|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MUHAMMAD ARIQ HAFIDL FATANAH G1F024014** | **Kelas Java** | **16 September 2024** |

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

1. Uraikan permasalahan dan variabel

Tuliskan kembali soal:

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, dan  
           b. perilaku/ behavior untuk method!

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

Analisis ciri umum Kelas Manusia:

1. Atribut variable (karakteristik dari objek kelas manusia)

* Nama
* Rambut
* Mata (opsional)
* Tinggi badan (opsional)
* Dll.

1. Perilaku (Method)

* Berjalan
* Berbicara
* Makan
* Dll.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma kelas manusia:

1. Deklarasi kelas Manusia

2. Deklarasi atribut:

- nama : String

- rambut : String

3. Buat constructor:

- Cetak "Kelas Manusia tanpa nama"

4. Buat method setNama(nama):

- Simpan nilai nama ke dalam atribut 'nama'

5. Buat method getNama():

- Kembalikan nilai 'nama'

6. Buat method berjalan():

- Cetak "[nama] sedang berjalan"

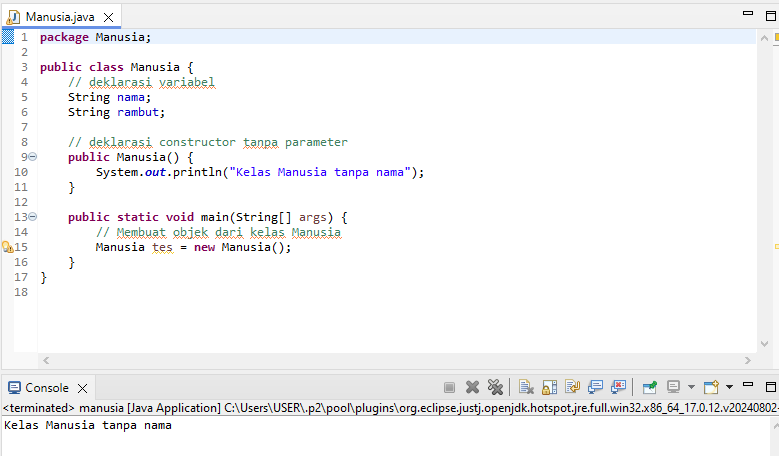
7. Buat method berbicara():

- Cetak "[nama] sedang berbicara"

2) Kode program dan luaran

1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan:

Luaran sudah sesuai dengan program yang disusun. Program ini membuat objek Manusia tanpa memberikan nilai atau parameter apapun, dan karena constructor tanpa parameter berfungsi, ia hanya mencetak pesan yang telah ditentukan dalam constructor. Tidak ada manipulasi atau penggunaan lebih lanjut dari variabel nama dan rambut, sehingga tidak ada keluaran yang berhubungan dengan kedua variabel tersebut.

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

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Kesimpulan Program ini berfungsi sesuai dengan perancangan algoritma. Program berhasil menginisialisasi objek dan memanggil constructor. Namun, variabel nama dan rambut belum digunakan atau diinisialisasi, sehingga program ini masih bisa dikembangkan lebih lanjut dengan menambahkan fungsionalitas seperti setter dan getter untuk mengelola data manusia.

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

1. Uraikan permasalahan dan variabel

Tuliskan kembali soal:

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

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

1. Saya mengusulkan untuk menyusun kembali kode dengan menambahkan data diri saya di dalam variabel, dimana saya akan menambahkan tinggi dan hobi kedalam variable.
2. Alasan saya menambahkan tinggi dan hobi adalah jika saya memiliki keturunan kedua ciri tersebut akan diturunkan kepada keturunan saya.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Algoritma pewarisan sifat dan prilaku:

1. Mulai

2. Inisialisasi Package dan Kelas:

* Buat package Manusia
* Buat kelas Ortu dalam package Manusia

3. Deklarasi Constructor:

Konstruktor Ortu menerima 4 parameter:

* nama (String)
* rambut (String)
* tinggi (String)
* hobi (String)

Di dalam konstruktor, cetak:

* "Nama saya: " diikuti dengan nilai nama
* "Warna Rambut: " diikuti dengan nilai rambut
* "Tinggi: " diikuti dengan nilai tinggi
* "Hobi: " diikuti dengan nilai hobi

4. Metode main:

* Buat objek satu dari kelas Ortu dengan parameter:
* Nama: "M.Ariq Hafidl Fatanah"
* Rambut: "Hitam"
* Tinggi: "171 cm"
* Hobi: "Belajar hal baru"

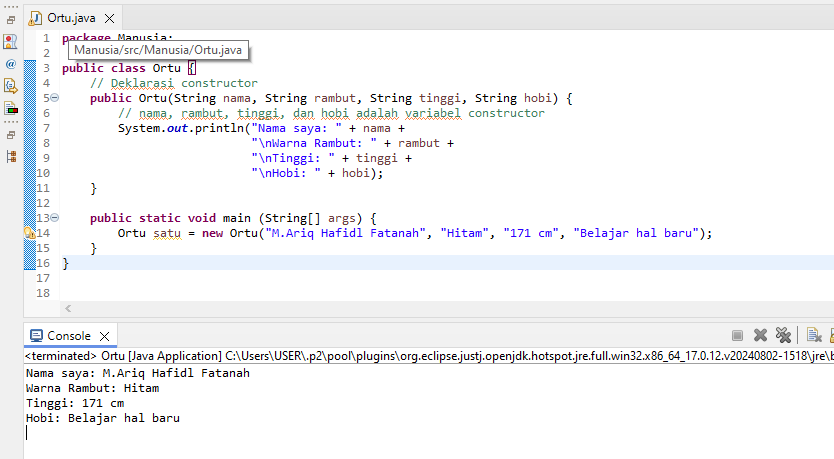
5. Tampilkan Output:

* Output dari konstruktor akan mencetak informasi yang diberikan ke konsol

6. Selesai

1. Kode program dan luaran
2. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan   
   Jawab:

Luaran sudah sesuai dengan program yang disusun. Secara keseluruhan, luaran dari program ini adalah hasil dari pencetakan informasi yang diinputkan ke dalam objek Ortu, menunjukkan bagaimana data dapat diproses dan ditampilkan dengan menggunakan konstruktor dan metode System.out.println.

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

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Program ini bertujuan untuk mendemonstrasikan penggunaan konstruktor dalam Java untuk menyimpan dan menampilkan informasi pribadi. Algoritma yang diterapkan melibatkan inisialisasi package dan kelas, deklarasi konstruktor untuk menerima dan mencetak data atribut seperti nama, warna rambut, tinggi, dan hobi, serta implementasi metode main yang membuat objek Ortu dan mengirimkan data ke konstruktor. Program ini tidak hanya memenuhi tujuan untuk menunjukkan cara kerja konstruktor, tetapi juga memberikan pemahaman yang jelas tentang bagaimana data dapat diolah dan ditampilkan dalam Java.

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

1. 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(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!  
3.2.  Tentukan kapan Anda perlu menggunakan constructor dan method?  
3.3.  Uraikan perbedaan berikut:  
        a) constructor overloading dan overriding  
        b) method overloading, dan method overriding  
        c) method yang mengembalikan nilai dan method tidak mengembalikan nilai

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

1. Analisa perbedaan deklarasi constructor, method, dan method utama!

: Constructor, method, dan method utama (main) dalam Java memiliki perbedaan penting. Constructor digunakan untuk menginisialisasi objek dengan nilai awal dan tidak memiliki tipe pengembalian. Method mendefinisikan perilaku objek dan bisa mengembalikan nilai atau void. Method utama (main) adalah titik awal eksekusi program yang harus bernama main, memiliki tipe pengembalian void, dan menerima parameter String[] args. Masing-masing memiliki peran yang berbeda: constructor untuk setup objek, method untuk aksi objek, dan method utama untuk memulai program.

1. Tentukan kapan Anda perlu menggunakan constructor dan method?

: ketika kita ingin menginisialisasi objek dengan nilai awal atau melakukan setup yang diperlukan segera setelah objek dibuat.

1. .  Uraikan perbedaan berikut:  
           a) constructor overloading dan overriding: Constructor overloading terjadi ketika ada beberapa constructor dalam satu kelas dengan nama yang sama tetapi memiliki parameter yang berbeda (jumlah atau tipe parameter.  
           b) method overloading, dan method overriding: Method overloading terjadi ketika ada beberapa metode dalam satu kelas dengan nama yang sama tetapi dengan parameter yang berbeda (jumlah atau tipe parameter).  
           c) method yang mengembalikan nilai dan method tidak mengembalikan nilai:

Metode yang mengembalikan suatu nilai setelah eksekusi. Metode ini harus memiliki tipe pengembalian yang sesuai dan menggunakan keyword return untuk mengembalikan nilai.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai
2. Deklarasikan kelas Manusia.
3. Deklarasikan atribut nama dan rambut di dalam kelas.
4. Buat constructor Manusia dengan parameter nama dan rambut:

* Cetak "Nama saya: " + nama.
* Cetak "Warna Rambut: " + rambut.

1. Buat method sukaNonton dengan parameter film:

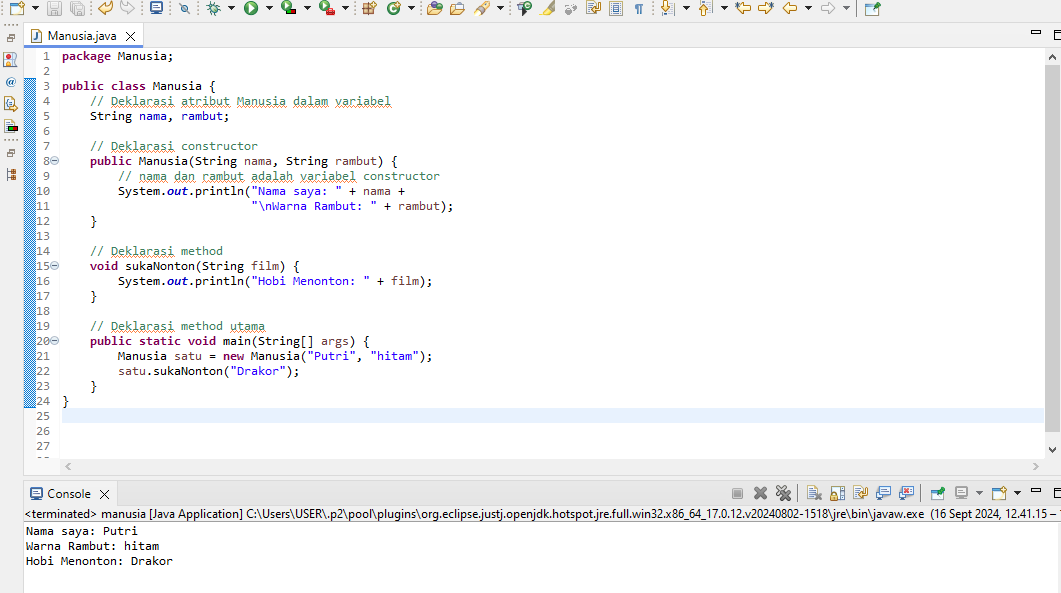
* Cetak "Hobi Menonton: " + film.

1. Dalam metode main:

* Buat objek satu dari kelas Manusia dengan argumen "Putri" dan "hitam".
* Panggil method sukaNonton pada objek satu dengan argumen "Drakor".

1. Selesai
2. Kode program dan luaran
3. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Program Java yang diberikan menghasilkan output yang menunjukkan informasi tentang objek Manusia dan aktivitas yang terkait dengannya.

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

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Program Java ini dirancang untuk menunjukkan penggunaan constructor dan method dalam kelas. Constructor Manusia digunakan untuk menginisialisasi objek dengan nama dan warna rambut, memastikan data penting diset saat objek dibuat. Method sukaNonton kemudian menampilkan hobi menonton film dari objek tersebut. Algoritma program memastikan bahwa data input melalui constructor dan method dikelola dengan baik, menghasilkan output yang sesuai dengan informasi yang diberikan. Keputusan untuk menggunakan constructor berfokus pada inisialisasi objek, sementara method berfokus pada perilaku tambahan, mengikuti prinsip pemrograman berorientasi objek dengan memisahkan setup dan aksi objek untuk kode yang lebih terstruktur dan mudah dipahami

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

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

1. perbandingan menunjukkan bagaimana subclass Anak memperluas dan memodifikasi fungsionalitas yang diwarisi dari kelas induk Ortu melalui method overriding dan penambahan method baru. Kelas Anak menambahkan spesifikasinya sendiri dan memperluas perilaku kelas induk untuk memenuhi kebutuhan tambahan.
2. Saya akan menambahkan method yang berbeda yaitu, method sukaMenulis dan sukaOlahraga, kelas Anak sekarang memiliki perilaku tambahan yang tidak ada di kelas Ortu, memperluas fungsionalitas dan menunjukkan cara subclass dapat menambah atau memodifikasi perilaku yang diwarisi.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.  
Berikut adalah algoritma sederhana:

1. Mulai

2. Deklarasikan kelas `Ortu` dengan method `sukaMenonton` dan `sukaMembaca`.

3. Deklarasikan kelas `Anak` yang mewarisi `Ortu`:

- Tambahkan method `sukaMenonton(int a, String b)`.

- Override method `sukaMenonton(String a)` dari `Ortu`.

- Override method `sukaMembaca(String a)` dari `Ortu`.

- Tambahkan method baru `sukaMenulis(String a)`.

- Tambahkan method baru `sukaOlahraga(String olahraga)`.

4. Di dalam `main`:

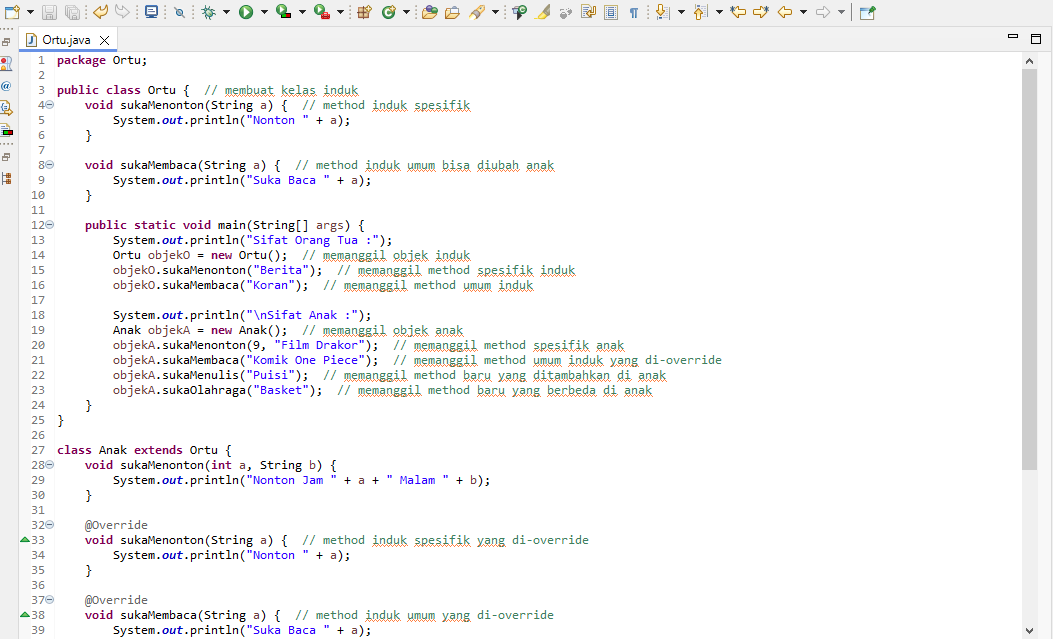
- Tampilkan informasi untuk objek `Ortu`.

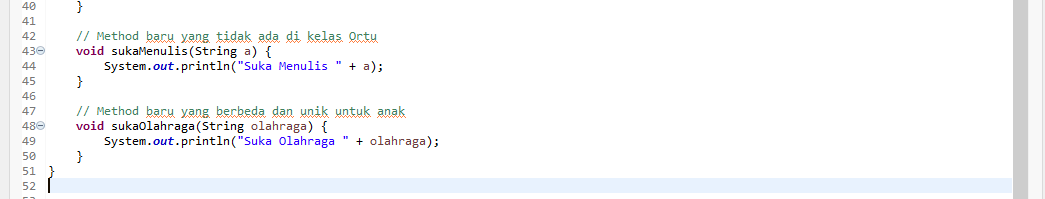
- Tampilkan informasi untuk objek `Anak` dengan semua method yang tersedia.

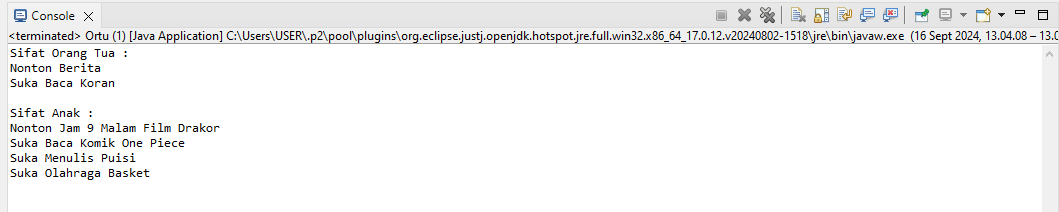
5. Selesai

1. Kode program dan luaran
2. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot







1. Analisa luaran yang dihasilkan :

Luaran sudah sesuai dengan program yang disusun. Program yang dimodifikasi menghasilkan output yang menggambarkan perilaku dan karakteristik dari kedua kelas Ortu dan Anak. Ketika kode dijalankan, output pertama-tama menampilkan informasi dari objek Ortu, di mana method sukaMenonton dan sukaMembaca dipanggil untuk mencetak aktivitas yang relevan. Selanjutnya, objek Anak menampilkan output yang lebih beragam berkat method tambahan dan modifikasi dari kelas induk.

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

1. **Analisa**
2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
3. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawaban Analisa:

Pada program itu saya menyimpulkan dari permasalahan, algoritma, dan kode program adalah bahwa program Java ini efektif dalam menunjukkan konsep pewarisan dan pengembangan method dalam pemrograman berorientasi objek. Kode tersebut mengilustrasikan bagaimana kelas Anak mewarisi dan memodifikasi method dari kelas Ortu, menambahkan method baru dan meng-override method yang ada untuk menyesuaikan perilaku sesuai kebutuhan spesifik. Program ini menghasilkan output yang memverifikasi kemampuan subclass untuk memperluas fungsionalitas yang diwarisi, dengan menampilkan variasi metode yang didefinisikan dalam masing-masing kelas. Pengambilan keputusan untuk menggunakan pewarisan dan method overriding bertujuan untuk memanfaatkan kembali dan memperluas kode, sehingga meningkatkan modularitas dan fleksibilitas dalam desain perangkat lunak.

**Refleksi:**

Hari ini saya belajar tentang konsep kelas dalam Java, terutama mengenai pewarisan dan pengembangan method. Saya memahami bagaimana subclass dapat mewarisi method dari kelas induk, meng-override method tersebut untuk menyesuaikan fungsionalitas, dan menambahkan method baru yang unik untuk subclass. Proses ini memungkinkan pemrograman yang lebih modular dan fleksibel, di mana kode yang ada bisa dimanfaatkan dan diperluas dengan cara yang terstruktur. Ini memberi saya pemahaman yang lebih baik tentang bagaimana mendesain kelas dan mengatur fungsionalitas dalam program Java.