiler

3. Melakukan perbaikan penulisan kode method untuk mendapatkan waktu yang efisien

4. Jalankan kode program

5. selesai

8) Kode program:

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

```
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
} }
```

(Gambar 4.1 Mengevaluasi kode program)

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 1.536 sec(s)
```

(Gambar hasil output 4.1 Mengevaluasi kode program)

```
1 public class Ortu {
2     // Method di kelas induk
3     void sukaMenonton(String a) { // method induk spesifik
4         System.out.println("Nonton " + a);
5     }
6
7     void sukaMembaca(String a) { // method induk yang dapat diubah anak
8         System.out.println("Suka Baca " + a);
9     }
10
11     public static void main(String[] args) {
12         System.out.println("Sifat Orang Tua :");
13         Ortu objekO = new Ortu(); // memanggil objek induk
14         objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
15         objekO.sukaMembaca("Koran"); // memanggil method induk
16
17         System.out.println("\nSifat Anak :");
18         Anak objekA = new Anak(); // memanggil objek anak
19         objekA.sukaMenonton(9, "Film Drakor"); // memanggil method yang overload
20         objekA.sukaMembaca("Komik One Piece"); // memanggil method induk yang diwariskan
21     }
22 }
23
24 class Anak extends Ortu {
25     // Method overloading dengan parameter tambahan
26     void sukaMenonton(int a, String b) {
27         System.out.println("Nonton Jam " + a + " Malam " + b);
28     }
29 }
30
```

(Gambar 4.2 perubahan kode yang efisien)

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

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

Compiled and executed in 1.227 sec(s)

(Hasil output 4.2 kode program)

- H) Analisa luaran yang dihasilkan :
- Luaran sudah sesuai dengan program yang disusun.
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.
- I) Penulisan ulang kode membuat kode tersebut menjadi lebih efisien agar codingan tersebut mudah di baca pada gambar 4.1 dan 4.2 terdapat perbedaan kode yang Dimana pada gambar 4.1 tampilan kode terlalu banyak tetapi berhasil berjalan tetapi waktu yang di hasilkan untuk menampilkan kode memakan banyak waktu, sedangkan pada gambar 4.2 melakukan perubahan kode atau membuat kode tersebut menjadi lebih mudah di baca dan membuat kode tersebut secara efisien terbukti dengan Ketika kode tersebut di jalankan hanya membutuhkan waktu 1.227 sec

[No.4] Kesimpulan

3) Analisa

- A. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- B. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Pada program ini saya melakukan Penulisan ulang kode membuat kode tersebut menjadi lebih efisien agar codingan tersebut mudah di baca pada gambar 4.1 dan 4.2 terdapat perbedaan kode yang Dimana pada gambar 4.1 tampilan kode terlalu banyak tetapi berhasil berjalan tetapi waktu yang di hasilkan untuk menampilkan kode memakan banyak waktu, sedangkan pada gambar 4.2 melakukan perubahan kode atau membuat kode tersebut menjadi lebih mudah di baca dan membuat kode tersebut secara efisien terbukti dengan Ketika kode tersebut di jalankan hanya membutuhkan waktu 1.227 sec.

Dasar alasan dalam pengambilan Keputusan ini adalah karena sebelum melakukan saya terlebih dahulu menganalisis bagaimana agar kode tersebut bisa berjalan lebih efisien dari kode sebelumnya.

Refleksi

Pengalaman belajar tentang class,method,objek dari seluruh tugas yang di berikan lumayan membingungkan karena butuh waktu untuk mempelajari materi nya agar Ketika di berikan Kembali pertanyaan-pertanyaan saya bisa menjawabnya apalagi Ketika code tersebut terjadi suatu permasalahan maka kita harus bisa menyelesaikan nya.