

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
[Nomor Soal] Identifikasi Masalah:		
<ol style="list-style-type: none">1) Uraikan permasalahan dan variabel2) 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		
<ol style="list-style-type: none">1) Uraikan rancangan solusi yang diusulkan.2) Analisis solusi, kaitkan dengan permasalahan.		
[Nomor Soal] Penyusunan Algoritma dan Kode Program		
<ol style="list-style-type: none">1) Rancang desain solusi atau algoritma2) Tuliskan kode program dan luaran<ol style="list-style-type: none">a) Beri komentar pada kodeb) Uraikan luaran yang dihasilkanc) Screenshot/ Capture potongan kode dan hasil luaran		
[Nomor Soal] Kesimpulan		
<ol style="list-style-type: none">1) Analisa<ol style="list-style-type: none">a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?2) Evaluasi<ol style="list-style-type: none">a) Apa konsekuensi dari skenario pemrograman ini?b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)3) Kreasi<ol style="list-style-type: none">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)		

Contoh Jawaban:

Nama & NPM	Topik:	Tanggal:
Vivilia Regita Pramesti G1F024060	Kelas Java	17 september 2024

[No. 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

```
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!

Jawab :

```
package Manusia;
```

```
public class Manusia {
```

```
    // Deklarasi atribut Manusia
```

```
    String nama, rambut;
```

```
    // Constructor dengan dua parameter
```

```
    public Manusia(String nama, String rambut) {
```

```
        nama = nama; // Menginisialisasi atribut nama
```

```
        rambut = rambut; // Menginisialisasi atribut rambut
```

```

        System.out.println("Nama saya : " + nama +
                            "\nWarna Rambut : " + rambut);
    }

    // Method utama

    public static void main(String[] args) {

        // Membuat objek dari kelas Manusia dengan parameter yang benar

        Manusia satu = new Manusia("Vivilia Regita Pramesti", "hitam");

    }

}

```

[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara langkah pertama untuk memperbaiki kesalahan pada kode kelas Manusia adalah memastikan bahwa constructor memiliki nama yang sama dengan kelas. Oleh karena itu, constructor diubah menjadi public Manusia (String nama, String rambut), dengan dua parameter untuk memulai atribut nama dan rambut. this adalah keyword yang membantu membedakan parameter dan atribut kelas, sehingga nilai dapat diberikan dengan benar.
- 2) Alasan solusi ini karena menggunakan ini pada constructor dan method memungkinkan program untuk membedakan antara parameter yang diterima dan atribut kelas, sehingga memastikan bahwa nilai atribut diinisialisasi dengan benar sesuai dengan data yang diberikan saat objek dibuat. Ini penting untuk menghindari kebingungan dan kesalahan dalam pengelolaan data objek.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 Algoritma adalah langkah-langkah penyelesaian masalah.
 Misalkan algoritma memasak mi instan:
 - (a) Mulai
 - (b) Deklarasi atribut nama dan rambut
 - (c) Buat constructor dengan parameter nama dan rambut
 - (d) Buat objek Manusia dengan data pribadi
 - (e) Selesai.
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
 Beri komentar pada kode yang di Screenshot

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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");  
    }  
}
```

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!

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah definisi constructor kelas Ortu agar sesuai dengan parameter yang diinginkan
- 2) Alasan solusi ini karena dapat memperbaiki program

[No.2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - 1) Mulai
 - 2) Deklarasi kelas
 - 3) Deklarasi atribut
 - 4) Buat constructor
 - 5) Buat metode main
 - 6) Tampilkan output
 - 7) Selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaran
Beri komentar pada kode yang di Screenshot

```

1 package Ortu;
2
3 public class Ortu {
4     // Deklarasi variabel
5     private String nama;
6     private String rambut;
7     private int tinggi; // tinggi dalam cm
8     private String mata;
9
10    // Deklarasi constructor yang menerima nama, rambut, tinggi, dan mata
11    public Ortu(String nama, String rambut, int tinggi, String mata) {
12        this.nama = nama;
13        this.rambut = rambut;
14        this.tinggi = tinggi;
15        this.mata = mata;
16    }
17
18    public void tampilkanInfo() {
19        // Menampilkan informasi
20        System.out.println("Nama saya : " + nama);
21        System.out.println("Warna Rambut : " + rambut);
22        System.out.println("Tinggi badan : " + tinggi + " cm");
23        System.out.println("Warna Mata : " + mata);
24    }
25
26    public static void main(String[] args) {
27        // Membuat objek Ortu
28        Ortu satu = new Ortu("Vivilia Regita Pramesti", "hitam", 163, "hitam");
29        System.out.println("Biodata Ortu:");
30        satu.tampilkanInfo();
31        System.out.println();
32
33        // Membuat objek Anak
34        Anak anakSatu = new Anak("Vivi", "hitam", 145, "Coklat", "Jujur", 14, "Masak");
35        System.out.println("Biodata Anak 1:");
36        anakSatu.tampilkanInfo();
37    }

```

```

32 // Membuat objek Anak
33 Anak anakSatu = new Anak("Vivi", "hitam", 145, "Coklat", "Jujur", 14, "Masak");
34 System.out.println("Biodata Anak 1:");
35 anakSatu.tampilkanInfo();
36 System.out.println();
37
38 // Menggunakan constructor kedua
39 Anak anakDua = new Anak("Vava", "Coklat", 156, "Hitam", "Disiplin", 15, "Bersepeda");
40 System.out.println("Biodata Anak 2:");
41 anakDua.tampilkanInfo();
42
43
44
45
46 // Kelas Anak yang diturunkan dari Ortu
47 class Anak extends Ortu {
48     private String karakter;
49     private int usia; // Tambahkan atribut usia
50     private String hobi; // Tambahkan atribut hobi
51
52     // Constructor dengan semua parameter
53     public Anak(String nama, String rambut, int tinggi, String mata, String karakter, int usia, String hobi) {
54         super(nama, rambut, tinggi, mata); // Memanggil constructor Ortu
55         this.karakter = karakter;
56         this.usia = usia;
57         this.hobi = hobi;
58     }
59
60     // Metode untuk menampilkan informasi anak
61     public void tampilkanInfo() {
62         super.tampilkanInfo(); // Memanggil info dari Ortu
63         System.out.println("Karakter: " + karakter);
64         System.out.println("Usia: " + usia + " tahun");
65         System.out.println("Hobi: " + hobi);
66     }
67 }

```

```

eclipse-workspace - Ortu/src/Ortu/Ortu.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
...
Problems Javadoc Declaration Console
<terminated> Ortu [Java Application] C:\Users\ASUS\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\javaw.exe (19 Sep
Biodata Ortu:
Nama saya : Vivilia Regita Pramesti
Warna Rambut : hitam
Tinggi badan : 163 cm
Warna Mata : hitam

Biodata Anak 1:
Nama saya : Vivi
Warna Rambut : hitam
Tinggi badan : 145 cm
Warna Mata : Coklat
Karakter: Jujur
Usia: 14 tahun
Hobi: Masak

Biodata Anak 2:
Nama saya : Vava
Warna Rambut : Coklat
Tinggi badan : 156 cm
Warna Mata : Hitam
Karakter: Disiplin
Usia: 15 tahun
Hobi: Bersepeda

```

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.2] Kesimpulan)

1) Analisa

- a) Salah satu masalah utama dengan kode program awal adalah kesalahan penamaan constructor dan deklarasi atribut yang tidak lengkap. Kedua masalah ini menyebabkan program tidak dapat dikompilasi. Algoritma memiliki langkah-langkah penyelesaian masalah yang jelas, termasuk pembuatan kelas, atribut, constructor, dan metode utama. Kode program beroperasi dengan baik dan menampilkan informasi yang diinginkan dengan baik berkat perbaikan pada penamaan dan inisialisasi atribut.
- b) Keputusan untuk memperbaiki kode berlandaskan prinsip-prinsip pemrograman yang baik, yaitu konsistensi dalam penamaan dan kejelasan dalam struktur program. Memperbaiki nama constructor menjadi sesuai dengan nama kelas dan mendeklarasikan atribut secara jelas sangat penting untuk memastikan bahwa program dapat dijalankan tanpa kesalahan.

[No. 3] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
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");  
    } }  
}
```

- 2) **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)
```

- 3) **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

[No.3] Analisis dan Argumentasi

- 1) aya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan metode overloading dan memperbaiki struktur kode agar sesuai dengan konvensi Java.
- 2) Alasan solusi ini karena metode overloading memungkinkan kita untuk mendefinisikan beberapa fungsi dengan nama yang sama namun dengan parameter yang berbeda, sehingga meningkatkan fleksibilitas dan keterbacaan kode. Memperbaiki struktur kode juga penting untuk memastikan bahwa tidak ada kesalahan dalam pengenalan metode dan konstruktor.

[No.3] Penyusunan Algoritma dan Kode Program

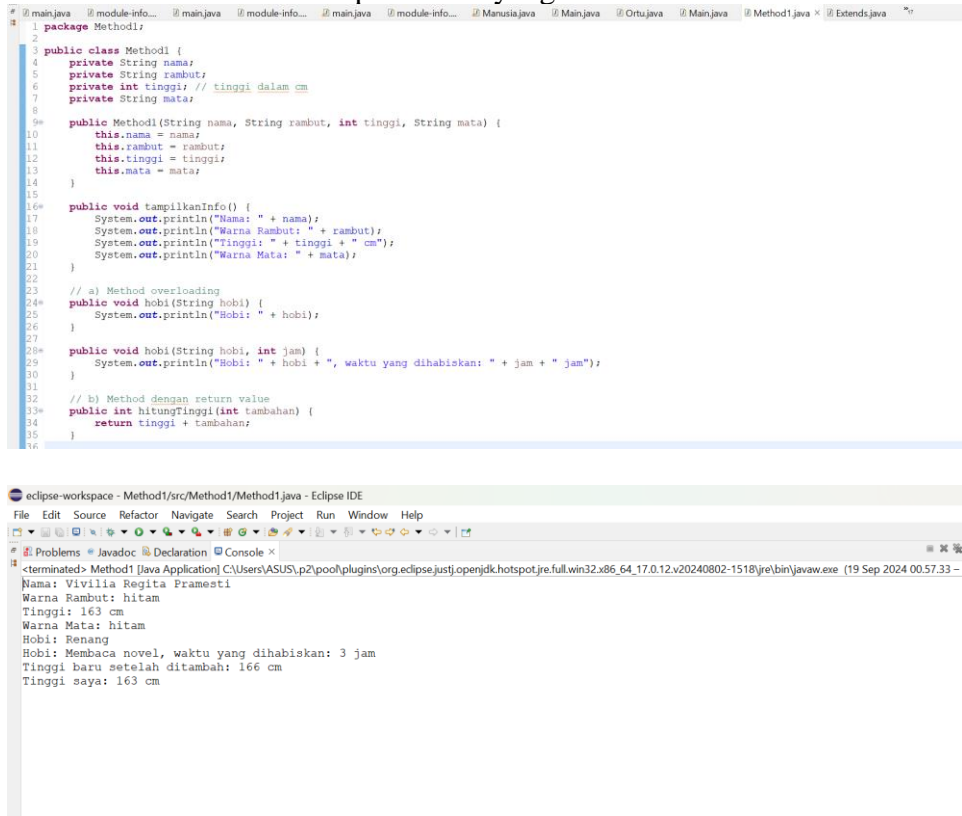
- 3) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

2. Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



```
1 package Method1;
2
3 public class Method1 {
4     private String nama;
5     private String rambut;
6     private int tinggi; // tinggi dalam cm
7     private String mata;
8
9     public Method1(String nama, String rambut, int tinggi, String mata) {
10         this.nama = nama;
11         this.ambut = rambut;
12         this.tinggi = tinggi;
13         this.mata = mata;
14     }
15
16     public void tampilkanInfo() {
17         System.out.println("Nama: " + nama);
18         System.out.println("Warna Rambut: " + rambut);
19         System.out.println("Tinggi: " + tinggi + " cm");
20         System.out.println("Warna Mata: " + mata);
21     }
22
23     // a) Method overloading
24     public void hobi(String hobi) {
25         System.out.println("Hobi: " + hobi);
26     }
27
28     public void hobi(String hobi, int jam) {
29         System.out.println("Hobi: " + hobi + ", waktu yang dihabiskan: " + jam + " jam");
30     }
31
32     // b) Method dengan return value
33     public int hitungTinggi(int tambahan) {
34         return tinggi + tambahan;
35     }
36 }
```

eclipse-workspace - Method1/src/Method1/Method1.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Problems Javadoc Declaration Console

<terminated> Method1 [Java Application] C:\Users\ASUS\AppData\Local\Temp\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\javaw.exe (19 Sep 2024 00:57:33 - I

Nama: Villia Regita Pramesti
Warna Rambut: hitam
Tinggi: 163 cm
Warna Mata: hitam
Hobi: Renang
Hobi: Membaca novel, waktu yang dihabiskan: 3 jam
Tinggi baru setelah ditambah: 166 cm
Tinggi saya: 163 cm

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.3] Kesimpulan

1. Analisa

- Berdasarkan analisis permasalahan yang dihadapi, terdapat kesalahan dalam struktur kode yang menyebabkan program tidak dapat berjalan dengan baik. Dengan menyusun algoritma yang jelas dan sistematis, kita dapat memperbaiki kesalahan tersebut. Penggunaan metode overloading pada kelas Manusia dan Ortu memungkinkan penanganan berbagai jenis input dengan cara yang lebih fleksibel dan terstruktur.
- Dasar alasan pengambilan keputusan dalam kasus ini berfokus pada pentingnya keterbacaan dan pemeliharaan kode. Kode yang terstruktur dengan baik memudahkan pengembang lain untuk memahami dan memelihara program di masa depan. Selain itu, penggunaan prinsip pemrograman berorientasi objek, seperti metode overloading, meningkatkan modularitas dan efisiensi.

[No. 4] Identifikasi Masalah:

]1) Uraikan permasalahan dan variabel

```
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!

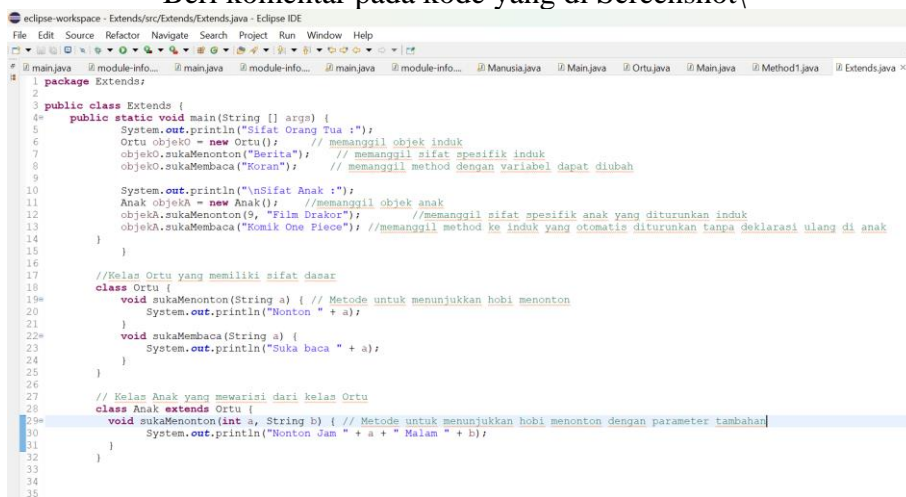
- 4.2. Setelah dirunning di JDoodle, catat waktu eksekusinya.
Susun kembali kode program yang dapat mengefisienkan waktu eksekusi!

[No.4] Analisis dan Argumentasi

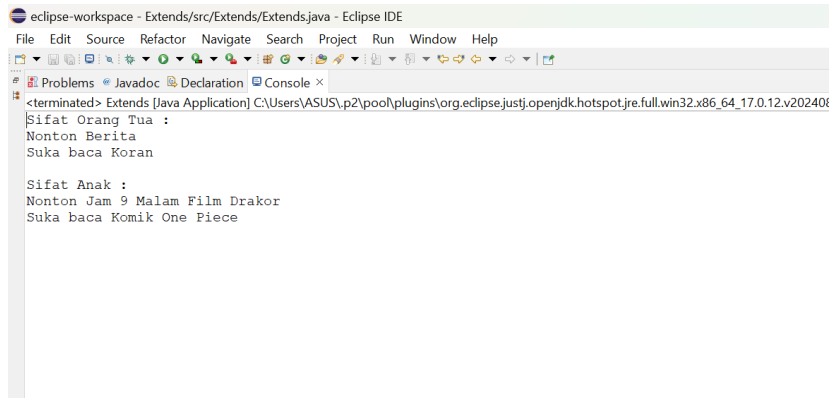
- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengurangi redundansi dalam metode dengan hanya mendefinisikan metode yang diperlukan dan menghindari pengulangan yang tidak perlu.
- 2) Alasan solusi ini karena Hal ini akan membuat kode lebih bersih, lebih mudah dipahami, dan lebih efisien dalam eksekusi. Menghindari duplikasi metode yang tidak perlu juga membantu dalam pemeliharaan kode di masa depan.

[No.4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
 - a) Mulai
 - c) Deklarasi kelas
 - d) Deklarasi atribut
 - e) Buat Metode di Kelas Ortu
 - f) Buat Kelas Anak yang Mengextends Kelas Ortu
 - g) Buat Metode main
 - h) Tampilkan output
 - i) selesai
- 2) Kode program dan luaran
 - a) Screenshot/ Capture potongan kode dan hasil luaranBeri komentar pada kode yang di Screenshot\



```
1 package Extends;
2
3 public class Extends {
4     public static void main(String [] args) {
5         System.out.println("Sifat Orang Tua :");
6         Ortu objekO = new Ortu(); // memanggil objek induk
7         objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
8         objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
9
10        System.out.println("\nSifat Anak :");
11        Anak objekA = new Anak(); // memanggil objek anak
12        objekA.sukaMenonton(5, "Film Drakor"); // memanggil sifat spesifik anak yang diturunkan induk
13        objekA.sukaMembaca("Komik One Piece"); // memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
14    }
15
16    //Kelas Ortu yang memiliki sifat dasar
17    class Ortu {
18        void sukaMenonton(String a) { // Metode untuk menunjukkan hobi menonton
19            System.out.println("Nonton " + a);
20        }
21        void sukaMembaca(String a) {
22            System.out.println("Suka baca " + a);
23        }
24    }
25
26    // Kelas Anak yang mewarisi dari kelas Ortu
27    class Anak extends Ortu {
28        void sukaMenonton(int a, String b) { // Metode untuk menunjukkan hobi menonton dengan parameter tambahan
29            System.out.println("Nonton Jam " + a + " Malam " + b);
30        }
31    }
32
33
34
35}
```

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. The main editor area displays the source code of a Java file named 'Extends.java'. The console window at the bottom shows the output of the program, which includes the text 'Sifat Orang Tua : Nonton Berita Suka baca Koran' and 'Sifat Anak : Nonton Jam 9 Malam Film Drakor Suka baca Komik One Piece'.

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

[No.4] Kesimpulan

1) Analisa

- a) Dalam penyelesaian masalah terkait kelas Ortu dan Anak, kita melihat bahwa penggunaan pewarisan (inheritance) memungkinkan anak untuk mewarisi sifat dan metode dari kelas induk. Metode yang didefinisikan dalam kelas Ortu dapat digunakan langsung di kelas Anak, meskipun kita juga dapat menambah atau mengubah metode yang spesifik untuk anak. Algoritma yang disusun membantu dalam merancang kode program dengan langkah-langkah yang jelas, sehingga memudahkan pemahaman alur kerja program.
- b) utusan untuk menggunakan pewarisan dalam kode ini didasarkan pada prinsip pemrograman berorientasi objek yang mendukung penggunaan kembali kode (code reuse). Dengan mendefinisikan metode di kelas induk, kita dapat menghindari pengulangan kode dan membuat program lebih efisien.

Refleksi

Minggu ini, saya belajar tentang konsep kelas dalam pemrograman Java, terutama mengenai pewarisan dan penggunaan konstruktor. Pengalaman ini sangat berharga karena saya dapat memahami bagaimana cara mendefinisikan atribut dan metode dalam sebuah kelas, serta bagaimana kelas anak dapat mewarisi perilaku dari kelas induk. Saya juga mempelajari pentingnya penggunaan konstruktor untuk menginisialisasi objek dengan parameter yang diperlukan. Melalui latihan, saya menyadari tantangan dalam memahami cara mengatur kode dengan baik dan memperbaiki kesalahan yang muncul, seperti masalah terkait definisi konstruktor dan penggunaan variabel. Salah satu pemaknaan penting yang saya ambil dari pembelajaran ini adalah bagaimana struktur dan organisasi kode dapat memengaruhi efektivitas pemrograman. Keterampilan ini akan sangat berguna untuk pengembangan aplikasi yang lebih kompleks di masa depan. Kesimpulannya, minggu ini memberikan fondasi yang kuat dalam memahami konsep dasar OOP di Java, serta meningkatkan kemampuan saya dalam menyelesaikan masalah melalui pemrograman.