

Nama & NPM	Topik:	Tanggal:
Sevi Rina Pertiwi G1F024044	Kelas, Objek dan Method	14 September 2024

Latihan 1

[No. 1] Identifikasi Masalah:

- 1) Analisa ciri-ciri umum Kelas Manusia yang dapat menjadi
 - a. Atribut variabel, dan
 - b. Perilaku/behavior untuk method

Jawaban:

- a. Atribut variabel dalam Kelas Manusia yaitu Nama, Umur, Jenis Kelamin, Alamat, Nomor KTP, Tinggi Badan, Berat Badan, dan Pekerjaan.
- b. Perilaku/behavior untuk method manusia yaitu Berbicara, Makan, Tidur, Bekerja, Minum, Belajar, Menonton, Bermain, Mandi, dan lain-lain

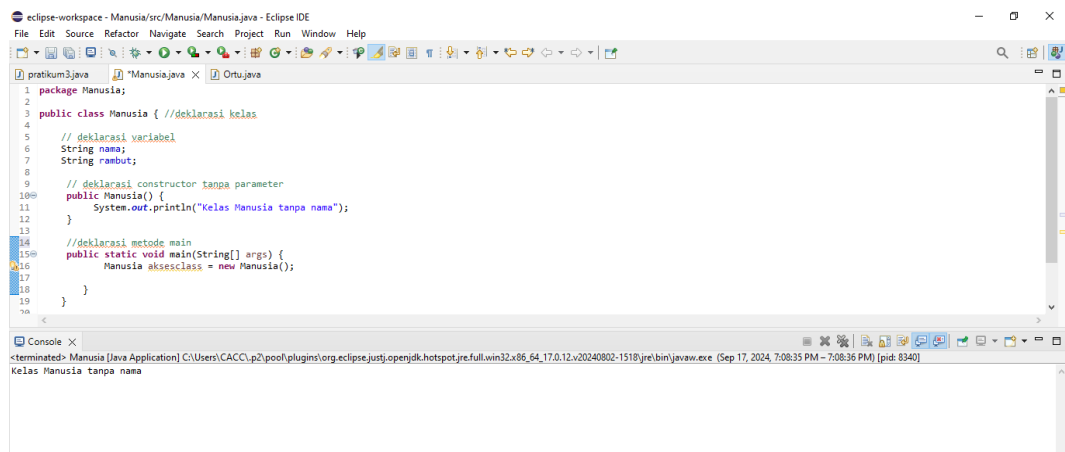
[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan public static void main(String[] args) {
- 2) Alasan solusi ini karena agar dapat memanggil isi dari class Manusia dan menampilkan luarannya.
- 3) Perbaikan kode program dengan cara menambahkan kode


```
public static void main(String[] args) {
    Manusia aksesclass = new Manusia();
```

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Membuat nama file project java
 - (b) Membuat nama class dan main java
 - (c) Deklarasikan method utama pada program
 - (d) Membuat public Manusia
 - (e) Mengisi system.out.println dan menambahkan metode main
 - (f) Menampilkan hasil luaran.
- 2) Kode program dan luaran
 - a) Kode program, komentar dan hasil luaran



```

eclipse-workspace - Manusia/src/Manusia/Manusia.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
pratikun3.java Manusia.java Ortut.java
1 package Manusia;
2
3 public class Manusia { //deklarasi kelas
4
5     // deklarasi variabel
6     String nama;
7     String rambut;
8
9     // deklarasi constructor tanpa parameter
10    public Manusia() {
11        System.out.println("Kelas Manusia tanpa nama");
12    }
13
14    //deklarasi metode main
15    public static void main(String[] args) {
16        Manusia aksesclass = new Manusia();
17    }
18 }
19
20
Console X
<terminated> Manusia [Java Application] C:\Users\CACCI\p2\prof\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1510\jre\bin\javaw.exe (Sep 17, 2024, 7:08:35 PM - 7:08:36 PM) [pid: 8340]
Kelas Manusia tanpa nama
  
```

- b) Tampilan luaran program sudah benar dan sesuai dengan program yang disusun. Data yang di tampilkan telah sesuai dengan permintaan di soal yaitu menampilkan Kelas Manusia tanpa nama.

[No.1] Kesimpulan

1) Analisa

Pada kode program itu saya menggunakan bentuk kelas public Manusia karena agar dapat diakses oleh kelas lain. Perbaiki kode dengan menambahkan public static void main(String[] args) { karena jika tidak ditambahkan maka tidak akan menghasilkan luaran. Kelas adalah entitas keadaan dan perilaku dari objek, kelas yaitu sekelompok objek yang memiliki sifat umum dengan kumpulan atribut variabel, fungsi dan method dari sebuah objek setelah kompilasi tersimpan pada file *.class(bytecode dari kelas).

Latihan 2

[No. 2] Identifikasi Masalah:

- 1) Susun kembali kode di contoh 2 dengan menambahkan data ciri-ciri Anda didalam variabel constructor

Jawaban:

Saya menyusun kembali kode di contoh 2 dengan menambahkan nama, warna rambut, jenis kelamin dan warna kulit.

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

Jawaban:

Analisa sifat (atribut) yang diturunkan yaitu nama, jenis kelamin, alamat, tinggi badan dan lain-lain.

Constructor yang dibuat yaitu (String nama, String jenis kelamin, String alamat, int tinggi badan).

Perilaku positif (behavior) yang diturunkan yaitu suka membaca, menyanyi, memasak, berlari, dan menonton.

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan data ciri-ciri saya seperti nama, warna rambut, jenis kelamin dan warna kulit ke dalam variabel constructor dengan objek kelas Ortu satu.
- 2) Alasan solusi ini karena agar kode program dapat berjalan dan menampilkan data ciri-ciri yang ada dalam variabel constructor.
- 3) Perbaiki kode program dengan cara menambahkan data ciri-ciri saya kedalam variabel constructor seperti berikut:

```
public Ortu(String nama, String rambut, String jenisKelamin, String
warnaKulit) {
    //nama, rambut, jenis kelamin dan warna kulit adalah variabel
    constructor
    System.out.println(" Nama saya : " + nama +
        "\n Warna Rambut : " + rambut +
        "\n Jenis Kelamin : " + jenisKelamin +
        "\n Warna Kulit : " + warnaKulit);
}

// Metode utama yang digunakan sebagai titik masuk program
public static void main (String[] args) {
```

```

        // Membuat objek Ortu dengan data yang diberikan
        // Objek ini akan menginisialisasi parameter nama, rambut,
        jenisKelamin, dan warnaKulit
        Ortu satu = new Ortu("Sevi Rina Pertiwi", "Hitam", "Perempuan",
        "Cokelat");
    }
}

```

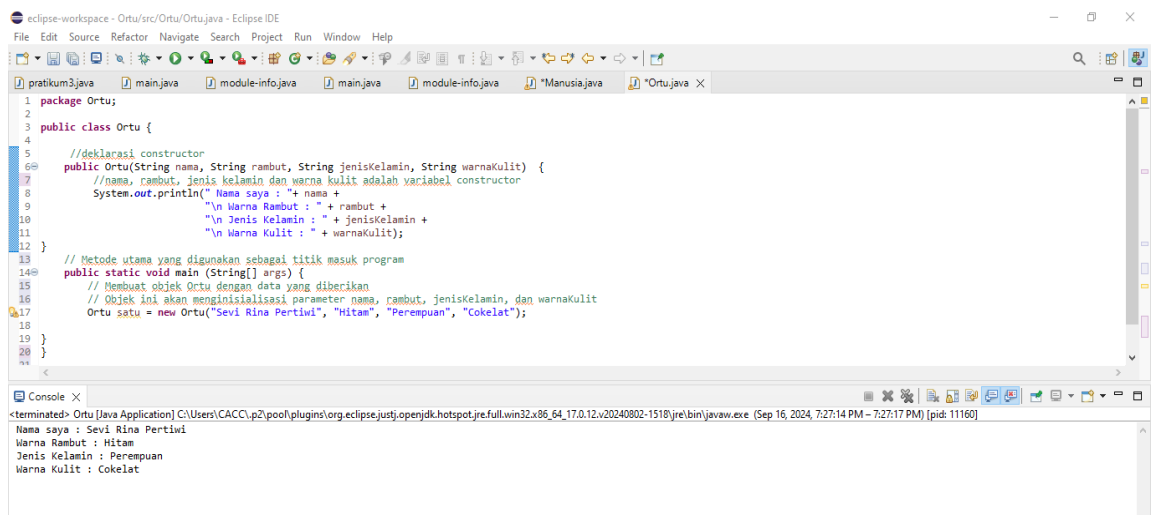
[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma

- Membuat nama file project java
- Membuat nama class dan main java
- Deklarasikan method utama pada program
- Membuat public Ortu
- Mengisi ciri-ciri ke variabel constructor dan mengisi variabelnya
- Menampilkan hasil luarannya.

2) Kode program dan luaran

a) Kode program, komentar dan hasil luaran



```

package Ortu;

public class Ortu {

    //deklarasi constructor
    public Ortu(String nama, String rambut, String jenisKelamin, String warnaKulit) {
        //nama, rambut, jenis kelamin dan warna kulit adalah variabel constructor
        System.out.println(" Nama saya : " + nama +
        "\n Warna Rambut : " + rambut +
        "\n Jenis Kelamin : " + jenisKelamin +
        "\n Warna Kulit : " + warnaKulit);
    }

    // Metode utama yang digunakan sebagai titik masuk program
    public static void main (String[] args) {
        // Membuat objek Ortu dengan data yang diberikan
        // Objek ini akan menginisialisasi parameter nama, rambut, jenisKelamin, dan warnaKulit
        Ortu satu = new Ortu("Sevi Rina Pertiwi", "Hitam", "Perempuan", "Cokelat");
    }
}

```

Console Output:

```

Nama saya : Sevi Rina Pertiwi
Warna Rambut : Hitam
Jenis Kelamin : Perempuan
Warna Kulit : Cokelat

```

- Tampilan luaran program sudah benar dan sesuai dengan program yang disusun. Data yang di tampilkan telah sesuai dengan menampilkan nama, warna rambut, jenis kelamin dan warna kulit dari ciri-ciri yang di masukan ke dalam variabel constructor.

[No.2] Kesimpulan

1) Analisa

Pada program di atas saya menggunakan public class Ortu karena mengikuti dari contoh 2. Perbaikan kode program saya lakukan dengan menambahkan data ciri-ciri seperti nama, warna rambut, jenis kelamin, dan warna kulit ke dalam variabel constructor dengan objek kelas Ortu satu. Objek adalah bentuk turunan dari kelas, sehingga otomatis memiliki atribut dan method dari kelas. Deklarasi objek : NamaKelas NamaObjek = new NamaKelas(parameter);.

Latihan 3

[No. 3] Identifikasi Masalah:

- 1) Analisa perbedaan deklarasi constructor, method, dan method utama!

Jawaban:

Perbedaan constructor, method, dan method utama yaitu kalo constructor tidak memiliki nilai return atau tidak memiliki tipe pengembalian dan namanya sama dengan class, sedangkan method memiliki nilai return dan method mengembalikan hasil (output) dengan kata kunci return dan namanya berbeda dengan nama kelas. Lalu method utama berbeda karena method utama adalah method yang dijalankan pertama kali di program java untuk memulai eksekusi program, selalu dinamai main dan memiliki signature khusus yaitu public static void main(String[] args).

- 2) Tentukan kapan Anda perlu menggunakan constructor dan method?

Jawaban:

- Constructor digunakan ketika akan menginisialisasi objek baru dengan parameter. Jika kelasnya memiliki parameter maka harus diisi saat membuat objek, constructor akan menyertakan parameter tersebut.
- Method digunakan saat membuat program dengan aksi atau operasi pada objek yang telah dibuat, seperti menambahkan method yang berkaitan dengan objek dan menjalankan fungsi yang berkaitan dengan objek tersebut.

- 3) Uraikan perbedaan berikut:

- a) Constructor overloading dan overriding
- b) Method overloading dan method overriding
- c) Method yang mengembalikan nilai dan method yang tidak mengembalikan nilai

Jawaban:

- a) Perbedaan constructor overloading dengan overriding yaitu constructor overloading memiliki banyak constructor dalam satu kelas dengan nama constructor sama namun beda tipe dan jumlah parameter , sedangkan constructor overriding hanya digunakan untuk inisialisasi objek dan tidak diwarisi.
- b) Perbedaan method overloading dan method overriding yaitu method overloading memiliki method dengan nama yang sama dalam satu kelas tetapi dengan isi parameter yang berbeda , sedangkan method overriding subclass menyediakan implementasi baru untuk method yang sudah ada dengan nama dan parameter yang sama.
- c) Perbedaan method yang mengembalikan nilai dan method yang tidak mengembalikan nilai yaitu method yang mengembalikan nilai adalah method yang memberikan hasil dari operasinya kepada pemanggilnya. Tipe pengembalian harus ditentukan dan menggunakan kata kunci return diikuti oleh nilai yang dikembalikan, seperti int, float, atau String. Sedangkan, method yang tidak mengembalikan nilai memiliki tipe pengembalian void dan hanya melakukan aksi tanpa memberikan hasil. Method ini bisa menggunakan return tanpa nilai untuk keluar lebih awal, tetapi tidak mengembalikan hasil.

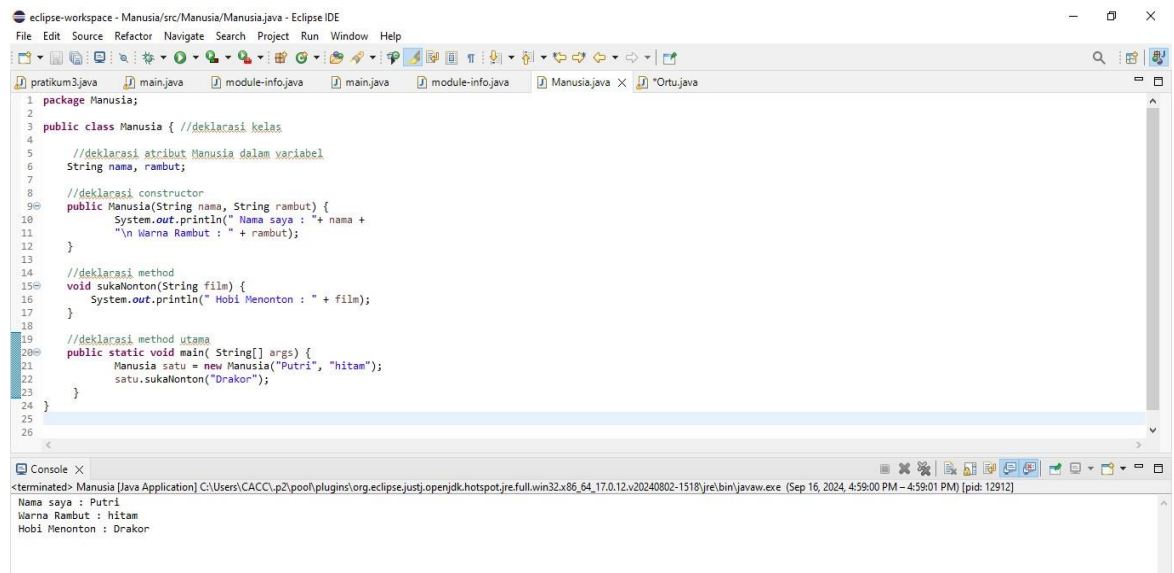
[No.3] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memahami penggunaan constructor, method, dan method utama dalam membuat kode program.
- 2) Alasan solusi ini karena agar program dapat berjalan dan tidak eror.

- 3) Perbaiki kode program dengan cara membuat kode program dengan public class Manusia yang dibuat constructor, method dan method utama.

[No.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Membuat nama file project java
 - (b) Membuat nama class dan main java
 - (c) Mendeklarasikan constructor
 - (d) Mendeklarasikan method
 - (e) Mendeklarasikan method utama dan mengisinya
 - (f) Menampilkan hasil luaran dari constructor, method dan method utama.
- 2) Kode program dan luaran
 - a) Kode program, komentar dan hasil luaran



```
1 package Manusia;
2
3 public class Manusia { //deklarasi kelas
4
5     //deklarasi atribut Manusia dalam variabel
6     String nama, rambut;
7
8     //deklarasi constructor
9     public Manusia(String nama, String rambut) {
10         System.out.println(" Nama saya : " + nama +
11             "\n Warna Rambut : " + rambut);
12     }
13
14     //deklarasi method
15     void sukaMonton(String film) {
16         System.out.println(" Hobi Menonton : " + film);
17     }
18
19     //deklarasi method utama
20     public static void main( String[] args) {
21         Manusia satu = new Manusia("Putri", "hitam");
22         satu.sukaMonton("Drakor");
23     }
24 }
25
26
```

Console X
<terminated> Manusia [Java Application] C:\Users\CACCI\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\javaw.exe (Sep 16, 2024, 4:59:00 PM - 4:59:01 PM) [pid: 12912]
Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor

- b) Tampilan luaran program sudah benar dan sesuai dengan program yang disusun. Data yang ditampilkan telah sesuai dengan menampilkan Nama saya : Putri, Warna Rambut : hitam dan Hobi Menonton : Drakor. Hasil luaran sesuai dengan konstruktor, method dan method utama yang dibuat.

[No.3] Kesimpulan

- 1) Analisa
Pada program itu saya menggunakan bentuk kelas public Manusia karena agar sesuai dengan isi method dan method utama. Program tersebut berisi constructor, method dan method utama karena pada java terdapat dua jenis constructor yaitu constructor overloading memiliki banyak constructor dalam satu kelas dengan nama constructor sama namun beda tipe dan jumlah parameter , sedangkan constructor overriding hanya digunakan untuk inisialisasi objek dan tidak diwarisi. Terdapat 2 jenis method yaitu method overloading memiliki method dengan nama yang sama dalam satu kelas tetapi dengan isi parameter yang berbeda , sedangkan method overriding subclass

menyediakan implementasi baru untuk method yang sudah ada dengan nama dan parameter yang sama. Dan ada juga method yang mengembalikan nilai dan method yang tidak mengembalikan nilai yaitu method yang mengembalikan nilai adalah method yang memberikan hasil dari operasinya kepada pemanggilnya. Tipe pengembalian harus ditentukan dan menggunakan kata kunci return diikuti oleh nilai yang dikembalikan, seperti int, float, atau String. Sedangkan, method yang tidak mengembalikan nilai memiliki tipe pengembalian void dan hanya melakukan aksi tanpa memberikan hasil. Method ini bisa menggunakan return tanpa nilai untuk keluar lebih awal, tetapi tidak mengembalikan hasil.

Latihan 4

[No. 4] Identifikasi Masalah:

- 1) Bandingkan method yang memiliki class Anak extends Ortu dengan method di class Ortu!

Jawaban:

Perbandingan method yang dimiliki class Anak extends Ortu dengan method di class Ortu yaitu di class Anak mewarisi method `sukaMenonton(String a)` dan `sukaMembaca(String a)` dari class Ortu, tetapi di class Anak juga menambahkan overload method `sukaMenonton(int a, String b)`. Dan di class Anak memiliki method `sukaMenonton(String a)` dan `sukaMembaca(String a)` yang sama seperti class Ortu, tetapi juga menyediakan method tambahan dengan parameter yang berbeda.

- 2) Ubahlah contoh 4 dengan menambahkan objek anak dengan method yang berbeda!

Jawaban:

Pada objek anak saya menambahkan method yang berbeda dengan yang ada di objek ortu yaitu method yang ditambahkan adalah `sukaBermainGame(String a)` dan `sukaMemasak(String a)`.

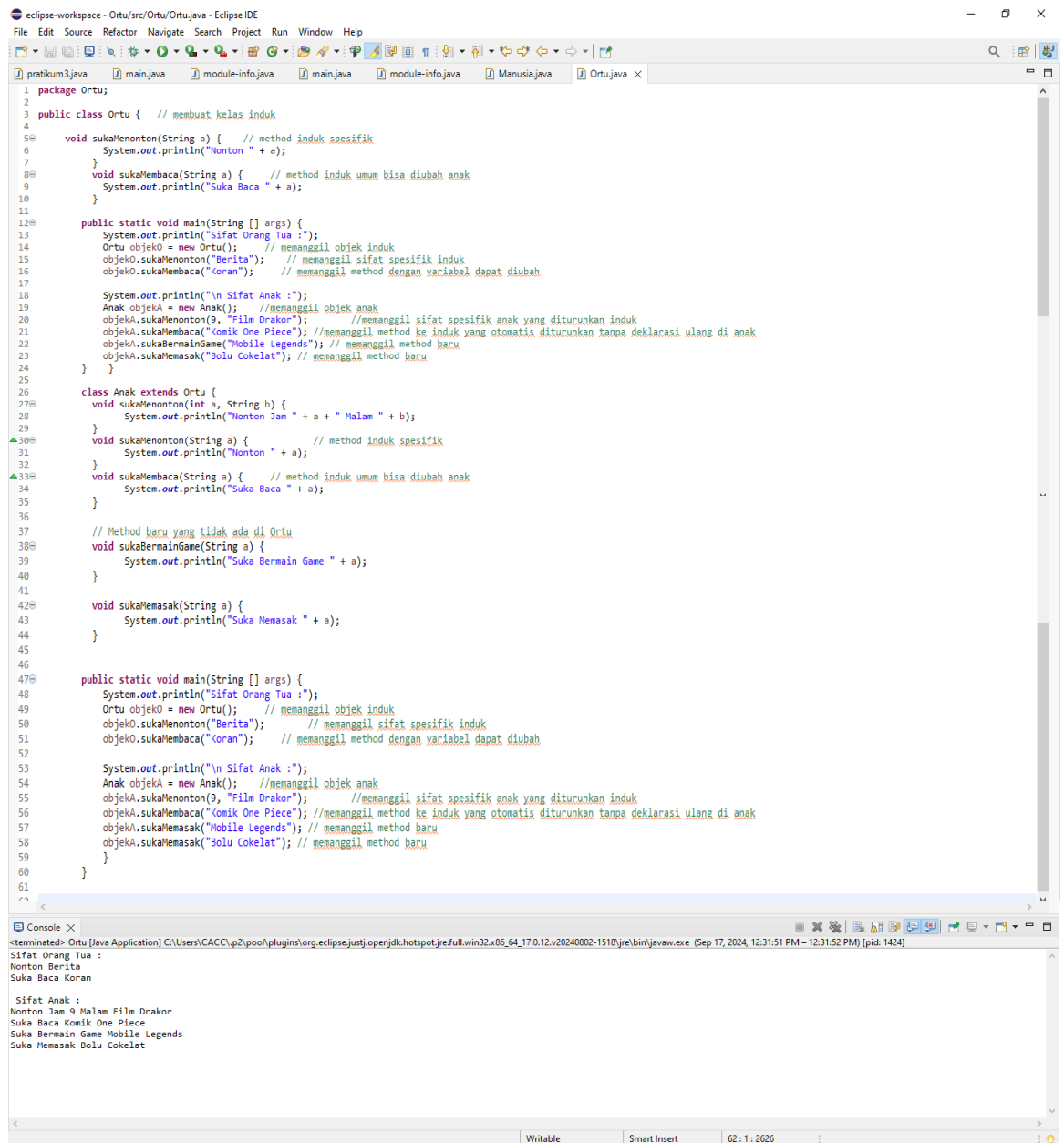
[No.4] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan method baru di objek anak yang berbeda dengan method yang ada di objek orang tua.
- 2) Alasan solusi ini karena dapat menambahkan method yang berbeda di objek anak.
- 3) Perbaiki kode program dengan cara menambahkan method baru di objek anak yang tidak ada di objek orang tua. Saya menambahkan method `sukaBermainGame(String a)` dan `sukaMemasak(String a)`.

[No.4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Membuat nama file project java
 - (b) Membuat nama class dan main java
 - (c) Membuat method di class Ortu dan mengisi variabelnya
 - (d) Membuat method di class Anak yang sama dengan method kelas Ortu
 - (e) Menambahkan method baru di class Anak yang tidak sama dengan class Ortu
 - (f) Memanggil dan menampilkan objek dan methodnya

- 2) Kode program dan luaran
- a) Kode program, komentar dan hasil luaran



```
1 package Ortu;
2
3 public class Ortu { // membuat kelas induk
4
5     void sukaMenonton(String a) { // method induk spesifik
6         System.out.println("Nonton " + a);
7     }
8     void sukaMembaca(String a) { // method induk umum bisa diubah anak
9         System.out.println("Suka Baca " + a);
10    }
11
12    public static void main(String [] args) {
13        System.out.println("Sifat Orang Tua :");
14        Ortu objekO = new Ortu(); // memanggil objek induk
15        objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
16        objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
17
18        System.out.println("\n Sifat Anak :");
19        Anak objekA = new Anak(); //memanggil objek anak
20        objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
21        objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
22        objekA.sukaBermainGame("Mobile Legends"); // memanggil method baru
23        objekA.sukaMemasak("Bolu Cokelat"); // memanggil method baru
24    }
25
26    class Anak extends Ortu {
27        void sukaMenonton(int a, String b) {
28            System.out.println("Nonton Jam " + a + " Malam " + b);
29        }
30        void sukaMenonton(String a) { // method induk spesifik
31            System.out.println("Nonton " + a);
32        }
33        void sukaMembaca(String a) { // method induk umum bisa diubah anak
34            System.out.println("Suka Baca " + a);
35        }
36
37        // Method baru yang tidak ada di Ortu
38        void sukaBermainGame(String a) {
39            System.out.println("Suka Bermain Game " + a);
40        }
41
42        void sukaMemasak(String a) {
43            System.out.println("Suka Memasak " + a);
44        }
45
46
47        public static void main(String [] args) {
48            System.out.println("Sifat Orang Tua :");
49            Ortu objekO = new Ortu(); // memanggil objek induk
50            objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
51            objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
52
53            System.out.println("\n Sifat Anak :");
54            Anak objekA = new Anak(); //memanggil objek anak
55            objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturunkan induk
56            objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan tanpa deklarasi ulang di anak
57            objekA.sukaMemasak("Mobile Legends"); // memanggil method baru
58            objekA.sukaMemasak("Bolu Cokelat"); // memanggil method baru
59        }
60    }
61 }
```

Console

```
<terminated> Ortu [Java Application] C:\Users\CACC\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.12.v20240802-1518\jre\bin\javaw.exe (Sep 17, 2024, 12:31:51 PM - 12:31:52 PM) [pid: 1424]
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton 9 Malam Film Drakor
Suka Baca Komik One Piece
Suka Bermain Game Mobile Legends
Suka Memasak Bolu Cokelat
```

- b) Tampilan luaran program sudah benar dan sesuai dengan program yang disusun. Data yang ditampilkan telah sesuai dengan menambahkan method baru yaitu Suka Bermain Game Mobile Legends dan Suka Memasak Bolu Cokelat.

[No.4] Kesimpulan

1) Analisa

Pada program itu saya menggunakan bentuk class Ortu dan class Anak karena membuat dua objek yang berbeda dan methodnya. Perbaikan program saya lakukan dengan cara menambahkan method yang baru dan berbeda dengan di class Ortu. Pada java kita bisa menambahkan method yang baru dan berbeda dengan class induk tanpa harus merubah method yang ada di class induk. Dalam program tersebut saya membandingkan method yang dimiliki class Anak extends Ortu dengan method di class Ortu yaitu di class Anak mewarisi method `sukaMenonton(String a)` dan `suka Membaca(String a)` dari class Ortu, tetapi di class Anak juga menambahkan overload method `sukaMenonton(int a, String b)`. Dan di class Anak memiliki method `sukaMenonton(String a)` dan `sukaMembaca(String a)` yang sama seperti class Ortu, tetapi juga menyediakan method tambahan dengan parameter yang berbeda.

Refleksi

Pengalaman belajar saya pada minggu ini cukup lumayan banyak, untuk memahami kelas, objek dan method. Pengetahuan baru yang di dapat cara membuat kelas, objek dan method, membuat kelas induk dan anak dan menambahkan method yang baru dan berbeda dengan kelas induk, menganalisis perbedaan constructor overloading dan overriding, perbedaan method overloading dan overriding, dan perbedaan method yang menampilkan nilai dan yang tidak menampilkan nilai. Tantangan yang dihadapi harus bisa menganalisis penyebab program eror apa saja dan overflow. Dari 4 latihan di atas banyak ilmu baru yang saya ketahui dari membuat kelas, objek dan method yang benar.