

## Tugas Pendahuluan: Modul 5

NIM: 105223024

Nama: Farhan Kholid

### Soal

#### 1. Cakupan variabel (Variable Scope):

- a. Jelaskan perbedaan antara variabel lokal, instance, kelas (static), dan parameter dalam Java!

**Jawaban:**

<b>Variabel Lokal</b>	: Dideklarasikan dalam method/blok, hanya bisa diakses di scope tersebut.
<b>Variabel Instance</b>	: Dimiliki objek, nilai unik per objek.
<b>Variabel Kelas (static)</b>	: Dimiliki kelas, nilai sama untuk semua objek.
<b>Parameter</b>	: Input untuk method/konstruktor.

- b. Mengapa variabel static memiliki nilai yang sama untuk semua objek dari kelas yang sama?

**Jawaban:** Karena variabel static disimpan di memori kelas, bukan objek, sehingga semua objek berbagi nilai yang sama.

#### 2. Java packages:

- a. Apa tujuan utama penggunaan package dalam Java?

**Jawaban:**

- Mengorganisasi kode.
- Menghindari konflik nama.
- Mengontrol akses dengan access modifier.

- b. Jelaskan perbedaan antara built-in package dan user-defined package dalam Java!

**Jawaban:**

- **Built-in:** Sudah disediakan Java (e.g., java.util).
- **User-defined:** Didefinisikan pengguna.

#### 3. Access Modifiers:

- a. Sebutkan dan jelaskan empat jenis access modifier dalam Java!

**Jawaban:**

<i>public</i>	:	Diakses di mana saja.
<i>private</i>	:	Hanya di dalam kelas.
<i>protected</i>	:	Di kelas, subclass, dan package.
<i>default</i>	:	Hanya di package yang sama.

**b. Mengapa access modifier private sering digunakan dalam konsep enkapsulasi?**

**Jawaban:** *private* mencegah akses langsung ke data, sehingga validasi via setter/getter terjaga.

#### **4. Enkapsulasi:**

**a. Apa yang dimaksud dengan enkapsulasi dalam pemrograman berorientasi objek?**

**Jawaban:** Prinsip OOP untuk menyembunyikan data dengan membatasi akses langsung ke variabel.

**b. Mengapa penggunaan getter dan setter penting dalam implementasi enkapsulasi?**

**Jawaban:** Memastikan validasi data dan fleksibilitas dalam modifikasi nilai.

#### **5. Array of object:**

**a. Apa perbedaan antara array biasa dan array of object dalam Java?**

**Jawaban:**

Array Biasa	:	Menyimpan primitif/objek langsung.
Array of Object	:	Menyimpan referensi ke objek.

**b. Sebutkan dan jelaskan dua cara untuk menginisialisasi array of object dalam Java!**

**Jawaban:**

1. Inisialisasi langsung: *Mahasiswa[] mhs = {new Mahasiswa()};*
2. Inisialisasi dengan loop.

#### **Referensi**

- [1]. Sierra, K. & Bates, B., Head First Java, 3rd Edition, O'Reilly Media, 2022.
- [2]. Oracle, "Java Tutorials: Variables",  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/variables.html>.

- [3]. Bloch, J., Effective Java, 3rd Edition, Addison-Wesley, 2018.
- [4]. Horstmann, C., Core Java Volume I, 12th Edition, Prentice Hall, 2021.
- [5]. Oracle, "Java Tutorials: Packages",  
<https://docs.oracle.com/javase/tutorial/java/package/packages.html>.
- [6]. GeeksforGeeks, "Packages in Java", <https://www.geeksforgeeks.org/packages-in-java/>.
- [7]. Oracle, "Controlling Access to Members of a Class",  
<https://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>.
- [8]. Jain, P., Java Programming for Beginners, Packt Publishing, 2020.
- [9]. Martin, R., Clean Code: A Handbook of Agile Software Craftsmanship, Prentice Hall, 2008.
- [10]. Larman, C., Applying UML and Patterns, 3rd Edition, Prentice Hall, 2004.
- [11]. Meyer, B., Object-Oriented Software Construction, 2nd Edition, Prentice Hall, 1997.
- [12]. Fowler, M., Refactoring: Improving the Design of Existing Code, Addison-Wesley, 2018.
- [13]. Sedgewick, R. & Wayne, K., Algorithms, 4th Edition, Addison-Wesley, 2011.
- [14]. Oracle, "Java Arrays",  
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>.
- [15]. Bloch, J., Effective Java, 3rd Edition, Addison-Wesley, 2018.