

| Nama & NPM | Topik: | Tanggal: |
|---|----------------------|-------------------|
| Mellinna Husadya G1F024006 Mosa Violeta Salsabila G1F024028 Cindy Aurora G1F024066 | Tugas kelompok Kelas | 19 September 2024 |

[No. 1] Identifikasi Masalah:

1. 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)!

Jawab:

Kelas induk `Mahasiswa` tidak memiliki atribut yang dideklarasikan secara eksplisit, sehingga informasi tentang mahasiswa dikelola melalui metode yang menerima parameter string. Metode yang ada, seperti `nama`, `npm`, `angkatan`, `kelas`, `prodi`, dan `universitas`, berfungsi untuk mencetak informasi mahasiswa secara langsung tanpa menyimpan data dalam atribut. Kelas ini juga tidak memiliki constructor khusus, menggunakan constructor default yang disediakan oleh Java. Kelas anak seperti `MahasiswaBaru` dan `MahasiswaBaru2` tidak menambahkan atribut baru, tetapi mengoverride metode `nama` dan `npm` untuk memberikan keterangan tambahan, yang menunjukkan perbedaan jenis mahasiswa.

- (b) Evaluasi perbedaan kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

Jawab:

Perbedaan utama antara kelas induk `Mahasiswa` dan kelas anak terletak pada fungsionalitas metode yang mereka tawarkan. Kelas induk menyediakan metode yang bersifat umum untuk mencetak informasi mahasiswa, sementara kelas anak `MahasiswaBaru` dan `MahasiswaBaru2` memperluas fungsionalitas ini dengan menambahkan keterangan pada output untuk membedakan mahasiswa baru dari yang lainnya. Dengan cara ini, kelas anak memberikan konteks yang lebih spesifik, meskipun tetap menggunakan struktur dan metode dasar dari kelas induk.

- (c) Rekomendasi atribut, method, dan constructor yang bisa digunakan bersama kelas induk dan kelas anak!

Jawab:

Untuk meningkatkan efisiensi dan kejelasan, disarankan agar kelas `Mahasiswa` memiliki atribut untuk menyimpan informasi seperti `nama`, `npm`, `angkatan`, `kelas`, `prodi`, dan `universitas`, serta menyertakan constructor yang memungkinkan inisialisasi data ini. Selain itu, menambahkan metode `displayInfo()` dalam kelas induk akan memudahkan kelas anak untuk mengoverride dan menampilkan informasi dengan format yang lebih sesuai, serta memastikan semua kelas memiliki akses ke metode yang sama untuk mencetak informasi mahasiswa.

- (d) Desain kode program Java yang berisi atribut, method, dan constructor dari kelas induk Mahasiswa dan kelas anak (turunan dari Mahasiswa)!

[No.1] Analisis dan Argumentasi

Pada kode program kali ini kami di minta untuk membuat desain kode program java sendiri yang berisikan atribut,method dan constructor dari kelas induk mahasiswa dan kelas anak (turunan dari mahasiswa),disini kami menggunakan web java compiler untuk menjalan kan kode program yang telah kami buat

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- Masuk ke web java compiler online
- Mulai kelas
- Menyusun kode program sesuai perintah
- Runkan kode program yang telah di susun
- Output keluar sesuai dengan permintaan data
- Selesai

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a 'Mahasiswa' class with several methods: 'nama', 'npm', 'angkatan', 'kelas', and 'prodi'. Each method takes a string parameter and prints it with a label. The 'main' method creates a 'Mahasiswa' object and calls these methods with specific data. The output window on the right shows the results of these method calls.

```

1 public class Mahasiswa { // kelas induk
2
3     // Method untuk mencetak nama mahasiswa
4     void nama(String a) {
5         System.out.println("Nama: " + a);
6     }
7
8     // Method untuk mencetak NPM mahasiswa
9     void npm(String a) {
10        System.out.println("NPM: " + a);
11    }
12
13    // Method untuk mencetak angkatan mahasiswa
14    void angkatan(String a) {
15        System.out.println("Angkatan: " + a);
16    }
17
18    // Method untuk mencetak kelas mahasiswa
19    void kelas(String a) {
20        System.out.println("Kelas: " + a);
21    }
22
23    // Method untuk mencetak prodi mahasiswa
24    void prodi(String a) {
25        System.out.println("Prodi: " + a);
26    }
27
28    // Method untuk mencetak universitas mahasiswa
29    void universitas(String a) {
30        System.out.println("Universitas: " + a);
31    }
32
33    // Method utama untuk menjalankan program
34    public static void main(String[] args) {
35        // Membuat objek mahasiswa pertama
36        Mahasiswa objek1 = new Mahasiswa();
37        objek1.nama("Mellinna Husadya"); // Memanggil method
        // untuk mencetak nama
38        objek1.npm("G1F024006"); // Memanggil method untuk
        // mencetak NPM
39        objek1.angkatan("2024"); // Memanggil method untuk
        // mencetak angkatan
40        objek1.kelas("B"); // Memanggil method untuk mencetak
        // kelas
41        objek1.prodi("Sistem Informasi"); // Memanggil method
        // untuk mencetak prodi
42    }
43 }
  
```

Output:

```

Nama: Mellinna Husadya
NPM: G1F024006
Angkatan: 2024
Kelas: B
Prodi: Sistem Informasi
Universitas: Bengkulu

Nama: Moza Violeta Salsabila
NPM: G1F024028
Angkatan: 2024
Kelas: B
Prodi: Sistem Informasi
Universitas: Bengkulu

Nama: Cindy Aurora
NPM: G1F024066
Angkatan: 2024
Kelas: B
Prodi: Sistem Informasi
Universitas: Bengkulu
  
```

== Code Execution Successful ==

This screenshot shows the 'main' method of the 'Mahasiswa' class. It includes comments explaining each step: creating a 'Mahasiswa' object, and then calling the 'nama', 'npm', 'angkatan', 'kelas', and 'prodi' methods with specific data values.

```

24 void prodi(String a) {
25     System.out.println("Prodi: " + a);
26 }
27
28 // Method untuk mencetak universitas mahasiswa
29 void universitas(String a) {
30     System.out.println("Universitas: " + a);
31 }
32
33 // Method utama untuk menjalankan program
34 public static void main(String[] args) {
35     // Membuat objek mahasiswa pertama
36     Mahasiswa objek1 = new Mahasiswa();
37     objek1.nama("Mellinna Husadya"); // Memanggil method
        // untuk mencetak nama
38     objek1.npm("G1F024006"); // Memanggil method untuk
        // mencetak NPM
39     objek1.angkatan("2024"); // Memanggil method untuk
        // mencetak angkatan
40     objek1.kelas("B"); // Memanggil method untuk mencetak
        // kelas
41     objek1.prodi("Sistem Informasi"); // Memanggil method
        // untuk mencetak prodi
  
```

```

42     objek1.universitas("Bengkulu"); // Memanggil method
        untuk mencetak universitas
43
44     System.out.println(); // Memisahkan output dengan baris
        kosong
45
46     // Membuat objek mahasiswa kedua
47     Mahasiswa objek2 = new Mahasiswa();
48     objek2.nama("Moza Violeta Salsabila");// Memanggil
        method untuk mencetak nama
49     objek2.npm("G1F024028"); // Memanggil method untuk
        mencetak NPM
50     objek2.angkatan("2024"); // Memanggil method untuk
        mencetak angkatan
51     objek2.kelas("B"); // Memanggil method untuk mencetak
        kelas
52     objek2.prodi("Sistem Informasi"); // Memanggil method
        untuk mencetak prodi
53     objek2.universitas("Bengkulu"); // Memanggil method
        untuk mencetak universitas
54
55     System.out.println(); // Memisahkan output dengan baris
        kosong

```

```

57     // Membuat objek mahasiswa ketiga
58     Mahasiswa objek3 = new Mahasiswa();
59     objek3.nama("Cindy Aurora"); // Memanggil method untuk
        mencetak nama
60     objek3.npm("G1F024066"); // Memanggil method untuk
        mencetak NPM
61     objek3.angkatan("2024"); // Memanggil method untuk
        mencetak angkatan
62     objek3.kelas("B"); // Memanggil method untuk mencetak
        kelas
63     objek3.prodi("Sistem Informasi"); // Memanggil method
        untuk mencetak prodi
64     objek3.universitas("Bengkulu"); // Memanggil method
        untuk mencetak universitas
65
66     System.out.println(); // Memisahkan output dengan
        baris kosong
67 }
68 }
69
70 class MahasiswaBaru extends Mahasiswa {
71     // Mengoverride method nama untuk menambahkan informasi
        khusus

```

```

72     @Override
73     void nama(String a) {
74         System.out.println("Nama (Mahasiswa Baru): " + a);
75     }
76
77     // Mengoverride method npm
78     @Override
79     void npm(String a) {
80         System.out.println("NPM (Mahasiswa Baru): " + a);
81     }
82 }
83
84 class MahasiswaBaru2 extends Mahasiswa {
85     // Mengoverride method nama untuk menambahkan informasi
        khusus
86
87     @Override
88     void nama(String a) {
89         System.out.println("Nama (Mahasiswa Baru 2): " + a);
90     }
91
92     // Mengoverride method npm
93     @Override
94     void npm(String a) {

```

```

93     void npm(String a) {
94         System.out.println("NPM (Mahasiswa Baru 2): " + a);
95     }
96 }
97

```

b) Analisa luaran yang dihasilkan

Jawab:

Luaran sudah sesuai dengan program yang disusun.

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

[No.1] Kesimpulan

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

1) Analisa

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

Jawab:

Kode program di atas mendefinisikan kelas induk Mahasiswa yang menyimpan informasi dasar tentang mahasiswa, seperti nama, NPM, angkatan, kelas, prodi, dan universitas, serta menyediakan metode untuk mencetak informasi tersebut. Terdapat dua kelas turunan, MahasiswaBaru dan MahasiswaBaru2, yang mengoverride metode tertentu untuk menambahkan label khusus pada output informasi mahasiswa baru. Di dalam metode main, tiga objek Mahasiswa dibuat dengan data yang berbeda, dan masing-masing objek mencetak informasi mereka dengan pemisah baris kosong untuk keterbacaan. Program ini menunjukkan penggunaan polimorfisme dengan memodifikasi metode dari kelas induk, meskipun struktur kodenya dapat ditingkatkan dengan menggunakan konstruktor untuk inisialisasi dan metode tunggal untuk mencetak semua informasi sekaligus, mengurangi duplikasi kode.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Dasar pengambilan keputusan dalam kode program ini berfokus pada cara menyusun kelas dan metode untuk menyimpan serta mencetak informasi tentang mahasiswa dengan rapi. Kelas induk Mahasiswa menyediakan berbagai metode untuk menampilkan informasi dasar, sementara kelas turunan MahasiswaBaru dan MahasiswaBaru2 mengoverride beberapa metode untuk menambahkan label yang lebih spesifik pada output, menunjukkan fleksibilitas dalam menyesuaikan informasi berdasarkan jenis mahasiswa. Penggunaan metode terpisah untuk mencetak setiap atribut memberikan kejelasan dan memudahkan pemahaman data, meskipun akan lebih efisien jika semua informasi dicetak dalam satu metode. Pendekatan ini mencerminkan prinsip-prinsip OOP seperti pewarisan dan polimorfisme, serta menghasilkan struktur yang mudah dikembangkan untuk kebutuhan di masa depan.