|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Meida Dinafani**  **G1F024058** | **Kelas Java** | **18 September 2024** |

**[No. 1] Identifikasi Masalah:**

**Contoh 1:**  
public class Manusia {  // deklarasi kelas  
    // deklarasi variabel  
    String nama;  
    String rambut;

// deklarasi constructor tanpa parameter  
     public Manusia() {  
          System.out.println("Kelas Manusia tanpa nama");  
     }  
  }

**Latihan 1:**  
1.1.     Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi  
           a. Atribut variabel,

Atribut variabel dalam kelas Manusia adalah:

1. String nama; nama akan menyimpan nama dari objek Manusia.
2. String rambut; rambut akan menyimpan informasi tentang jenis atau warna rambut objek Manusia.

 b. perilaku/ behavior untuk method!

Kelas Manusia hanya memiliki satu metode, yaitu konstruktor tanpa parameter,

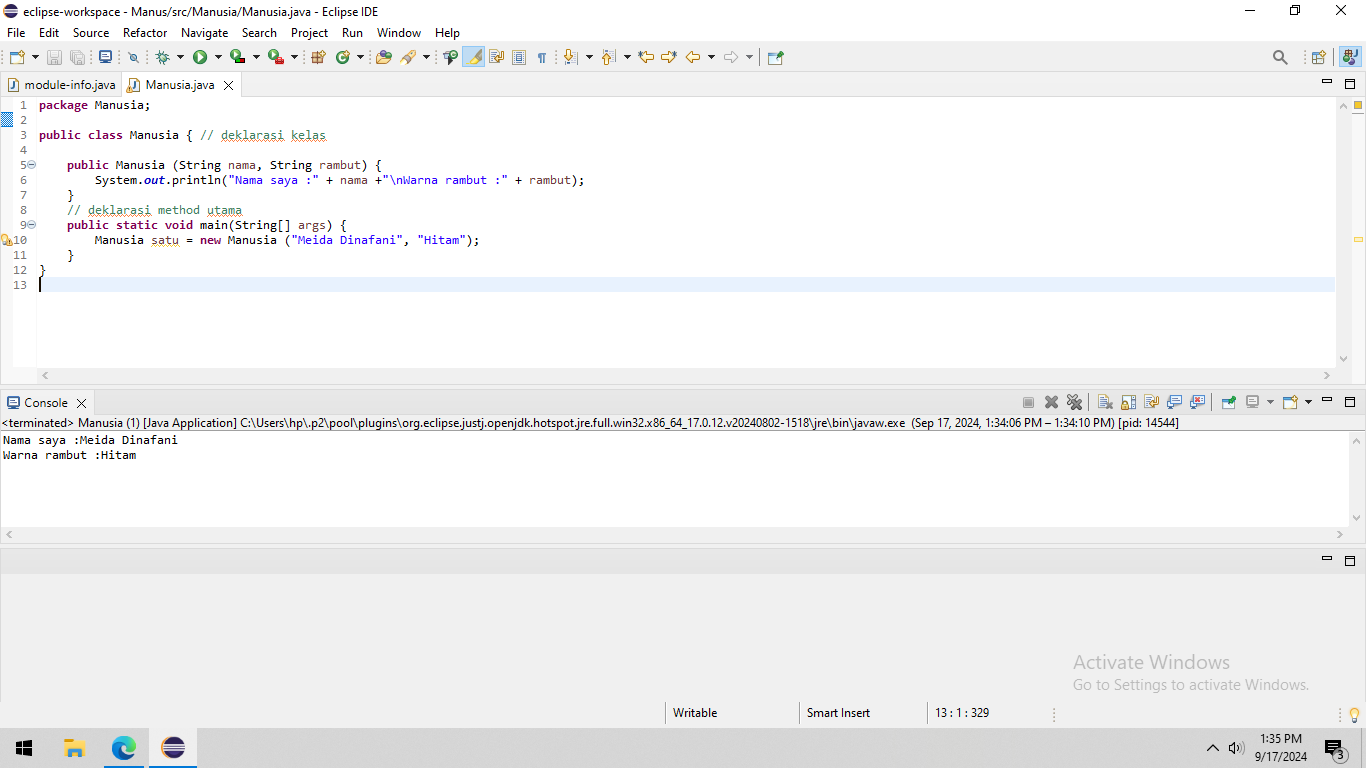
Konstruktor ini adalah metode khusus yang dipanggil ketika sebuah objek Manusia dibuat

**[No.1] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara memanggil konstruktor dan tidak menyimpan data ke dalam atribut kelas.
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan bentuk Kelas.
3. Perbaikan kode program dengan cara
4. Menambahkan konstruktor dengan parameter untuk menginisialisasi atribut saat objek dibuat.
5. Menambahkan getter dan setter untuk mengakses dan mengubah nilai atribut.
6. Menambahkan metode untuk menampilkan informasi objek.

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

1. Algoritma
2. Mulai
3. Deklarasi variabel
4. Inisialisasi variabel
5. Cetak hasil
6. Selesai
7. Kode program dan luaran
8. Screenshot Kode program, hasil luaran, dan Komentar



1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta bentuk kelas yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.1] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Pada program itu saya menggunakan bentuk kelas karena kelas adalah entitas yang menggambarkan keadaan dan perilaku dari objek. Sehingga, kelas memiliki kumpulan objek dengan sifat yang umum. Ada beberapa kode program dalam kelas :

1. Menambahkan konstruktor dengan parameter untuk menginisialisasi atribut saat objek dibuat.
2. Menambahkan getter dan setter untuk mengakses dan mengubah nilai atribut.
3. Menambahkan metode untuk menampilkan informasi objek.

**[No. 2] Identifikasi Masalah:**

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

public class Ortu {  
    //deklarasi 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:**  
Nama saya : Putri  
Warna Rambut : hitam

**Latihan 2:**  
2.1. Susun kembali kode di contoh 2 dengan menambahkan data ciri-ciri Anda di dalam variabel constructor! Pada kode tersebut saya menambahkan data diri seperti :

Nama saya: Meida Dinafani

Warna Rambut: hitam

Usia: 18

Tinggi Badan: 160.0 cm

2.2. Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan?

Sifat (Atribut) yang akan diturunkan:

* Warna Rambut: Keturunan bisa mewarisi warna rambut dari orang tua.
* Tinggi Badan: Genetika dapat mempengaruhi tinggi badan keturunan.
* Hobi: Meskipun hobi bukan sifat genetik, keturunan bisa terinspirasi oleh hobi orang tua.

Constructor untuk Keturunan:

public class Keturunan extends Ortu {

public Keturunan(String nama, String rambut, int tinggi, String hobi) {

super(nama, rambut, tinggi, hobi);

}

}

Perilaku Positif (Behavior) yang akan diturunkan:

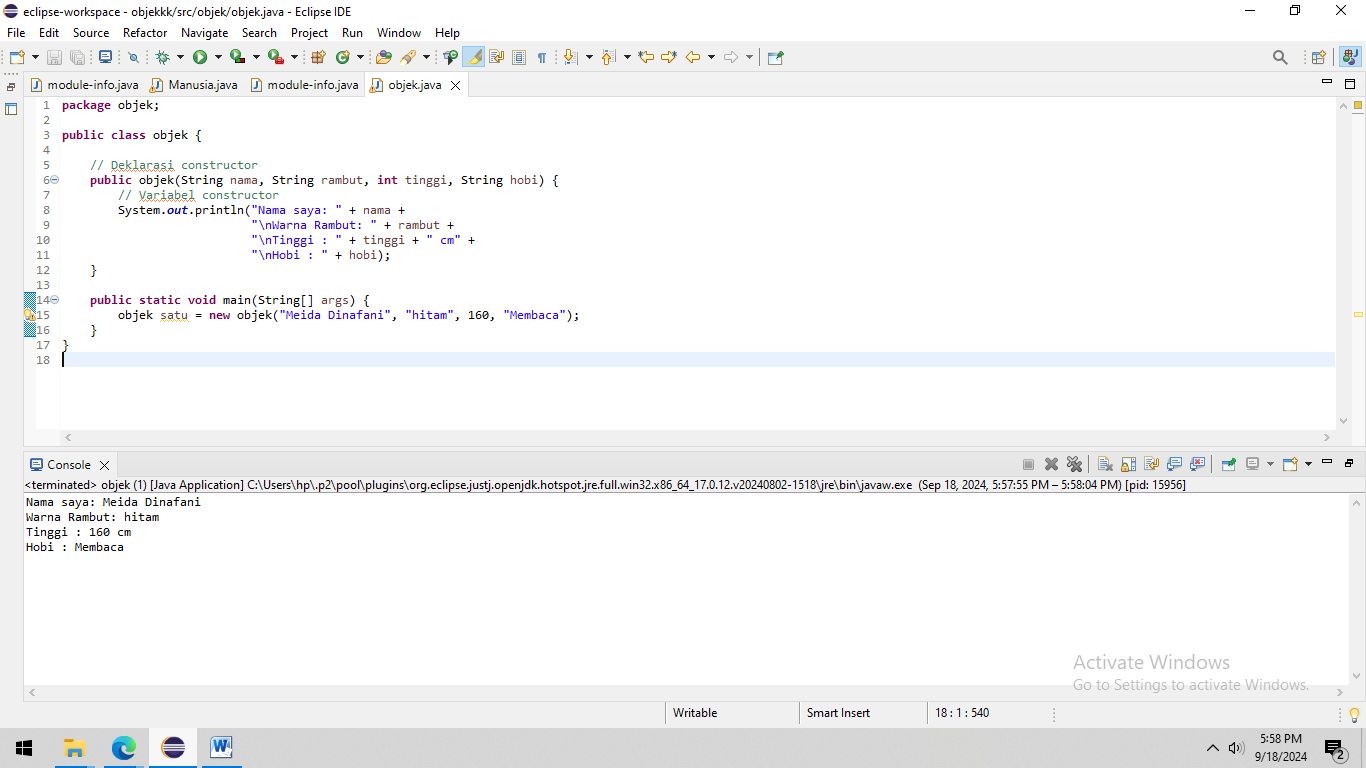
* Kemandirian: Mengajarkan keturunan untuk mandiri dalam berpikir dan bertindak.
* Kreativitas: Jika salah satu orang tua memiliki hobi yang kreatif (seperti melukis), ada kemungkinan keturunan akan mengembangkan kreativitas.
* Kedisiplinan: Menanamkan nilai disiplin dalam kegiatan sehari-hari yang bisa membantu keturunan dalam mencapai tujuan.

**[No.2] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbarui kode program agar menambahkan lebih banyak ciri-ciri dan dapat memberikan informasi lebih lengkap tentang karakter atau identitas.
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan bentuk objek (keturunan dari kelas).
3. Perbaikan kode program dengan cara menyusun kembali kode dengan menambahkan data ciri-ciri ke dalam variabel constructor.

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

1. Algoritma
2. Mulai
3. Definisikan Kelas Objek
4. Panggil Konstruktok
5. Tampilkan hasil
6. Selesai
7. Kode program dan luaran
8. Screenshot Kode program, hasil luaran, dan Komentar



1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta bentuk objek yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.2] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Pada program itu saya menggunakan bentuk Objek  karena Objek adalah bentuk keturunan dari kelas, sehingga otomatis memiliki atribut dan method dari kelas. Dalam kode program tersebut terdapat berbagai penyusunan kode dengan menambahkan data ciri-ciri dalam variabel constructor, serta analisis tentang sifat, constructor, dan perilaku positif yang dapat diturunkan kepada keturunan.

**[No. 3] Identifikasi Masalah:**

**Contoh 3:**Salin dan tempel kode program berikut ke Eclipse atau JDoodle.  
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(String film) {  
        System.out.println(" Hobi Menonton : " + film);  
    }  
              
    //deklarasi method utama  
    public static void main( String[] args) {  
            Manusia satu = new Manusia("Putri", "hitam");  
            satu.sukaNonton("Drakor");  
    }  
}

**Luaran 3:**  
 Nama saya : Putri  
 Warna Rambut : hitam  
 Hobi Menonton : Drakor

**Latihan 3:**  
3.1.  Analisa perbedaan deklarasi constructor, method, dan method utama!

1. Constructor:

* Constructor adalah sebuah metode khusus yang dipanggil ketika objek dibuat.
* Namanya sama dengan nama kelas dan tidak memiliki tipe pengembalian.
* Bertujuan untuk menginisialisasi atribut objek.
* Contoh: public Manusia(String nama, String rambut).

1. Method:

* Method adalah fungsi yang dapat dipanggil untuk melakukan suatu tindakan pada objek.
* Memiliki tipe pengembalian (bisa void atau tipe data lainnya).
* Dapat digunakan untuk menjalankan logika atau memanipulasi data objek.
* Contoh: void sukaNonton(String film)

1. Method Utama (main method):

* Merupakan titik awal eksekusi program Java.
* Selalu harus memiliki signature tertentu: public static void main(String[] args).
* Digunakan untuk memanggil constructor dan method lainnya.
* Contoh: public static void main(String[] args)

3.2.  Tentukan kapan Anda perlu menggunakan constructor dan method?

1. Constructor:

Menggunakan constructor itu saat ingin menginisialisasi atribut objek, saat objek tersebut dibuat. Ini penting untuk memastikan objek berada dalam keadaan yang valid.

1. Method:

Menggunakan method untuk menjalankan logika atau operasi tertentu setelah objek dibuat. Method berguna untuk interaksi dengan objek atau memodifikasi data objek setelah inisialisasi.

3.3.  Uraikan perbedaan berikut:  
        a) Constructor overloading dan overriding

* Constructor Overloading: Memungkinkan lebih dari satu constructor dalam sebuah kelas dengan parameter yang berbeda. Misalnya, Anda bisa memiliki constructor dengan satu parameter dan constructor lainnya dengan dua parameter.
* Overriding: Merupakan proses di mana subclass memiliki metode yang sama dengan superclass-nya dengan implementasi yang berbeda. Ini tidak berlaku untuk constructor karena constructor tidak bisa diwariskan.

b)method overloading, dan method overriding

* Method Overloading: Memungkinkan beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda dalam satu kelas. Ini berguna untuk membuat metode lebih fleksibel.
* Method Overriding: Terjadi ketika subclass memiliki metode yang sama dengan superclass-nya dan memberikan implementasi yang berbeda. Ini memungkinkan subclass untuk mendefinisikan perilakunya sendiri.

        c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

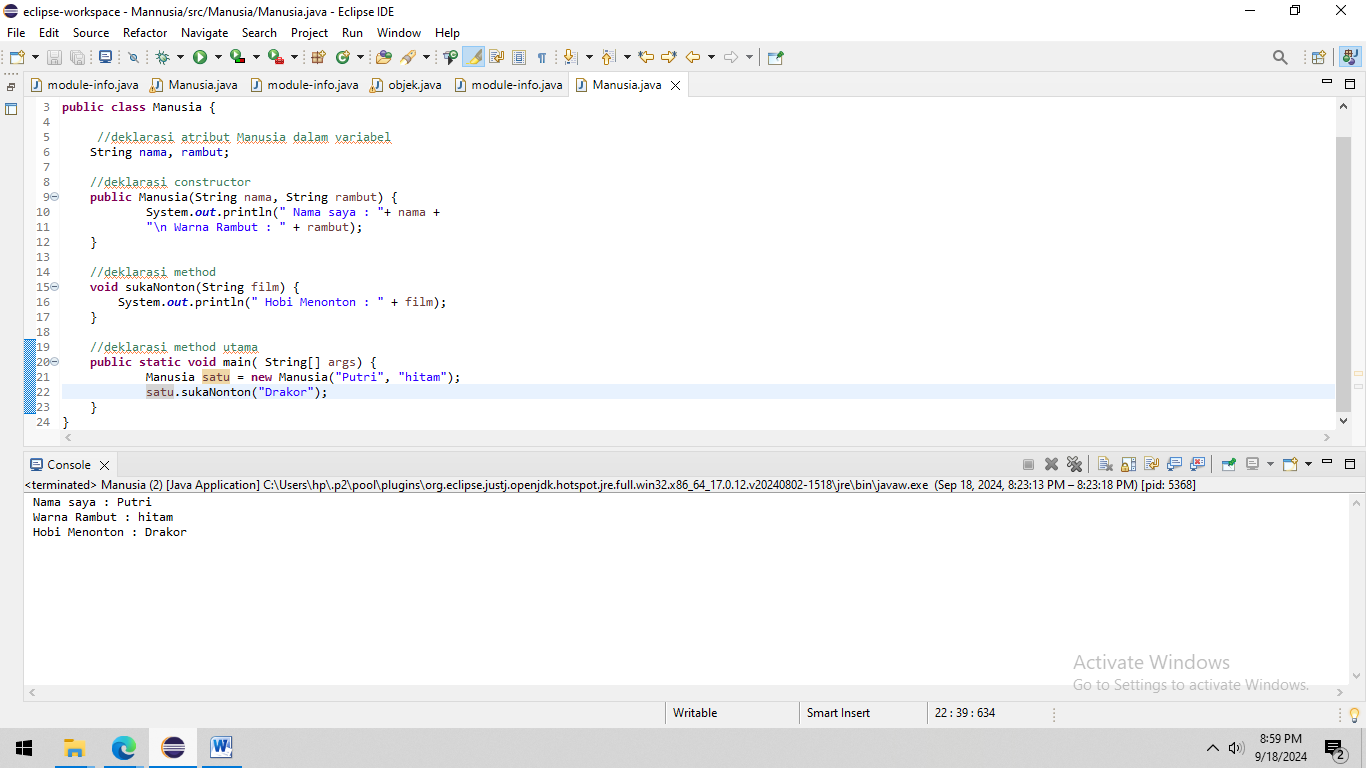
* Method yang Mengembalikan Nilai: Memiliki tipe pengembalian dan mengembalikan nilai menggunakan kata kunci return. Contoh: int getAge() { return age; }.
* Method Tidak Mengembalikan Nilai: Dideklarasikan dengan tipe void dan tidak mengembalikan nilai. Contoh: void printName() { System.out.println(name); }.

**[No.3] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara mendeklarasikan kelas sebagai Manusia. Karena constructor disini dinamai Manusia1.
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan bentuk Method (Perilaku yang dilakukan objek).
3. Perbaikan kode program dengan cara menyamakan Nama Constructor dengan nama kelas, sebagai Manusia.

**[No.3 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma
2. Mulai
3. Deklarasi Kelas
4. Deklarasi Atribut
5. Deklarasi Constructor
6. Deklarasi Method
7. Deklarasi Method Utama
8. Tampilkan hasil
9. Selesai
10. Kode program dan luaran
11. Screenshot Kode program, hasil luaran, dan Komentar



1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta bentuk Method yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.3] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Pada program itu saya menggunakan bentuk Method karena method itu adalah perilaku yang dilakukan objek. Program kelas Manusia menunjukkan penggunaan dasar konsep pemrograman sebagai berikut:

1. Constructor : Digunakan untuk menginisialisasi atribut objek saat objek dibuat. Constructor membantu memastikan bahwa objek berada dalam kondisi yang valid dengan data yang diperlukan.
2. Method : Menyediakan fungsionalitas untuk objek, memungkinkan interaksi dan manipulasi data. Metode sukaNonton memberikan contoh bagaimana objek dapat berperilaku.
3. Method Utama : Merupakan titik awal eksekusi program, berfungsi untuk menciptakan objek dan memanggil metode lainnya.
4. Perbedaan Konsep : Memahami perbedaan antara constructor, method, serta konsep overloading dan overriding penting untuk merancang kelas yang fleksibel dan dapat digunakan kembali.

**[No. 4] Identifikasi Masalah:**

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

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 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. Bandingkan method yang dimiliki class Anak extends Ortu dengan method di class Ortu!

Kelas Ortu:

* sukaMenonton(String a): Metode ini menampilkan informasi tontonan berdasarkan satu parameter (jenis tayangan).
* sukaMembaca(String a): Metode ini menampilkan informasi bacaan berdasarkan satu parameter (jenis bacaan).

Kelas Anak:

* sukaMenonton(int a, String b): Metode ini menampilkan informasi tontonan dengan dua parameter (jam dan jenis tayangan), memberikan informasi yang lebih spesifik.
* sukaMenonton(String a): Mengoverride metode dari Ortu, tetapi dengan logika yang sama. Ini membolehkan pemanggilan dengan satu parameter.
* sukaBermain(String a): Metode baru yang tidak ada di kelas Ortu, menampilkan informasi tentang aktivitas bermain.

4.2. Ubahlah Contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

Penambahan Metode Berbeda di Kelas Anak yaitu Metode Baru sukaOlahraga(String a): dengan menambahkan variasi lain di kelas Anak, yang memberikan informasi tentang olahraga yang disukai. Dengan cara ini,memiliki variasi metode di kelas Anak yang berbeda dari yang ada di Ortu.

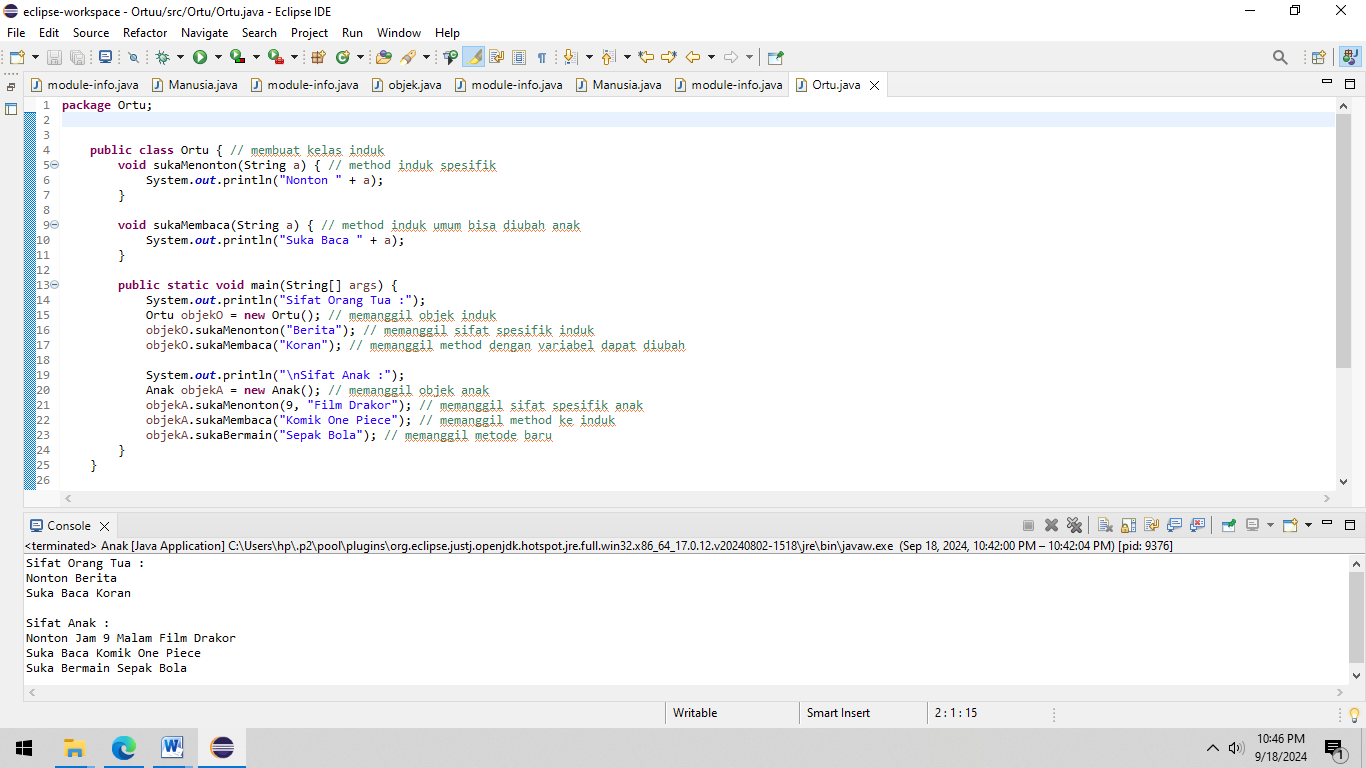
**[No.4] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan objek anak dengan method yang berbeda.
2. Alasan solusi ini karena supaya program dapat berjalan dan dengan menghasilkan bentuk yang berbeda dari yang ada di Ortu.

1. Perbaikan kode program dengan cara Penambahan Metode Berbeda di Kelas Anak yaitu Metode Baru sukaOlahraga(String a): . Dengan cara ini,memiliki variasi metode di kelas Anak yang berbeda dari yang ada di Ortu.

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

1. Algoritma
2. Mulai
3. Deklarasikan Kelas Ortu
4. Deklarasikan Metode main di Kelas Ortu
5. Deklarasikan Kelas Anak yang Mengextends Kelas Ortu
6. Deklarasikan Metode main di Kelas Anak
7. Selesai
8. Kode program dan luaran
9. Screenshot Kode program, hasil luaran, dan Komentar



1. Analisa luaran yang dihasilkan.

Luaran sudah sesuai dengan program yang disusun. Serta bentuk Extends yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

**[No.4] Kesimpulan   
(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)**

1. **Analisa**

Pada program itu saya menggunakan bentuk Extends karena kata kerja dari keturunan yang merujuk ke kelas induk, sehingga otomatis memiliki atribut dan method dari kelas induk.

Program yang dibangun menggunakan konsep pemrograman berorientasi objek, di mana kelas Ortu sebagai kelas dasar menyediakan metode dasar untuk menonton dan membaca, sementara kelas Anak memperluas fungsionalitas tersebut dengan menambahkan metode baru seperti sukaBermain dan sukaOlahraga.

**Refleksi :**

Pengalaman belajar yang didapat yakni mendapatkan pelajaran dan pemahaman mengenai pemrograman yakni Kelas, Objek, Method, dan Extends yang ada di java meskipun ada yang masih belum dipahami. Pengetahuan yang di dapatkan yakni dapat mengetahui bagaimana cara kerja yang ada pada Kelas, Objek, Method, dan Extends . Serta mendapat pengetahuan juga mengenai Coding. Tantangan yang saya dihadapi yakni agak kesusahan dalam melakukan coding dan terkadang terjadi error.