

## Tugas Pendahuluan: Modul 4

NIM: 105223002

Nama: Gerald Eberhard

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!

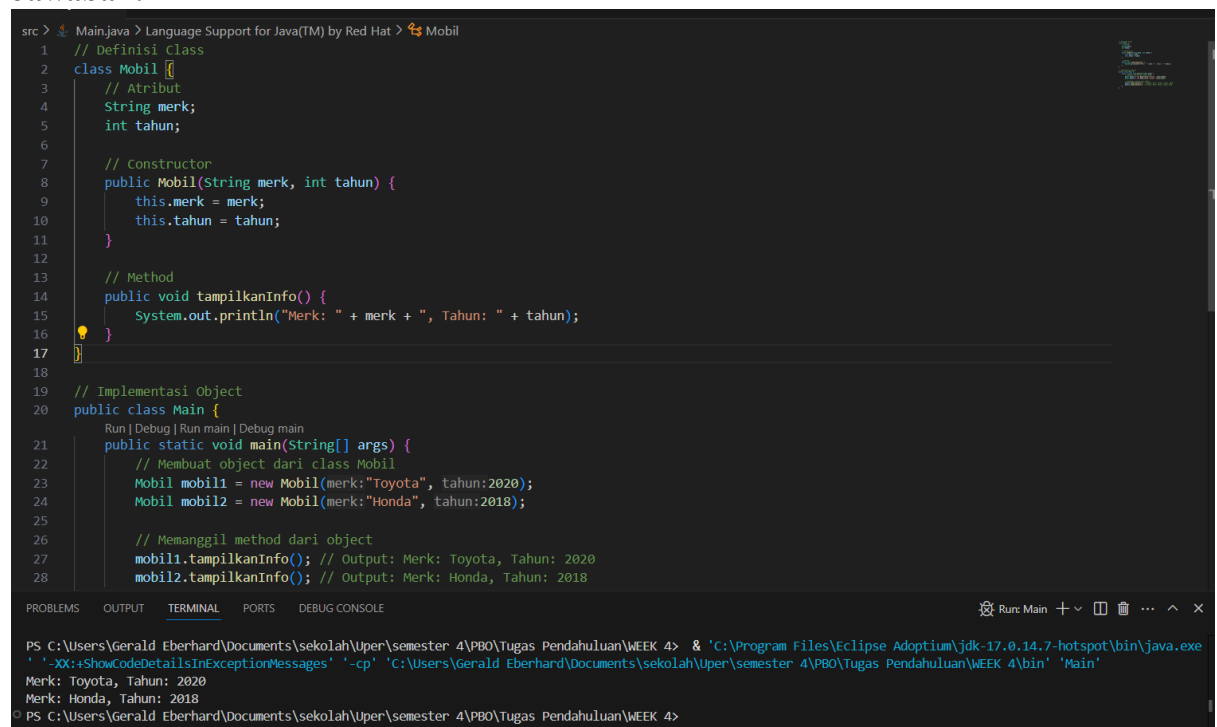
#### **Jawaban:**

Class adalah blueprint atau cetakan yang mendefinisikan atribut (data) dan perilaku (method) dari suatu entitas. Class tidak memiliki keberadaan nyata di memori sampai sebuah object dibuat darinya. Sedangkan Object adalah instance (contoh nyata) dari sebuah class. Object diciptakan dari class dan memiliki data serta perilaku yang didefinisikan oleh class tersebut. Jadi perbedaan utamanya adalah Class

s merupakan blue print dari objek. Sedangkan Objek adalah apa yang dihasilkan oleh blueprint yang tersimpan di dalam class.

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

#### **Jawaban:**



```
src > Main.java > Language Support for Java(TM) by Red Hat > Mobil
1 // Definisi Class
2 class Mobil {
3     // Atribut
4     String merk;
5     int tahun;
6
7     // Constructor
8     public Mobil(String merk, int tahun) {
9         this.merk = merk;
10        this.tahun = tahun;
11    }
12
13    // Method
14    public void tampilkanInfo() {
15        System.out.println("Merk: " + merk + ", Tahun: " + tahun);
16    }
17
18
19 // Implementasi Object
20 public class Main {
21     Run | Debug | Run main | Debug main
22     public static void main(String[] args) {
23         // Membuat object dari class Mobil
24         Mobil mobil1 = new Mobil(merk:"Toyota", tahun:2020);
25         Mobil mobil2 = new Mobil(merk:"Honda", tahun:2018);
26
27         // Memanggil method dari object
28         mobil1.tampilkanInfo(); // Output: Merk: Toyota, Tahun: 2020
29         mobil2.tampilkanInfo(); // Output: Merk: Honda, Tahun: 2018
30     }
31 }

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.14.7-hotspot\bin\java.exe'
'-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4\bin' 'Main'
Merk: Toyota, Tahun: 2020
Merk: Honda, Tahun: 2018
PS C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4>
```

Classnya adalah mobil dan objeknya adalah Toyota dan honda.

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

a. Jelaskan konsep Encapsulation dalam OOP.

**Jawaban:**

Encapsulation adalah konsep dalam OOP yang mengacu pada pembungkusan data (atribut) dan method yang bekerja pada data tersebut ke dalam satu unit, yaitu class. Encapsulation juga melibatkan pembatasan akses langsung ke data dengan menggunakan modifier akses seperti private, protected, atau public, dan menyediakan akses melalui method (getter dan setter). Tujuannya adalah untuk melindungi data dari akses yang tidak sah dan memastikan integritas data.

b. Apa perbedaan antara inheritance dan polymorphism?

**Jawaban:**

Inheritance (Pewarisan) adalah konsep di mana sebuah class (subclass) mewarisi atribut dan method dari class lain (superclass). Ini memungkinkan kode untuk digunakan kembali dan membangun hubungan "is-a" (contoh: "Mobil adalah Kendaraan"). Sedangkan, Polymorphism (Polimorfisme) adalah konsep di mana sebuah method dapat memiliki implementasi yang berbeda tergantung pada object yang memanggilnya. Ada dua jenis: compile-time (method overloading) dan runtime (method overriding).

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

**Jawab:**

Abstraction adalah konsep di mana kita menyembunyikan detail implementasi dan hanya menunjukkan fungsionalitas yang relevan kepada pengguna. Ini biasanya dicapai dengan menggunakan class abstrak atau interface. Perbedaannya enkapsulasi fokus pada pengamanan data dengan membatasi akses langsung ke atribut dan menyediakan getter/setter. Sedangkan, Abstraction fokus pada penyembunyian detail implementasi dan hanya menunjukkan apa yang perlu diketahui oleh pengguna.

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

**Jawab:**

- Adanya fleksibilitas dan skalabilitas yang tinggi. Karena memungkinkan untuk bekerja lintas file. Sedangkan pemograman prosedural tidak.
- Pendekatan pemograman yang berbeda. Pada OOP fokusnya adalah data (atribut) yang memiliki data dan perilaku dibuat untuk membuat object dan interaksi antar object. Sedangkan pada pemograman procedural berfokus pada fungsi dan prosedur yang dipanggil secara berutan.
- Struktur data, Pada OOP data dan fungsi yang bekerja tersebut dibungkus di dalam class. Sedangkan Pada Prosedural data biasanya disimpan dalam variabel global dan fungsi terpisah mengoperasikan data tersebut.

4. Jelaskan peran konstruktor dalam Java!

**Jawab:**

- Mengatur nilai awal untuk atribut object saat object dibuat.
- Memiliki nama yang sama dengan nama class dan tidak memiliki tipe pengembalian (bahkan bukan void).
- Secara otomatis dipanggil saat object dibuat menggunakan keyword new.

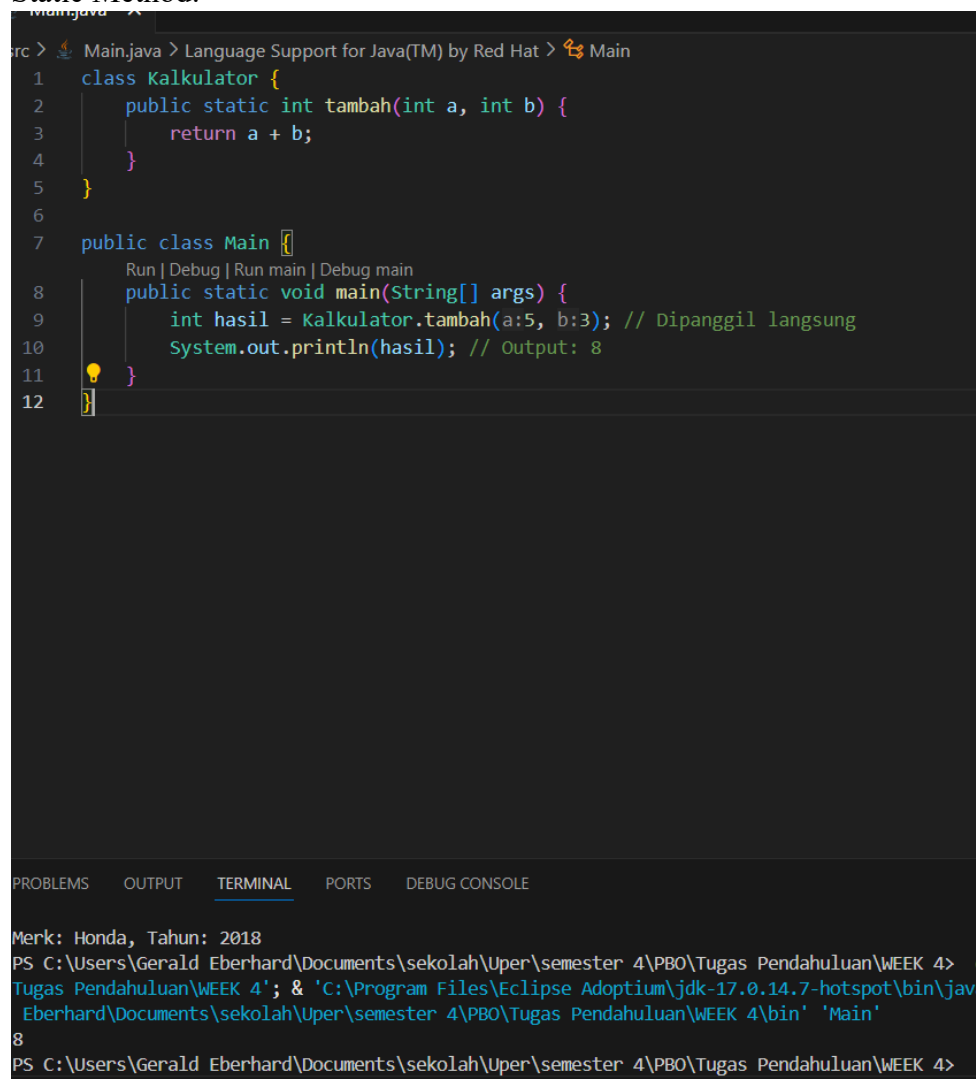
Pada intinya dari constructor ini adalah constructor bertujuan untuk menginisialisasi objek

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

**Jawab:**

Static method memiliki keyword static pada saat didefinisikan dan bukan object sehingga dapat dipanggil tanpa membuat instance. Sedangkan Non-Static Method tidak menggunakan keyword static dan merupakan bagian dari object sehingga harus dipanggil dengan membuat instance dari kelas tersebut. Contohnya ada pada kedua code berikut:

Static Method:

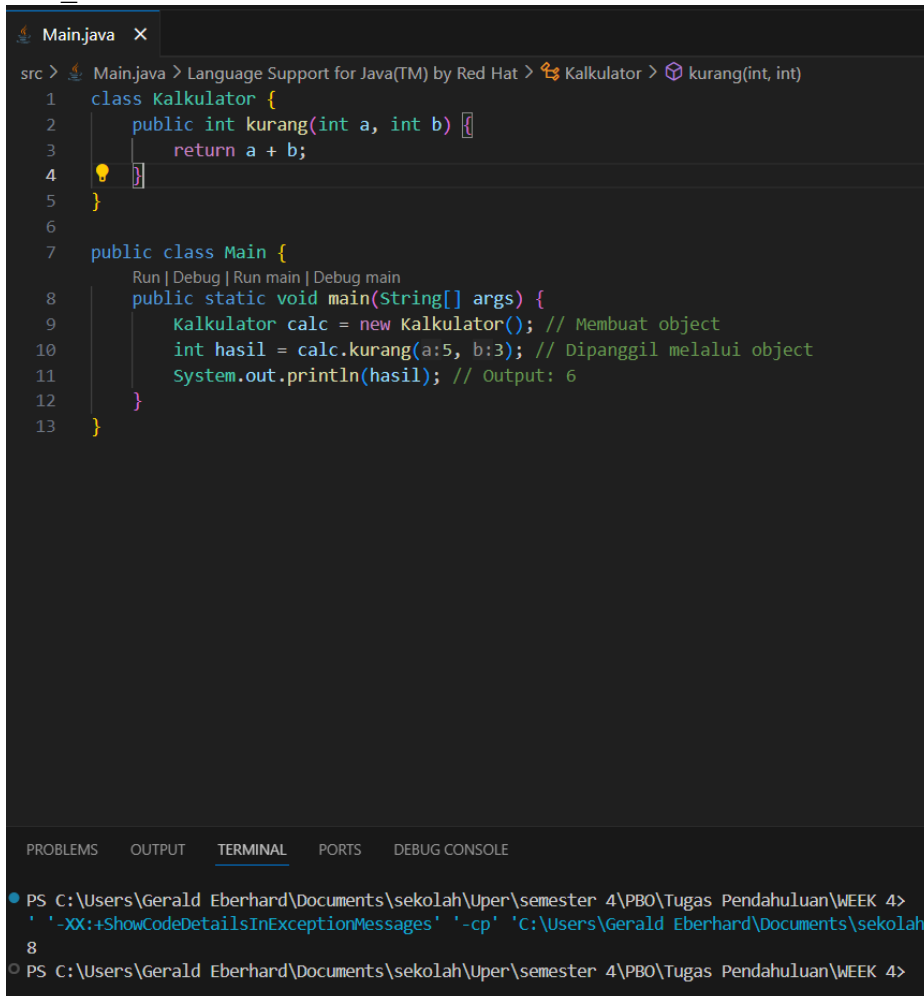


```
src > Main.java > Language Support for Java(TM) by Red Hat > Main
1  class Kalkulator {
2      public static int tambah(int a, int b) {
3          return a + b;
4      }
5  }
6
7  public class Main {
8      public static void main(String[] args) {
9          int hasil = Kalkulator.tambah(a:5, b:3); // Dipanggil langsung
10         System.out.println(hasil); // Output: 8
11     }
12 }
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

Merk: Honda, Tahun: 2018  
PS C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4> .\Tugas Pendahuluan\WEEK 4\ & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.14.7-hotspot\bin\java' Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4\bin\ 'Main'  
8  
PS C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4>

## Non\_Static Method:



```
src > Main.java > Language Support for Java(TM) by Red Hat > Kalkulator > kurang(int, int)
1  class Kalkulator {
2      public int kurang(int a, int b) {
3          return a + b;
4      }
5  }
6
7  public class Main {
8      public static void main(String[] args) {
9          Kalkulator calc = new Kalkulator(); // Membuat object
10         int hasil = calc.kurang(a:5, b:3); // Dipanggil melalui object
11         System.out.println(hasil); // Output: 6
12     }
13 }
```

PROBLEMS OUTPUT **TERMINAL** PORTS DEBUG CONSOLE

PS C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4> &  
' -XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Gerald Eberhard\Documents\sekolah\  
8  
PS C:\Users\Gerald Eberhard\Documents\sekolah\Uper\semester 4\PBO\Tugas Pendahuluan\WEEK 4>

Nah sudah terlihat ya perbedaannya, dimana yang satunya tidak harus ada instance sedangkan yang satunya harus ada instance.

## Referensi

- [1]. [Classes and Objects in Java - GeeksforGeeks](#)
- [2]. [Java Encapsulation](#)
- [3]. [Java Inheritance \(With Examples\)](#)
- [4]. [Java Polymorphism \(With Examples\)](#)
- [5]. [Abstraction in Java - GeeksforGeeks](#)
- [6]. [Differences between Procedural and Object Oriented Programming - GeeksforGeeks](#)
- [7]. [Java Constructors](#)
- [8]. [Java Static Keyword \(With Examples\)](#)