

Nama & NPM	Topik:	Tanggal:
Nabilla Azizah Mutiara Suriadi G1F024031 Ivanaly dees Tumangger G1F024033 Cut Fitriani G1F024079	Kelas	18 September 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Diketahui dari soal :

Apabila diketahui kelas induk adalah Mahasiswa dan Kelas anak adalah turunan dari mahasiswa maka:

- a) Analisa atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!
- (b) Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!
- (c) Rekomendasi atribut, method, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!
- (d) Desain kode program Java yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

Jawab:

- a) -Disini kami tidak menggunakan atribut didalam kode pemrogramannya.

-Method yang ada yaitu:

- tipe data string Organisasi(String a)
- sukaBelajar(String a)

-Tidak ada konstruktor yang dideklarasikan maka konstruktor tanpa parameter yang dipanggil secara otomatis saat objek dari kelas dibuat.

- b) Kelas Mahasiswa tidak adanya atribut yang dispesifikan, tetapi memiliki metode seperti, mengikuti organisasi, dan belajar minat bakat.
Kelas Anak, menerima semua atribut dan metode kelas induk(mahasiswa) dari kelas induk dan menambahkan fitur-fitur khusus untuk mewakili karakteristik yang lebih spesifik
- c) Atribut, method, dan constructor yang kami gunakan yaitu seperti organisasi yang diikuti pada masa kuliah dan kegiatan minat bakat, seperti paduan suara dan fotik (minat pada bidang fotografi).

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menerapkan prinsip pemrograman berorientasi objek (OOP) untuk mendesain kelas dan objek yang sesuai dengan hierarki data yang dibutuhkan.
- 2) Alasan solusi ini karena pendekatan OOP memungkinkan penggunaan kembali kode (reuse) melalui inheritance, yang mengurangi duplikasi dan memudahkan pemeliharaan. Selain itu, OOP memudahkan pemodelan dunia nyata, sehingga lebih intuitif bagi pengembang dan pengguna.
- 3) Perbaiki kode program dengan cara menyusun ulang kelas-kelas dengan memisahkan atribut dan method yang umum dalam kelas induk, dan menambahkan fitur spesifik di kelas anak.

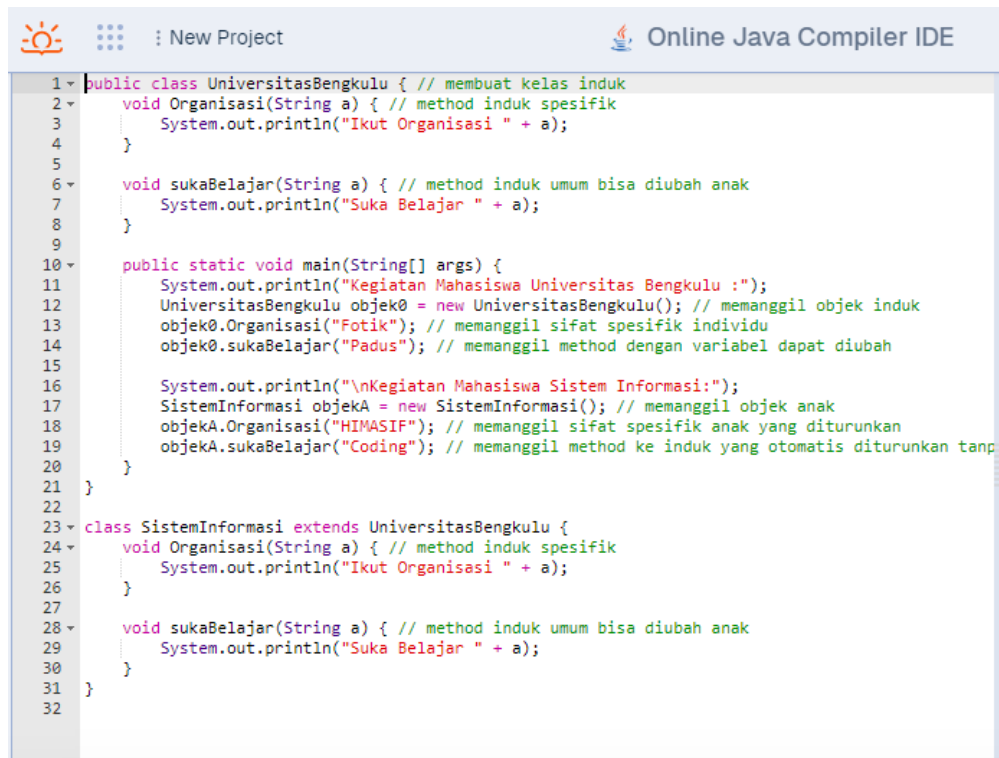
[No.3] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma public class mahasiswa dan kelas anak (turunan dari mahasiswa)

- (a) 1.Mendefinisikan kelas induk menjadi Mahasiswa
- (b) Mendeklarasikan atribut ke dalam variabel
- (c) Membuat constructor untuk menginisialisasi atribut
- (d) Membuat method:
- (e) Mendefinisikan Kelas Anak
- (f) Mendeklarasikan atribut tambahan:
- (g) Membuat constructor:
- (h) Panggil constructor induk dengan menggunakan super().
- (i) Membuat method tambahan
- (j) Panggil method untuk menampilkan data dan aktivitas.
- (k) Mencetak hasil dengan perintah System.out.println
- (l) Selesai

2) Kode program dan luaran



```
1 public class UniversitasBengkulu { // membuat kelas induk
2     void Organisasi(String a) { // method induk spesifik
3         System.out.println("Ikut Organisasi " + a);
4     }
5
6     void sukaBelajar(String a) { // method induk umum bisa diubah anak
7         System.out.println("Suka Belajar " + a);
8     }
9
10    public static void main(String[] args) {
11        System.out.println("Kegiatan Mahasiswa Universitas Bengkulu :");
12        UniversitasBengkulu objek0 = new UniversitasBengkulu(); // memanggil objek induk
13        objek0.Organisasi("Fotik"); // memanggil sifat spesifik individu
14        objek0.sukaBelajar("Padus"); // memanggil method dengan variabel dapat diubah
15
16        System.out.println("\nKegiatan Mahasiswa Sistem Informasi:");
17        SistemInformasi objekA = new SistemInformasi(); // memanggil objek anak
18        objekA.Organisasi("HIMASIF"); // memanggil sifat spesifik anak yang diturunkan
19        objekA.sukaBelajar("Coding"); // memanggil method ke induk yang otomatis diturunkan tanpa
20    }
21 }
22
23 class SistemInformasi extends UniversitasBengkulu {
24     void Organisasi(String a) { // method induk spesifik
25         System.out.println("Ikut Organisasi " + a);
26     }
27
28     void sukaBelajar(String a) { // method induk umum bisa diubah anak
29         System.out.println("Suka Belajar " + a);
30     }
31 }
32 }
```

```
Kegiatan Mahasiswa Universitas Bengkulu :
Ikut Organisasi Fotik
Suka Belajar Padus

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

- a) Screenshot/ Capture potongan kode dan hasil luaran
- b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Dan output yang dihasilkan sesuai dengan apa yang diinginkan.

Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.4] Kesimpulan

1) Analisa

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

jawaban Analisa:

- a) Kelas induk Mahasiswa berfungsi sebagai kerangka dasar yang mendefinisikan atribut dan method umum untuk objek mahasiswa. Kelas anak seperti MahasiswaSpesialisasi dapat mewarisi atribut dan method dari kelas induk, sekaligus menambahkan fungsionalitas baru seperti spesialisasi. Kelas anak dapat menggunakan `super()` untuk memanggil constructor dan method dari kelas induk. Konsep pewarisan dalam OOP memungkinkan pembuatan hierarki kelas yang modular dan dapat diperluas, sehingga membuat kode lebih terorganisir
- b) Penggunaan konstruktor di kelas turunan yang memanggil konstruktor kelas induk (nama, nim, jurusan) menunjukkan pentingnya inisialisasi atribut yang diwarisi.

Refleksi

Dalam desain pemrograman yang menggunakan konsep pewarisan (inheritance) di Java, kami bisa memahami bagaimana penggunaan kelas induk dan kelas anak membawa keuntungan dalam penggunaan modularitas, pemeliharaan kode, serta perluasan fungsionalitas.