Nama & NPM	Topik:	Tanggal:
AISYA WARDATUL HADI G1F024012	KELAS, OBJEK, METHOD	15 SEPTEMBER 2024

[Nomor 1] Identifikasi Masalah:

1) Pada latihan nomor satu yang membahas tentang kelas, kita akan menganalisa ciri umum kelas manusia yang dapat menjadi atribut, variabel, dan perilaku untuk dijadikan method.

```
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
2) b. perilaku/ behavior untuk method!

[Nomor 1] Analisis dan Argumentasi

1) Kita akan menganalisa ciri umum dari kelas tersebut
2) Membuat method atribut, variabel, dan perilaku

[Nomor 1] Penyusunan Algoritma dan Kode Program

1) Algoritma
   - Mulai
   - Membuka e-learning
   - Menyalin kode program pada e-learning
   - Paste pada website jdoodle
   - Menganalisis kode program yang telah disalin
   - Membuat method baru tentang sifat dan atribut

```
public class Ortu {       // membuat kelas induk
  void sifat(String a) {    // method induk spesifik
    System.out.println("sifatnya " + a);
  }
  void atribut(String a) {     // method induk umum bisa diubah anak
    System.out.println("memakai");
  }

public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu();     // memanggil objek induk
    objekO.sifat("baik");    // memanggil sifat spesifik induk
    objekO.atribut("pakaian ortu");     // memanggil method dengan variabel dapat diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak();    //memanggil objek anak
    objekA.sifat("baik");        //memanggil sifat spesifik anak yang diturunkan induk
    objekA.atribut("pakaian anak anak"); //memanggil method ke induk yang otomatis diturunkan
}    }

class Anak extends Ortu {
  void sifat(int a, String b) {
        System.out.println("sifatnya");
  }
  void atribut(String a) {            // method induk spesifik
        System.out.println("memakai");
  }

public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu();     // memanggil objek induk
    objekO.sifat("baik");        // memanggil sifat spesifik induk
    objekO.atribut("pakaian ortu");     // memanggil method dengan variabel dapat diubah
  }

public static void main(String [] args) {
    System.out.println("Sifat Orang Tua :");
    Ortu objekO = new Ortu();     // memanggil objek induk
    objekO.sifat("baik");        // memanggil sifat spesifik induk
    objekO.atribut("pakaian ortu");     // memanggil method dengan variabel dapat diubah

    System.out.println("\n Sifat Anak :");
    Anak objekA = new Anak();    //memanggil objek anak
    objekA.sifat(9, "baik");        //memanggil sifat spesifik anak yang diturunkan induk
    objekA.atribut("pakaian anak anak"); //memanggil method ke induk yang otomatis diturunkan
    }
}
```

a) Dari kode program diatas terlihat bahwa ada kelas ortu sebagai mahasiswa dan kelas anak turunan. Sifat yang diturunkan dari orang tua yaitu ortu yang bersifat baik dan anak turunan juga bersifat baik

b) Kemudian atribut yang diturunkan ortu memakai pakaian orang tua dan anak anak memakai pakaian anak anak.

[Nomor 1] Kesimpulan

1) Kreasi

a) Pada pemrograman kode program kali ini kita membuat kelas induk dan kelas turunan, dengan mahasiswa sebagai kelas induk

b) Kode program menggunakan tipe data string untuk kalimat, karena tipe data string sendiri digunakan untuk menyimpan kalimat.

c) Kode program telah berjalan dengan baik dan tidak terdapat error .

Nama & NPM	Topik:	Tanggal:
AISYA WARDATUL HADI G1F024012	KELAS, OBJEK, METHOD	15 SEPTEMBER 2024

[No. 2] Identifikasi Masalah:

1) Pada soal nomor 2 kita membuat kode dengan menambahkan ciri ciri pada variabel konstructor
2) Dan apabila nanti memiliki keturunan, analisa sifat (atribut), constructor, dan perilaku positif (behavior) apa yang akan diturunkan

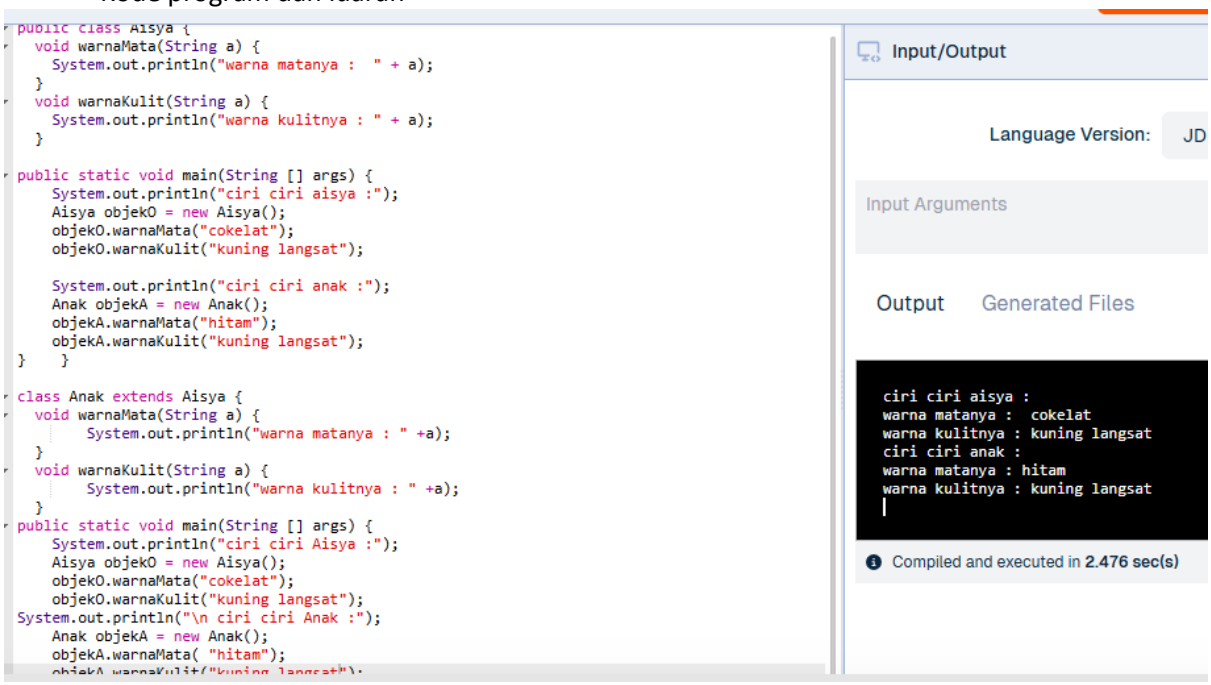
[No.2] Analisis dan Argumentasi

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyalin terlebih dahulu kode program pada e-learning dan menganalisis kode tersebut
2) Memasukkan data diri, dan apabila nanti memiliki keturunan sifat apa saja yang diturunkan kepada keturunan tersebut.

[No.2] Penyusunan Algoritma dan Kode Program

- Algoritma
  (a) Mulai
  (b) Menyalin kode program dari e-learning
  (c) Paste pada website jdoodle
  (d) Menganalisa kode program
  (e) Memasukkan data diri kita pada kode program tersebut
  (f) Menganalisis jika nanti mempunyai keturunan sifat apa yang akan diwarisi kepada keturunan nantinya.
  (g) Menambahkan kelas anak turunan
  (h) Pewarisan sifat
  (i) Mengecek apakah kode program telah berjalan dengan lancar dan tidak terdapat error
  (j) Selesai

- Kode program dan luaran



```
public class Aisya {
   void warnaMata(String a) {
     System.out.println("warna matanya :  " + a);
   }
   void warnaKulit(String a) {
     System.out.println("warna kulitnya : " + a);
   }

public static void main(String [] args) {
     System.out.println("ciri ciri aisya :");
     Aisya objekO = new Aisya();
     objekO.warnaMata("cokelat");
     objekO.warnaKulit("kuning langsat");

     System.out.println("ciri ciri anak :");
     Anak objekA = new Anak();
     objekA.warnaMata("hitam");
     objekA.warnaKulit("kuning langsat");
}    }

class Anak extends Aisya {
   void warnaMata(String a) {
         System.out.println("warna matanya : " +a);
   }
   void warnaKulit(String a) {
         System.out.println("warna kulitnya : " +a);
   }
public static void main(String [] args) {
     System.out.println("ciri ciri Aisya :");
     Aisya objekO = new Aisya();
     objekO.warnaMata("cokelat");
     objekO.warnaKulit("kuning langsat");
System.out.println("\n ciri ciri Anak :");
     Anak objekA = new Anak();
     objekA.warnaMata( "hitam");
```

```
Input/Output

Language Version:  JD

Input Arguments

Output   Generated Files

ciri ciri aisya :
warna matanya :  cokelat
warna kulitnya : kuning langsat
ciri ciri anak :
warna matanya : hitam
warna kulitnya : kuning langsat

Compiled and executed in 2.476 sec(s)
```

- Pada kode program diatas telah menganalisis ciri dari diri sendiri dan ciri apa yang akan diturunkan kepada keturunan nanti
- Dengan memasukkan ciri diri sendiri memiliki mata cokelat dan kulit kuning langsat, dan yang diturunkan kepada keturunan yaitu mata hitam dan warna kulit kuning langsat,

[No.2] Kesimpulan

1) Evaluasi

a) Dari kode program tersebut telah membuat kelas induk dan kelas anak turunan
b) Dengan menganalisa sifat apa saja yang akan diturunkan kepada anak.
c) Kode diatas sudah berjalan dengan lancar dan tidak terdapat error didalamnya

Nama & NPM	Topik:	Tanggal:
AISYA WARDATUL HADI G1F024012	KELAS, OBJEK, METHOD	15 SEPTEMBER 2024

[No. 3] Identifikasi Masalah:

1. Pada soal latihan nomor 3 kita akan membedakan deklarasi konstruktor, method, dan method utama.
2. Menentukan kapan kita perlu menggunakan constructor dan method.
3. Menjelaskan apa itu constructor overloading dan overriding, method overloading dan method overriding, method yang mengembalikan nilai dan method yang tidak mengembalikan nilai.

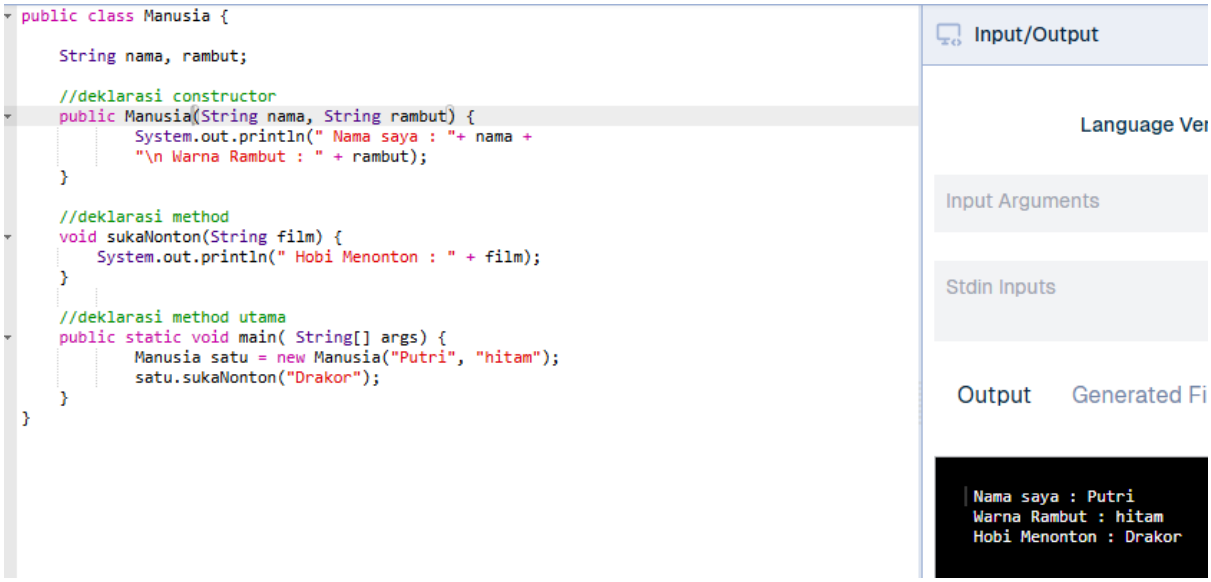
[No.3] Analisis dan Argumentasi

3) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menyalin terlebih dahulu kode program yang telah tertera di e-learning dan menganalisis kode tersebut.
4) Alasan solusi ini karena dengan menganalisis nantinya kita akan memahami bagaimana struktur kode tersebut dan mengetahui operasi yang digunakan.
5) Perbaikan kode program dengan cara menghapus angka 1 pada kode program yang membuat kode tersebut error.

[No.3] Penyusunan Algoritma dan Kode Program

- Algoritma
  - Mulai
  - Membuka e-learning
  - Menyalin kode program dari e-learning ke website jdoodle
  - Menganalisis kode program
  - Pada kode masih terdapat error yaitu pada line 6 penulisan Manusia1 yang seharusnya hanya menuliskan Manusia
  - Menghapus angka 1 pada line 6 yang menyebabkan error program
  - Memahami tentang constructor overloading dan overriding, method overloading dan method overriding, method yang mengembalikan nilai dan method yang tidak mengembalikan nilai.
  - Melampirkannya pada laporan tugas
  - Selesai

- Kode program dan luaran



- Program telah diperbaiki dan Luaran sudah sesuai dengan program yang disusun. Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.
- Constructor overloading adalah adalah teknik di mana suatu kelas dapat memiliki sejumlah konstruktor yang berbeda dalam daftar parameter. Kompilator membedakan konstruktor ini dengan mempertimbangkan jumlah parameter dalam daftar dan tipenya.
- Constructor overriding adalah teknik di mana suatu kelas dapat memiliki sejumlah konstruktor yang berbeda dalam daftar parameter. Kompilator membedakan konstruktor ini dengan mempertimbangkan jumlah parameter dalam daftar dan tipenya.
- Method overloading adalah membuat beberapa method dengan nama yang sama, tapi dibedakan dari jumlah dan/atau tipe parameter.Seharusnya, kita tidak bisa membuat method dengan nama yang sama. Mirip seperti penamaan variabel, compiler Java akan error jika menemukan 2 atau lebih method dengan nama yang sama. Akan tetapi jika jumlah argument dan/atau tipe data argument berbeda, maka akan dianggap sebagai method yang berbeda pula.Method overloading bisa saja terjadi dalam satu class yang sama, dan bisa juga dari class turunan.Method overriding adalah adalah salah satu konsep penting dalam pemrograman berorientasi objek (OOP). Method overriding terjadi ketika kelas anak (subclass) mendefinisikan ulang metode yang sudah didefinisikan di kelas induk (superclass). Dalam artikel ini, kita akan membahas konsep method overriding dalam Java, serta memberikan contoh penggunaan dan outputnya. Method overriding terjadi ketika kelas anak (subclass) memiliki metode dengan nama, parameter, dan tipe pengembalian yang sama seperti metode yang ada di kelas induk (superclass). Ketika objek dari kelas anak membuat pemanggilan ke metode yang diwarisi dari kelas induk, metode dalam kelas anak yang didefinisikan ulang (override) akan dieksekusi, bukan metode dalam kelas induk.Manfaat dari method overriding adalah memungkinkan kelas anak untuk memberikan implementasi khusus untuk metode yang sudah ada di kelas induk, sesuai dengan kebutuhan kelas anak.
- Method yang mengembalikan nilai adalah kita akan mengembalikan sebuah nilai pada fungsi atau method kita lalu niali kembalian tersebut sesuai dengan tipe data yang digunakan.
- Method yang tidak mengembalikan nilai yaitu Void adalah method yang tidak memiliki nilai kembali/return, bisanya digunakan tidak untuk mencari nilai dalam suatu operasi, untuk mendeklarasikannya kita harus menembahkan kata kunci *void*. Agar method tersebut dapat berjalan, kita perlu mamanggilnya pada method *main*, kita harus

membuat objek dari class yang kita gunakan terlibuih dahulu, lalu panggil pada method main.

[No.3] Kesimpulan

1. **Evaluasi**
   - Pada program kali ini kita mengetahui tentang constructor overloading dan overriding, method overloading dan method overriding, method yang mengembalikan nilai dan method yang tidak mengembalikan nilai.
   - Kapan kita menggunakan konstruktor ? Konstruktor adalah metode khusus yang dipanggil saat objek dibuat, sedangkan metode adalah fungsi yang dipanggil pada objek untuk melakukan tugas tertentu. Konstruktor digunakan untuk menginisialisasi status objek, sedangkan metode melakukan tindakan pada status atau perilaku objek.
   - Kapan kita menggunakan method ? memecah program kompleks menjadi bagian-bagian kecil sehingga nantinya dapat kita gunakan secara berulang-ulang tanpa harus menulis baris kode yang sama.

Nama & NPM	**Topik:**	**Tanggal:**
AISYA WARDATUL HADI G1F024012	**KELAS, OBJEK, METHOD**	**15 SEPTEMBER 2024**

[No. 4] Identifikasi Masalah:

1. Menganalisis kode program pada soal latihan nomor 4
2. Mengubah contoh 4 dengan menambahkan objek anak dengan method yang berbeda

[No.4] Analisis dan Argumentasi

- Saya mengusulkan permasalahan ini dapat diatasi dengan cara menganalisis terlebih dahulu tentang objek apa yang akan kita ubah pada kelas anak.

[No4] Penyusunan Algoritma dan Kode Program

- Algoritma
- Mulai
- Menyalin kode program ke website jdoodle
- Menganalisis kode program
- Mengubah dan menambahkan objek anak dengan method yang berbeda
- Mengecek kode program, apakah masih terdapat error atau tidak
- Jika sudah benar, berarti kode program telah berjalan dengan baik.
- Melampirkan kode program pada laporan tugas
- Selesai

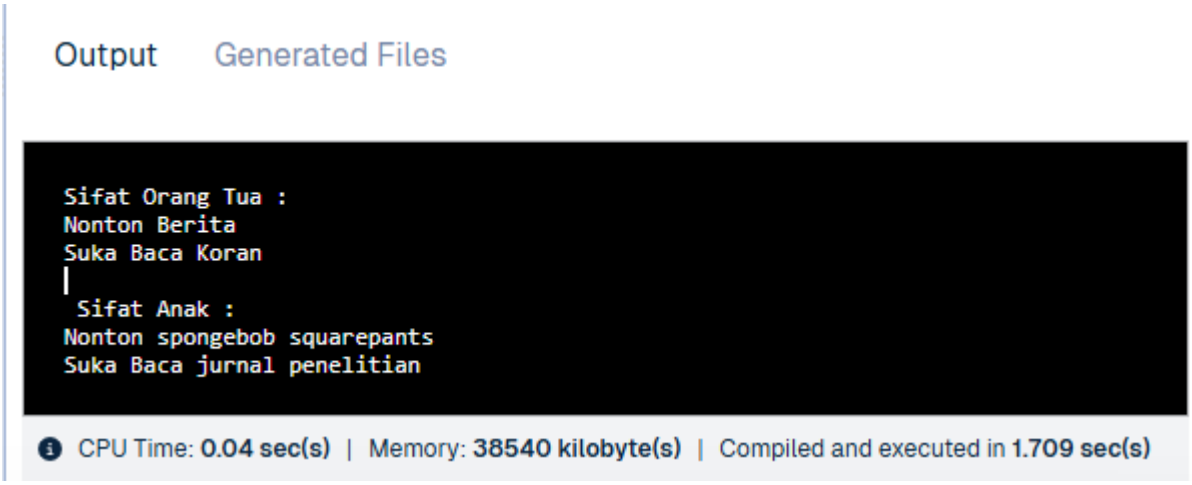
- Kode program dan luaran

```
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( "spongebob squarepants");        //memanggil sifat spesifik anak yan
    objekA.sukaMembaca("jurnal penelitian"); //memanggil method ke induk yang otomatis dituru
}    }

class Anak extends Ortu {
  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
```

```
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("spongebob squarepants");        //memanggil sifat spesifik anak yang
    objekA.sukaMembaca("jurnal penelitian"); //memanggil method ke induk yang otomatis dituru
    }
}
```

3. Pada kode program diatas telah mengubah method anak dari suka menonton drakor, menjadi suka menonton spongebob squarepants, dan dari suka membaca komik one piece menjadi suka membaca jurnal penelitian.

4. Kode sudah berjalan dengan baik dan menghasilkan luaran sesuai dengan yang diinginkan.luaran yang dihasilkan akan terlihat seperti gambar dibawah ini.



[No.4] Kesimpulan

1) **Kreasi**

Dalam membuat kode program kali ini menurut saya sedikit sulit, karena membutuhkan banyak objek dan method yang digunakan, serta membutuhkan ketelitian lebih. Dengan menggunakan kelas induk ortu dan kelas anak turunannya dapat membuat sifat turunan bagi anak dari induknya. Program sudah berjalan dengan baik dan menghasilkan output yang sesuai dengan yang diinginkan. Tidak terdapat error pada kode program,