

## Tugas Pendahuluan: Modul 4

NIM: 105223024

Nama: Farhan Kholid

Instruksi: Jawablah pertanyaan-pertanyaan berikut dengan jelas dan rinci. Gunakan contoh kode jika diperlukan untuk memperjelas jawaban Anda. Kumpulkan dengan format **TP4\_NIM\_PrakPBO.pdf**. Deadline, Kamis 20 Maret 2025, pukul 12:00 (sebelum praktikum). Link pengumpulan: <https://forms.gle/Sjwi5tLqHVzWFiVD8>

### Soal

#### 1. Konsep OOP I: Class dan Object.

##### a. Jelaskan perbedaan antara class dan object dalam OOP!

- Class adalah blueprint atau template untuk membuat objek. Ia mendefinisikan atribut (data) dan metode (perilaku) yang dimiliki oleh objek.
- Object adalah instansiasi dari class. Ia mewakili entitas konkret yang memiliki nilai spesifik untuk atribut dan dapat melakukan tindakan melalui metode.

##### b. Berikan contoh implementasi class dan object dalam Java!

```
1  // Class
2  public class Car {
3      // Attributes
4      private String brand;
5      private int year;
6
7      // Constructor
8      public Car(String brand, int year) {
9          this.brand = brand;
10         this.year = year;
11     }
12
13     // Method
14     public void displayInfo() {
15         System.out.println("Brand: " + brand + ", Year: " + year);
16     }
17 }
```

Ini adalah penerapan Class pada Bahasa pemograman Java

```

1 // Importing the Car class
2 public class Main {
3     public static void main(String[] args) {
4         Car myCar = new Car(brand:"Toyota", year:2020);
5         myCar.displayInfo(); // Output: Brand: Toyota, Year: 2020
6     }
7 }

```

ini adalah penerapan dalam membuat Objek pada Bahasa pemograman Java.

## 2. Konsep OOP II: Encapsulation, polymorphism, abstraction.

### a. Jelaskan konsep Encapsulation dalam OOP.

Encapsulation (enkapsulasi) adalah konsep menyembunyikan data dengan cara:

- Membuat atribut kelas sebagai private.
- Menyediakan metode public (getter/setter) untuk mengakses atau memodifikasi data. Tujuannya adalah mencegah akses langsung ke data dan memastikan validasi.
- Contoh :

```

1 public class Encapsulation {
2     private String name; // Atribut private
3
4     // Getter
5     public String getName() {
6         return name;
7     }
8
9     // Setter
10    public void setName(String name) {
11        if (name != null) {
12            this.name = name;
13        }
14    }
15 }

```

### b. Apa perbedaan antara inheritance dan polymorphism?

- Inheritance (Pewarisan): Mekanisme di mana sebuah kelas (subclass) mewarisi atribut dan metode dari kelas lain (superclass). Contoh: *class Cat extends Animal*.
- Polymorphism (Polimorfisme): Kemampuan objek untuk memiliki banyak bentuk. Misalnya, metode `sound()` pada kelas `Animal` bisa di-override di subclass `Cat` dan `Dog`.

**c. Apa itu abstraction dalam OOP? Jelaskan perbedaannya dengan encapsulation!**

Abstraction (Abstraksi): Menyembunyikan detail kompleks dan menampilkan fungsionalitas esensial. Contoh: menggunakan *abstract class* atau *interface*.

Perbedaan dengan Encapsulation:

- *Encapsulation* fokus pada *data hiding* (melindungi data).
- *Abstraction* fokus pada *complexity hiding* (menyederhanakan interaksi).

**3. Sebutkan minimal 3 perbedaan utama antara pemrograman berorientasi objek (OOP) dan pemrograman prosedural!**

Struktur:

- OOP: Berbasis objek (data + metode).
- Prosedural: Berbasis fungsi dan data terpisah.

Reusability:

- OOP: Menggunakan inheritance dan polymorphism.
- Prosedural: Menggunakan fungsi berulang.

Keamanan:

- OOP: Encapsulation melindungi data.
- Prosedural: Data bersifat global, rentan modifikasi.

**4. Jelaskan peran konstruktor dalam Java!**

Konstruktor adalah metode khusus yang:

- Dipanggil saat objek dibuat (new).
- Menginisialisasi nilai awal atribut objek.
- Memiliki nama sama dengan kelas dan tidak memiliki return type.

Contoh :

```

1  public class Constructor {
2      private String name;
3
4      // Konstruktor
5      public Student(String name) {
6          this.name = name;
7      }
8  }

```

## 5. Jelaskan perbedaan antara static method dan non-static method dalam Java!

Static Method:

- Dimiliki oleh kelas (bukan objek).
- Dipanggil via nama kelas (e.g., *Math.sqrt()*).
- Tidak bisa mengakses atribut non-static.

Non-Static Method:

- Dimiliki oleh objek.
- Dipanggil via objek (e.g., *car.startEngine()*).
- Bisa mengakses atribut static dan non-static.

Contoh:

```

1  public class Calculator {
2      // Static method untuk menambah 2 nomor
3      public static int add(int a, int b) {
4          return a + b;
5      }
6
7      // Non-static method untuk mengalikan 2 nomor
8      public int multiply(int a, int b) {
9          return a * b;
10     }
11
12     Run | Debug
13     public static void main(String[] args) {
14         // Static method call
15         int sum = Calculator.add(a:2, b:3);
16         System.out.println("Sum: " + sum);
17
18         // Non static method call
19         Calculator calc = new Calculator();
20         int product = calc.multiply(a:2, b:3);
21         System.out.println("Product: " + product);
22     }

```

## **Referensi**

- [1]. Oracle Java Documentation
- [2]. "Head First Java" by Kathy Sierra & Bert Bates