

# Object Oriented Programming Course

## Quiz 1

Lenka Melinda Florienka

D4 - Informatics Engineering, Information Technology, Malang State Polytechnic  
[lenkamelinda7@gmail.com](mailto:lenkamelinda7@gmail.com)

---

### 1. Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?

"Class" dalam pemrograman berorientasi objek adalah blueprint yang mendefinisikan struktur atau karakteristik sebuah objek. Itu berisi atribut (variabel) dan metode (fungsi) yang akan dimiliki oleh objek yang dibuat berdasarkan class tersebut.

### 2. Bagaimana Anda mendefinisikan objek dari suatu class dalam bahasa pemrograman Java?

Kita dapat mendefinisikan objek dari suatu class dalam bahasa pemrograman Java dengan menggunakan kata kunci "new" diikuti oleh nama class, seperti ini:

```
NamaClass namaObjek = new NamaClass();
```

### 3. Misalkan Anda memiliki class "Barang" dalam sistem informasi inventaris. Bagaimana Anda akan membuat objek "laptop" dari class tersebut?

Untuk membuat objek "laptop" dari class "Barang" dalam sistem informasi inventaris, kita perlu melakukan langkah-langkah berikut:

- Import class "Barang" jika belum dilakukan.
- Kemudian, Anda bisa membuat objek "laptop" dengan menggunakan syntax berikut:

```
Barang laptop = new Barang();
```

Ini akan membuat instance objek baru dengan nama "laptop" yang memiliki atribut dan metode yang telah ditentukan dalam class "Barang".

### 4. Jelaskan konsep encapsulation dalam pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan sistem informasi inventaris barang.

Konsep encapsulation dalam pemrograman berorientasi objek adalah praktik menyembunyikan detail internal dari suatu objek dan hanya memberikan akses terkontrol melalui metode atau fungsi yang telah ditentukan. Hal ini penting dalam pengembangan sistem informasi inventaris barang karena melindungi data inventaris dari akses yang tidak sah atau tidak semestinya. Dengan cara ini, data inventaris dapat diakses dan dimanipulasi hanya melalui metode yang telah ditentukan, sehingga meningkatkan keamanan dan integritas sistem.

5. Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.

Contoh atribut (variabel) dalam sistem informasi inventaris yang harus di-encapsulate adalah harga, jumlah, dan kode barang. Ini perlu di-encapsulate karena informasi ini adalah bagian dari inti dari sistem inventaris dan harus dijaga dari perubahan yang tidak sah. Dengan menerapkan encapsulation, kita dapat mengontrol akses ke atribut-atribut ini dan memastikan bahwa mereka hanya dapat diubah melalui metode yang telah ditentukan, sehingga menjaga konsistensi dan integritas data inventaris.

6. Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?

Relasi antara kelas dalam pemrograman berorientasi objek adalah cara kelas-kelas berinteraksi satu sama lain dalam suatu program. Ini mencakup bagaimana satu kelas dapat menggunakan atau mengandalkan kelas lain untuk mencapai tujuan tertentu, seperti pewarisan, asosiasi, atau penggunaan objek dari kelas lain.

7. Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?

Dalam sistem informasi inventaris barang, relasi antara kelas "Barang" dan kelas "Kategori" dapat digambarkan sebagai relasi asosiasi. Artinya, kelas "Barang" memiliki atribut atau variabel yang merujuk ke kelas "Kategori" untuk menentukan kategori atau jenis barang yang bersangkutan. Misalnya, kita dapat memiliki atribut "kategori" dalam kelas "Barang" yang merupakan objek dari kelas "Kategori". Dengan cara ini, setiap barang terkait dengan kategori tertentu dalam sistem inventaris.

8. Berdasarkan kasus sistem informasi inventaris barang, coba buat sebuah class sederhana beserta atribut dan metodenya yang menggambarkan suatu entitas dalam sistem tersebut (misalnya, class "Barang").

```
1  public class Barang {
2      // Attribute
3      private String kode;
4      private String nama;
5      private int jumlah;
6      private double harga;
7
8      // Method
9      public Barang(String kode, String nama, int jumlah, double harga) {
10         this.kode = kode;
11         this.nama = nama;
12         this.jumlah = jumlah;
13         this.harga = harga;
14     }
15
16     public String getKode() {
17         return kode;
18     }
19
20     public void setKode(String kode) {
21         this.kode = kode;
22     }
23
24     public String getNama() {
25         return nama;
26     }
27
28     public void setNama(String nama) {
29         this.nama = nama;
30     }
31
32     public int getJumlah() {
33         return jumlah;
34     }
35
36     public void setJumlah(int jumlah) {
37         this.jumlah = jumlah;
38     }
39
40     public double getHarga() {
41         return harga;
42     }
43
44     public void setHarga(double harga) {
45         this.harga = harga;
46     }
47 }
48
```

9. Bagaimana Anda akan menggunakan encapsulation untuk melindungi atribut-atribut dalam class tersebut?

Untuk menggunakan encapsulation dalam class "Barang," atribut-atribut seperti kode, nama, jumlah, dan harga telah dideklarasikan sebagai private. Ini berarti atribut-atribut ini tidak dapat diakses langsung dari luar class. Untuk mengakses atau mengubah nilai atribut tersebut, kita harus menggunakan metode get dan set yang telah ada. Contohnya, kita dapat menggunakan getKode() untuk mendapatkan nilai kode barang dan setKode(String kode) untuk mengubahnya. Dengan cara ini, kita dapat mengontrol akses dan memastikan validitas data.

10. Gambarkan hierarki class atau hubungan antar class yang mungkin ada dalam sistem informasi inventaris barang di jurusan Teknologi Informasi. Berikan contoh relasi antar class (misalnya, inheritance atau association) dalam konteks tersebut.

Dalam sistem informasi inventaris barang di jurusan Teknologi Informasi, hierarki class dan relasi antar class mungkin akan terlihat seperti ini:

- Class "Barang" bisa memiliki relasi asosiasi dengan class "Kategori," di mana setiap barang terkait dengan satu kategori.
- Class "Kategori" bisa memiliki class "SubKategori" sebagai subclass atau turunannya.
- Class "Peminjaman" bisa berhubungan dengan class "Barang" melalui asosiasi, yang menggambarkan kapan dan oleh siapa barang dipinjamkan.
- Class "User" bisa memiliki hubungan dengan class "Peminjaman" untuk melacak aktivitas peminjaman oleh pengguna tertentu.