

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Risneliya Elisa G1F024005	Kelas, Objek, Method	18/09/2024

[Nomor Soal] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

```
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");
    } }
```

Latihan 1:

- 1.1. Perbaiki pesan kesalahan Contoh 1!

Nama kelas dan nama konstruktor harus sama pada jika nama “manusia” maka konstruktornya juga “manusia”, penambahan paramenter (String rambut)

```
public class Ortu {
    //deklarasi constructor (variabel constructor)
    public ortu {
        //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!

Tidak adanya paramenter (String nama, String rambut), nama dan konsruktor berbeda (Ortu) dan (ortu)

```
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");
    } }

```

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)

```

Latihan 3:

3.1. Evaluasi penyebab kesalahan dan perbaiki kode tersebut!

- **void sukaNonton(String film, int tahun):** Menambahkan informasi tahun pada film yang ditonton.
- **int sukaNonton(int episode, int durasi):** Menghitung total jam menonton berdasarkan jumlah episode dan durasi per episode, dan mengembalikan hasilnya.
- **String sukaNonton(int episode, int durasi, boolean dalamJam):** Menghitung total waktu nonton dalam format jam dan menit jika dalamJam true, dan mengembalikan hasil dalam format string.
- **void sukaNonton(String film):** Menampilkan hobi menonton berdasarkan nama film tanpa mengembalikan nilai.

Latihan 4

```

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
    } }

```

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!

- Hapus metode `sukamenonton` String nama di kelas `Anak` jika tidak mengubah implementasi dari kelas `Ortu`. Jika Anda ingin menambahkan variasi metode, cukup tambahkan metode yang berbeda (misalnya `sukaMenonton(int a, String b)`).
- Jika `sukaMembaca` di kelas `Anak` tidak diubah dari versi di kelas `Ortu`, tidak perlu mendeklarasikannya ulang. `Anak` akan mewarisi implementasi dari `Ortu` secara otomatis.
- Jika Anda memang perlu mengubah implementasi metode dari `Ortu`, pastikan untuk menyertakan implementasi baru yang sesuai dengan kebutuhan di kelas `Anak`. Ini akan memanfaatkan konsep overriding dengan tepat.

- 2) 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

- Uraikan rancangan solusi yang diusulkan.
 1. Ubah nama konstruktor sehingga sama dengan nama kelas, tambah parameter String rambut setelah String nama
 2. Tambah String nama dan String rambut, samakan nama kelas dan struktur
 3. Menambahkan informasi tahun pada film yang ditonton, Menghitung total jam menonton berdasarkan jumlah episode dan durasi per episode, dan mengembalikan hasilnya, Menghitung total waktu nonton dalam format jam dan menit jika `dalamJam` true, dan mengembalikan hasil dalam format string, Menampilkan hobi menonton berdasarkan nama film tanpa mengembalikan nilai.

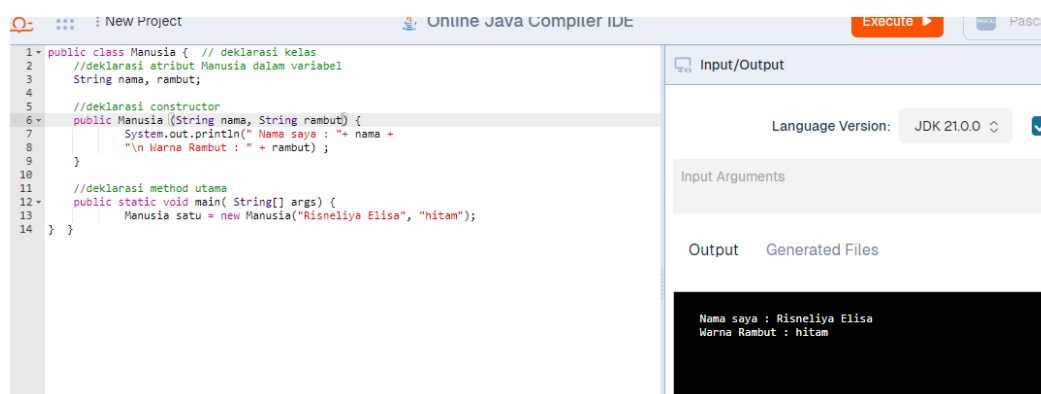
4. Hapus metode **sukamenonton** String nama di kelas Anak jika tidak mengubah implementasi dari kelas Ortu, Jika **sukaMembaca** di kelas Anak tidak diubah dari versi di kelas Ortu, tidak perlu mendeklarasikannya ulang. Jika Anda memang perlu mengubah implementasi metode dari Ortu, pastikan untuk menyertakan implementasi baru yang sesuai dengan kebutuhan di kelas Anak.

- Analisis solusi, kaitkan dengan permasalahan.
Dengan masalah nomer 1 sampai empat Solusi yang diberikan sudah jelas sehingga program dapat berjalan dengan seharusnya.

[Nomor Soal] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 1. Buka e learning Unib
 2. Buka portal mata kuliah computer dan pemrograman
 3. Pilih kelas diambil
 4. Lihat materi sesuai dengan kelasnya
 5. Tempel kode ke online java compiler
 6. Analisis perintah yang diberikan
 7. Lakukan perintah dan selesai
 - 2) Tuliskan kode program dan luaran
 1. Beri komentar pada kode
 1. 1.1 adalah kode untuk memperbaiki kesalahan pada kode sebelumnya
 - 1.2 adalah kode untuk perintah mengganti paramenter menjadi data diri saya
 2. 2.2 adalah kode untuk perintah Analisa sifat keturunan
 - 2.3 adalah kode untuk perintah untuk merancang program untuk sifat dan konstruktor
 3. 3.1 adalah kode untuk evaluasi kesalahan pada kode pertama
 - 3.2 adalah kode untuk mengembangkan method
 4. 4.1 adalah kode untuk evaluasi method
 - 4.2 adalah kode untuk Menyusun Kembali kode yang lebih efisien
 2. Uraikan luaran yang dihasilkan
Luaran yang dihasilkan dari beberapa program dibawah sudah benar
 3. Screenshot/ Capture potongan kode dan hasil luaran
- Latihan 1:

1.1 perbaiki kesalahan



The screenshot shows an online Java IDE with a code editor on the left and an 'Input/Output' panel on the right. The code defines a class 'Manusia' with attributes 'nama' and 'rambut', a constructor, and a static 'main' method. The 'main' method creates an instance of 'Manusia' with the name 'Risnelliya Elisa' and hair color 'hitam', then prints them out. The 'Output' panel shows the execution result: 'Nama saya : Risnelliya Elisa' and 'Warna Rambut : hitam'.

```
1 public class Manusia { // deklarasi kelas
2     //deklarasi atribut Manusia dalam variabel
3     String nama, rambut;
4
5     //deklarasi constructor
6     public Manusia (String nama, String rambut) {
7         System.out.println(" Nama saya : "+ nama +
8             "\n Warna Rambut : " + rambut);
9     }
10
11     //deklarasi method utama
12     public static void main( String[] args) {
13         Manusia satu = new Manusia("Risnelliya Elisa", "hitam");
14     } }
```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Output Generated Files

Nama saya : Risnelliya Elisa
Warna Rambut : hitam

```
1- public class Manusia { // deklarasi kelas
2- //deklarasi atribut Manusia dalam variabel
3- String nama, rambut, tinggi, berat;
4-
5- //deklarasi constructor
6- public Manusia (String nama, String rambut, String tinggi, String berat) {
7-     System.out.println(" Nama saya : "+ nama +
8-     "\n Warna Rambut : " + rambut +
9-     "\n tinggi : " + tinggi +
10-     "\n berat : " + berat );
11- }
12-
13- //deklarasi method utama
14- public static void main( String[] args) {
15-     Manusia satu = new Manusia("Risneliya Elisa", "hitam", "153", "45");
16- }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```
Nama saya : Risneliya Elisa
Warna Rambut : hitam
tinggi : 153
berat : 45
```

Compiled and executed in 1.375 sec(s)

1.2 Cermati contoh 1. susun kode menggunakan constructor dengan parameter data pribadi anda!

Latihan 2:

2.2 Apabila nanti Anda akan memiliki keturunan, analisa sifat (atribut) dan constructor sebagai Ortu apa yang akan diturunkan (gunakan data karakter pribadi anda) ?

```
1- public class Ortu {
2- //deklarasi constructor (variabel constructor)
3- public Ortu(String nama, String rambut, String mata, String hobi, String bakat) {
4- //nama dan rambut adalah variabel constructor
5- System.out.println(" Nama saya : "+ nama +
6- "\n Warna Rambut : " + rambut + "\n Warna mata : " + mata + "\n Hobi : " + hobi +
7- "\n Bakat: " + bakat);
8- }
9- public static void main( String[] args) {
10- Ortu satu = new Ortu("Risneliya elisa", "hitam", "hitam", "memasak", "memasak");
11- }
```

Online Java Compiler IDE

Execute

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```
Nama saya : Risneliya elisa
Warna Rambut : hitam
Warna mata : hitam
Hobi : memasak
Bakat: memasak
```

Compiled and executed in 1.793 sec(s)

```
1- public class Ortu {
2- // Deklarasi variabel instansi
3- private String nama;
4- private String rambut;
5- private String mata;
6- private String hobi;
7- private String bakat;
8-
9- // Constructor default (tanpa parameter)
10- public Ortu() {
11-     nama = "Risneliya Elisa";
12-     rambut = "Hitam";
13-     mata = "Hitam";
14-     hobi = "Memasak";
15-     bakat = "Memasak";
16-     printInfo();
17- }
18-
19- // Constructor dengan beberapa parameter
20- public Ortu(String nama, String rambut) {
21-     nama = "Risneliya Elisa";
22-     rambut = "Hitam";
23-     mata = "Hitam";
24-     hobi = "Memasak";
25-     bakat = "Memasak";
26-     printInfo();
27- }
28-
29- // Constructor dengan lebih banyak parameter
30- public Ortu(String nama, String rambut, String mata) {
31-     nama = "Risneliya Elisa";
32-     rambut = "Hitam";
33-     mata = "Hitam";
34-     hobi = "Memasak";
35-     bakat = "Memasak";
36-     printInfo();
37- }
```

Online Java Compiler IDE

Execute

Language Version: Rust

Input/Output

Output Generated Files

```
Nama saya : Risneliya Elisa
Warna Rambut : Hitam
Warna Mata : Hitam
Hobi : Memasak
Bakat : Memasak
Nama saya : Risneliya Elisa
Warna Rambut : Hitam
Warna Mata : Hitam
Hobi : Memasak
Bakat : Memasak
Nama saya : Risneliya Elisa
Warna Rambut : Hitam
Warna Mata : Hitam
Hobi : Memasak
Bakat : Memasak
Nama saya : Risneliya Elisa
Warna Rambut : Hitam
Warna Mata : Hitam
Hobi : Memasak
Bakat : Memasak
```

Compiled and executed in 1.964 sec(s)

2.3 Rancanglah kode program untuk sifat (atribut) dan constructor overloaded dari Latihan 2.2!

Latihan 3:

3.1 evaluasi kesalahan kode tersebut

```

1 public class Manusia {
2     // Deklarasi variabel instansi
3     private String nama;
4     private String rambut;
5
6     // Constructor
7     public Manusia(String nama, String rambut) {
8         nama = nama;
9         rambut = rambut;
10        System.out.println("Nama saya : " + nama +
11                               "\nWarna Rambut : " + rambut);
12    }
13
14    // Method untuk menampilkan hobi menonton
15    void sukaNonton(String film) {
16        System.out.println("Hobi Menonton : " + film);
17    }
18
19    // Method untuk menghitung total jam menonton
20    int sukaNonton(int episode, int durasi) {
21        return episode * durasi;
22    }
23
24    // Method utama
25    public static void main(String[] args) {
26        // Membuat instance dari Manusia
27        Manusia satu = new Manusia("Putri", "hitam");
28
29        // Memanggil metode sukaNonton dengan parameter string
30        satu.sukaNonton("Drakor");
31
32        // Memanggil metode sukaNonton dengan parameter int
33        int jumlahJam = satu.sukaNonton(2, 2);
34
35        // Menampilkan hasil
36    }
37 }

```

Output: Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor
Jam nonton = 4 jam

```

5
6 // Constructor
7 public Manusia(String nama, String rambut) {
8     nama = nama;
9     rambut = rambut;
10    System.out.println("Nama saya : " + nama +
11                           "\nWarna Rambut : " + rambut);
12}
13
14 // Method untuk menampilkan hobi menonton
15 void sukaNonton(String film) {
16     System.out.println("Hobi Menonton : " + film);
17 }
18
19 // Method untuk menghitung total jam menonton
20 int sukaNonton(int episode, int durasi) {
21     return episode * durasi;
22 }
23
24 // Method utama
25 public static void main(String[] args) {
26     // Membuat instance dari Manusia
27     Manusia satu = new Manusia("Putri", "hitam");
28
29     // Memanggil metode sukaNonton dengan parameter string
30     satu.sukaNonton("Drakor");
31
32     // Memanggil metode sukaNonton dengan parameter int
33     int jumlahJam = satu.sukaNonton(2, 2);
34
35     // Menampilkan hasil
36     System.out.println("Jam nonton = " + jumlahJam + " jam");
37 }
38
39

```

Output: Nama saya : Putri
Warna Rambut : hitam
Hobi Menonton : Drakor
Jam nonton = 4 jam

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:

- method overloading,
- method dengan return value
- method tanpa return value

```

27 }
28
29 // Method untuk menghitung total waktu nonton dalam jam dan menit (method overloading)
30 String sukaNonton(int episode, int durasi, boolean dalamJam) {
31     int totalMenit = episode * durasi;
32     if (dalamJam) {
33         int jam = totalMenit / 60;
34         int menit = totalMenit % 60;
35         return jam + " jam " + menit + " menit";
36     } else {
37         return totalMenit + " menit";
38     }
39 }
40
41 // Method utama
42 public static void main(String[] args) {
43     // Membuat instance dari Manusia
44     Manusia satu = new Manusia("Risneliya Elisa", "hitam");
45
46     // Memanggil metode sukaNonton dengan parameter string
47     satu.sukaNonton("Drakor");
48
49     // Memanggil metode sukaNonton dengan parameter int
50     int jumlahJam = satu.sukaNonton(2, 2);
51     System.out.println("Jam nonton = " + jumlahJam + " jam");
52
53     // Memanggil metode sukaNonton dengan parameter string dan int
54     satu.sukaNonton("Drakor", 2024);
55
56     // Memanggil metode sukaNonton dengan parameter int, int, dan boolean
57     String waktuNonton = satu.sukaNonton(2, 120, true);
58     System.out.println("Total waktu nonton = " + waktuNonton);
59 }
60
61

```

Output: Nama saya : Risneliya Elisa
Warna Rambut : hitam
Hobi Menonton : Drakor
Tahun: 2024
Total waktu nonton = 4 jam 0 menit

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!

The screenshot shows an IDE with a Java program. The code defines a base class `Ortu` with methods `sukaMembaca` and `sukaMenonton`, and a subclass `Anak` that extends `Ortu` and overrides `sukaMenonton`. The `main` method creates instances of both classes and calls these methods. The output window shows the results of the execution, indicating that the program compiled and executed in 1.802 seconds.

```
4 }
5 void sukaMembaca(String a) { // method induk umum bisa diubah anak
6     System.out.println("Suka Baca " + a);
7 }
8
9 public static void main(String [] args) {
10     System.out.println("Sifat Orang Tua :");
11     Ortu objekO = new Ortu(); // memanggil objek induk
12     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
13     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
14
15     System.out.println("\n Sifat Anak :");
16     Anak objekA = new Anak(); //memanggil objek anak
17     objekA.sukaMenonton(9, "Film Drakor"); //memanggil sifat spesifik anak yang diturun
18     objekA.sukaMembaca("Komik One Piece"); //memanggil method ke induk yang otomatis diturunkan
19 }
20
21 class Anak extends Ortu {
22     void sukaMenonton(int a, String b) {
23         System.out.println("Nonton Jam " + a + " Malam " + b);
24     }
25     void sukaMembaca(String a) { // method induk spesifik
26         System.out.println("Nonton " + a);
27     }
28 }
29 void sukaMembaca(String a) { // method induk umum bisa diubah anak
30     System.out.println("Suka Baca " + a);
31 }
32 public static void main(String [] args) {
33     System.out.println("Sifat Orang Tua :");
34     Ortu objekO = new Ortu(); // memanggil objek induk
35     objekO.sukaMenonton("Berita"); // memanggil sifat spesifik induk
36     objekO.sukaMembaca("Koran"); // memanggil method dengan variabel dapat diubah
37
38     System.out.println("\n Sifat Anak :");
39     Anak objekA = new Anak(); //memanggil objek anak
```

Output:

```
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
```

Compiled and executed in 1.802 sec(s)

4.2. Setelah dirunning di JDoodle, catat waktu eksekusinya.

Susun kembali kode program yang dapat mengefisienkan waktu eksekusi!

The screenshot shows the same IDE with a revised Java program. The code is more concise, using `@Override` and removing redundant comments. The output window shows the results of the execution, indicating that the program compiled and executed in 1.277 seconds, which is faster than the previous version.

```
1 public class Ortu {
2     // Method untuk menampilkan hobi menonton
3     void sukaMenonton(String a) {
4         System.out.println("Nonton " + a);
5     }
6
7     // Method untuk menampilkan hobi membaca
8     void sukaMembaca(String a) {
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(); // Membuat objek induk
15        objekO.sukaMenonton("Berita"); // Memanggil method induk
16        objekO.sukaMembaca("Koran"); // Memanggil method induk
17
18        System.out.println("\nSifat Anak :");
19        Anak objekA = new Anak(); // Membuat objek anak
20        objekA.sukaMenonton(9, "Film Drakor"); // Memanggil method overloaded di anak
21        objekA.sukaMembaca("Komik One Piece"); // Memanggil method overridden di anak
22    }
23 }
24
25 class Anak extends Ortu {
26     // Method overloaded untuk menampilkan hobi menonton dengan jam
27     void sukaMenonton(int a, String b) {
28         System.out.println("Nonton Jam " + a + " Malam " + b);
29     }
30
31     // Method overridden dari Ortu
32     @Override
33     void sukaMembaca(String a) {
34         System.out.println("Suka Baca " + a);
35     }
36 }
```

Output:

```
Sifat Orang Tua :
Nonton Berita
Suka Baca Koran

Sifat Anak :
Nonton Jam 9 Malam Film Drakor
Suka Baca Komik One Piece
```

Compiled and executed in 1.277 sec(s)

[Nomor Soal] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Pada program kali ini kesalahan yang terjadi mulai dari parameter yang tidak lengkap, kurangnya konstruktor, berbedanya nama konstruktor dan nama kelas.
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Pada program diatas saya menggunakan kelas public karena saya ingin semua kelas dapat mengakses program ini, dan permasalahan yang terjadi dapat diselesaikan sesuai dengan aturan pada setiap kesalahan.

Nama kelas dan nama konstruktor harus sama, parameter dan konstruktor harus sesuai Ketika memasukan 3 parameter maka konstruktornya juga 3

2) Evaluasi

- Apa konsekuensi dari skenario pemrograman ini?
- Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)

3) Kreasi

- 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:
Putri G1A000001	Tipe Data	26 Agustus 2022

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel
Contoh:
Tuliskan kembali soal:
Pada soal masih ada pesan kesalahan _____
Atau
Diketahui dari soal : variabel _____

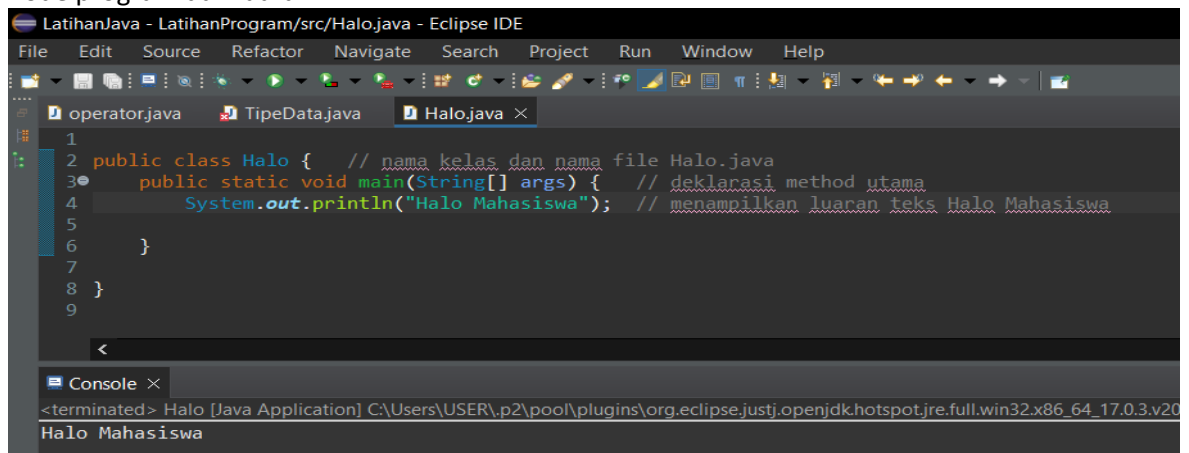
[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara _____
- 2) Alasan solusi ini karena _____
- 3) Perbaiki kode program dengan cara _____

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
Algoritma adalah langkah-langkah penyelesaian masalah.
Misalkan algoritma memasak mi instan:
(a) Masak air
(b) Buka bungkus
(c) Masukkan mie
(d) Masukkan bumbu
(e) Hasilnya mie matang, taruh di piring
(f) Mie siap disantap.

- 2) Kode program dan luaran



```
LatihanJava - LatihanProgram/src/Halo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
operator.java TipeData.java Halo.java x
1
2 public class Halo { // nama kelas dan nama file Halo.java
3     public static void main(String[] args) { // deklarasi method utama
4         System.out.println("Halo Mahasiswa"); // menampilkan luaran teks Halo Mahasiswa
5     }
6 }
7
8 }
9

Console x
<terminated> Halo [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.3.v20
Halo Mahasiswa
```

- a) Screenshot/ Capture potongan kode dan hasil luaran
Beri komentar pada kode yang di Screenshot
- b) Analisa luaran yang dihasilkan
Contoh:
Luaran sudah sesuai dengan program yang disusun.
Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

(Tuliskan penjelasan dari program yang dibuat, apakah kode dan luaran sudah benar?)

[No.1] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Contoh jawaban Analisa:

Pada program itu saya menggunakan bentuk kelas public karena _____

Perbaiki program dengan menambahkan _____ karena struktur java mengharuskan _____

(penjelasan analisa mengulangi kembali materi yang sudah diberikan)

(penjelasan mengkaitkan dengan materi yang ada)

2) Evaluasi

- Apa konsekuensi/dampak dari kode program yang dibuat?
- Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)

Contoh jawaban Evaluasi:

Pada program itu saya mengkonversi bentuk kelas public karena _____

Setelah dikonversi, saya mengevaluasi bahwa tipe data _____ lebih baik digunakan untuk bentuk data seperti _____

(penjelasan evaluasi mengulangi kembali materi yang sudah diberikan dan mengetahui kekurangan dari materi hasil eksperimen)

(misal tipe data ____ ternyata tidak dapat dipakai untuk _____ karena _____)

3) Kreasi

- Apakah ada pengetahuan baru yang dikembangkan dan konsep baru sebagai usulan solusi?
- Susunlah hubungan antara variabel yang berbeda dengan konsep yang anda ketahui! (jika ada)

Contoh jawaban Kreasi:

Pada program itu saya mengkonversi bentuk kelas public karena _____

Setelah dikonversi, saya mengevaluasi bahwa tipe data _____ lebih baik digunakan untuk bentuk data seperti _____

Saya telah mencoba mengubah menjadi kelas private dan protected, ternyata menghasilkan _____

Berarti kelas private dan protected mempengaruhi _____

(sampaikan temuan Anda yang baru diketahui, eksperimen baru diluar materi yang diberikan)

(penjelasan kreasi mengulangi kembali materi yang sudah diberikan dan

menambahkan pengetahuan baru dari pengalaman dari hasil eksperimen)

Lanjutkan ke soal nomor 2 – 3 – ... – dan seterusnya

Refleksi

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)