

Nama & NPM	Topik:	Tanggal:
Nadya Putri Marsela G1F024036 Karina Hadiyah Ramadona G1F024040	Tugas Kelompok Kelas	18 September 2024
<b>[No.1] Identifikasi Masalah:</b>		
<p>1) Uraikan permasalahan dan variable</p> <p>1. Apabila diketahui kelas induk adalah Mahasiswa dan Kelas anak adalah turunan dari mahasiswa maka:</p> <ul style="list-style-type: none"> <li>• (a) Analisa atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!</li> <li>• (b) Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!</li> <li>• (c) Rekomendasi atribut, method, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!</li> <li>• (d) Desain kode program Java yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!</li> </ul> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p><a href="https://www.youtube.com/watch?v=60IdOc8m8Es%C2%A0">https://www.youtube.com/watch?v=60IdOc8m8Es%C2%A0</a>  <a href="https://www.youtube.com/watch?v=6qULMlcv-eg">https://www.youtube.com/watch?v=6qULMlcv-eg</a></p>		
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Analisa Atribut, Method, dan Constructor</p> <p>1. Kelas Induk: Mahasiswa</p> <p>Atribut: String nama: Untuk menyimpan nama mahasiswa. String nim: Untuk menyimpan nomor induk mahasiswa. String jurusan: Untuk menyimpan jurusan mahasiswa.</p> <p>Method: void tampilInfo(): Untuk menampilkan informasi dasar tentang mahasiswa, seperti nama, NIM, dan jurusan. void belajar(String mataKuliah): Untuk menyatakan mata kuliah yang sedang dipelajari mahasiswa.</p> <p>Constructor Mahasiswa(String nama, String nim, String jurusan): Constructor untuk menginisialisasi atribut nama, nim, dan jurusan saat objek Mahasiswa dibuat.</p> <p>2. Kelas Anak: Misalnya MahasiswaS2</p> <p>Atribut: String topikTesis: Untuk menyimpan topik tesis mahasiswa S2.</p> <p>Method: void tampilInfo(): Override dari method di kelas Mahasiswa untuk menambahkan informasi tentang topik tesis. void menulisTesis(): Untuk menyatakan aktivitas menulis tesis.</p> <p>Constructor: MahasiswaS2(String nama, String nim, String jurusan, String topikTesis): Constructor yang menginisialisasi atribut dari kelas induk dan juga atribut tambahan untuk kelas MahasiswaS2.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi ini menggunakan konsep pewarisan dalam OOP dengan kelas Mahasiswa sebagai kelas induk yang menyediakan atribut dan method dasar seperti nama, nim, jurusan, serta method</p>		

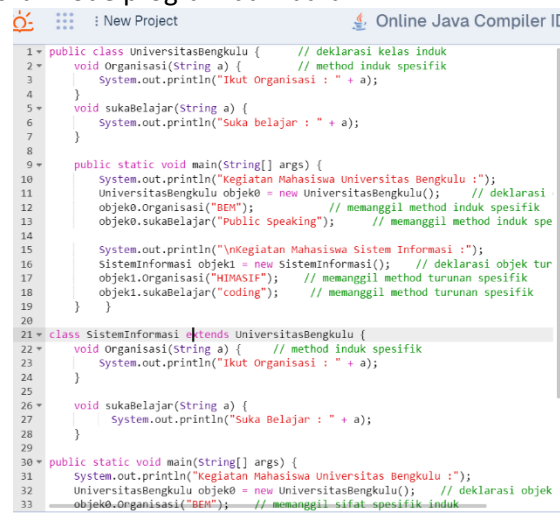
tampilInfo() dan belajar(). Kelas MahasiswaS2, sebagai turunan, menambahkan atribut khusus topikTesis dan method menulisTesis(), serta meng-override method tampilInfo() untuk menyertakan informasi tambahan. Ini memungkinkan kelas anak untuk memanfaatkan dan memperluas fungsionalitas dari kelas induk, sehingga menyelesaikan permasalahan dengan memberikan fleksibilitas untuk menambahkan fitur spesifik tanpa mengubah struktur dasar yang sudah ada.

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

#### 1) Rancang desain solusi atau algoritma

1. Definisikan kelas induk dengan atribut, constructor, dan method
2. Definisikan kelas turunan
3. Algoritma dengan mendefinisikan kelas dan implementasi program utama
4. Selesai.

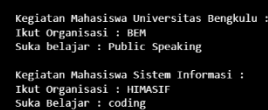
#### 2) Tuliskan kode program dan luaran



```
1 public class UniversitasBengkulu { // deklarasi kelas induk
2 void Organisasi(String a) { // method induk spesifik
3     System.out.println("Ikut Organisasi : " + a);
4 }
5 void sukaBelajar(String a) {
6     System.out.println("Suka belajar : " + a);
7 }
8
9 public static void main(String[] args) {
10     System.out.println("Kegiatan Mahasiswa Universitas Bengkulu :");
11     UniversitasBengkulu objek0 = new UniversitasBengkulu(); // deklarasi
12     objek0.Organisasi("BEM"); // memanggil method induk spesifik
13     objek0.sukaBelajar("Public Speaking"); // memanggil method induk spe
14
15     System.out.println("\nKegiatan Mahasiswa Sistem Informasi :");
16     SistemInformasi objek1 = new SistemInformasi(); // deklarasi objek tur
17     objek1.Organisasi("HIMASIF"); // memanggil method turunan spesifik
18     objek1.sukaBelajar("coding"); // memanggil method turunan spesifik
19 }
20
21 class SistemInformasi extends UniversitasBengkulu {
22 void Organisasi(String a) { // method induk spesifik
23     System.out.println("Ikut Organisasi : " + a);
24 }
25
26 void sukaBelajar(String a) {
27     System.out.println("Suka Belajar : " + a);
28 }
29
30 public static void main(String[] args) {
31     System.out.println("Kegiatan Mahasiswa Universitas Bengkulu :");
32     UniversitasBengkulu objek0 = new UniversitasBengkulu(); // deklarasi objek
33     objek0.Organisasi("BEM"); // memanggil sifat spesifik induk
34 }
```

#### Luaran

Output Generated Files



```
Kegiatan Mahasiswa Universitas Bengkulu :
Ikut Organisasi : BEM
Suka belajar : Public Speaking

Kegiatan Mahasiswa Sistem Informasi :
Ikut Organisasi : HIMASIF
Suka Belajar : coding
```

Compiled and executed in 1.835 sec(s)

#### Source code:

```
public class UniversitasBengkulu { // deklarasi kelas induk
    void Organisasi(String a) { // method induk spesifik
        System.out.println("Ikut Organisasi : " + a);
    }
    void sukaBelajar(String a) {
        System.out.println("Suka belajar : " + a);
    }

    public static void main(String[] args) {
        System.out.println("Kegiatan Mahasiswa Universitas Bengkulu :");
        UniversitasBengkulu objek0 = new UniversitasBengkulu(); // deklarasi objek induk
        objek0.Organisasi("BEM"); // memanggil method induk spesifik
        objek0.sukaBelajar("Public Speaking"); // memanggil method induk spesifik
    }
}
```

```

        System.out.println("\nKegiatan Mahasiswa Sistem Informasi :");
        SistemInformasi objek1 = new SistemInformasi(); // deklarasi objek turunan
        objek1.Organisasi("HIMASIF"); // memanggil method turunan spesifik
        objek1.sukaBelajar("coding"); // memanggil method turunan spesifik
    } }

class SistemInformasi extends UniversitasBengkulu {
    void Organisasi(String a) { // method induk spesifik
        System.out.println("Ikut Organisasi : " + a);
    }

    void sukaBelajar(String a) {
        System.out.println("Suka Belajar : " + a);
    }

    public static void main(String[] args) {
        System.out.println("Kegiatan Mahasiswa Universitas Bengkulu :");
        UniversitasBengkulu objek0 = new UniversitasBengkulu(); // deklarasi objek induk
        objek0.Organisasi("BEM"); // memanggil sifat spesifik induk
        objek0.sukaBelajar("Public Speaking"); // memanggil method dengan variabel

        System.out.println("\nKegiatan Mahasiswa Sistem Informasi :");
        SistemInformasi objek1 = new SistemInformasi(); // deklarasi objek turunan
        objek1.Organisasi("HIMASIF"); // memanggil sifat spesifik turunan
        objek1.sukaBelajar("coding"); // memanggil method dengan variabel
    }
}

```

#### **[No.1] Kesimpulan**

##### 1) Analisa

##### a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

- Kelas Induk dan Kelas Turunan:

Ada kelas utama bernama UniversitasBengkulu, yang memiliki dua aktivitas: ikut organisasi dan suka belajar.

Kelas lain, SistemInformasi, adalah turunan dari kelas utama. Ini berarti ia "mewarisi" aktivitas dari UniversitasBengkulu, tetapi juga bisa memiliki versi aktivitasnya sendiri.

- Memanggil Aktivitas:

Di bagian utama program, kita membuat objek dari kedua kelas. Ketika kita memanggil aktivitas dari kelas SistemInformasi, program menjalankan versi aktivitas yang telah diubah untuk kelas ini.

- Hasil yang Ditampilkan:

Program ini akan mencetak kegiatan mahasiswa di Universitas Bengkulu secara umum, lalu menampilkan kegiatan khusus untuk mahasiswa Sistem Informasi.

Secara keseluruhan, program ini memberikan gambaran sederhana tentang bagaimana objek dapat saling terhubung dan berbagi fitur dalam pemrograman, sehingga memudahkan kita dalam mengatur dan memahami kode.

Contoh Jawaban:

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Putri G1A000001</b>	<b>Tipe Data</b>	<b>26 Agustus 2022</b>

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

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

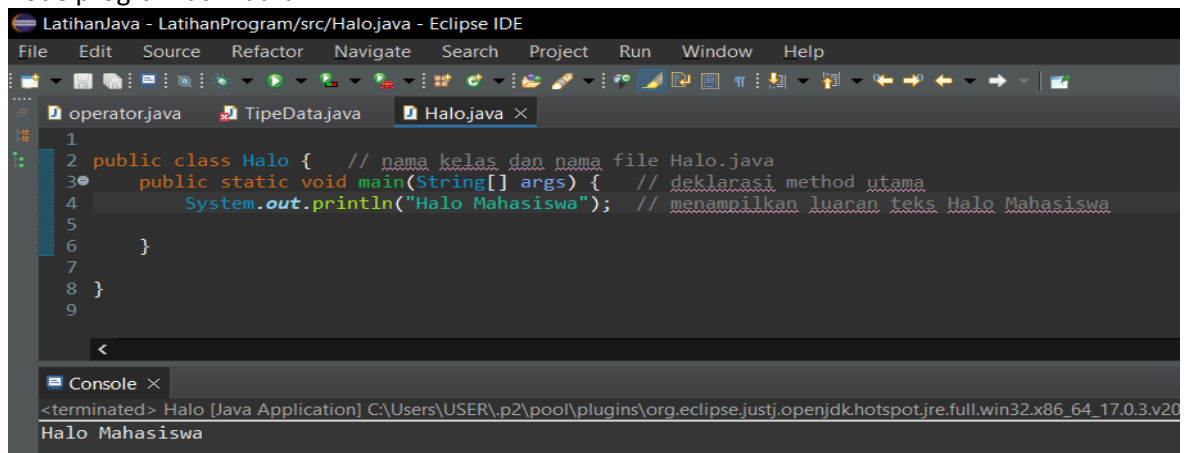
**[No.1] Analisis dan Argumentasi**

- 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



The screenshot shows the Eclipse IDE interface. The main editor window displays the code for `Halo.java`. The code is as follows:

```
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
```

Below the editor, the Console window shows the output of the program:

```
<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)