

# **OBJECT ORIENTED PROGRAMMING QUIZ 1**



**Nama : Muhammad Nurfauzi Ikhsan**  
**NIM : 2241720076**  
**No. Absen : 18**

**STATE POLYTECHNIC OF MALANG  
2023**

### 1. Class dan Object :

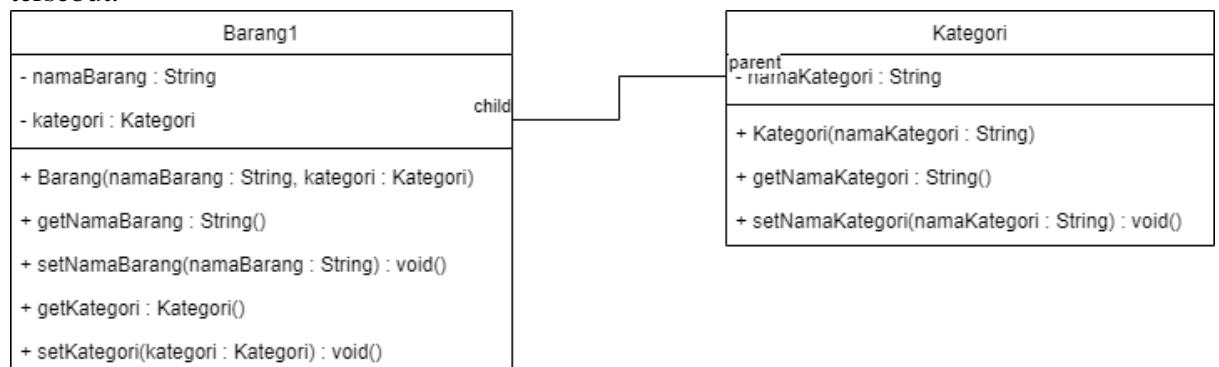
- Class adalah tempat yang digunakan untuk membuat objek dimana didalamnya terdapat atribut dan method.
- `Barang brg1 = new Barang()`
- `Brg1.namaBrg = "Laptop"`

### 2. Encapsulation :

- Konsep encapsulation adalah salah satu prinsip dalam pemrograman berorientasi objek yang mengacu pada pembungkusan atau pengapsulan data bersama metode yang beroperasi pada data tersebut dalam satu unit tunggal atau class. Dalam sistem inventaris barang, encapsulation memainkan peran penting dalam menjaga konsistensi dan keamanan data. Dengan membatasi akses langsung ke data inventaris, dapat dipastikan bahwa perubahan data hanya terjadi melalui metode-metode yang telah ditetapkan, memastikan bahwa data tetap akurat dan terkendali.
- Atribut yang harus di encapsulation salah satunya yaitu detail barang. Informasi spesifik mengenai setiap barang, seperti nama barang, deskripsi, harga, dan atribut lainnya, perlu di-encapsulate. Dengan cara ini, dapat mengontrol akses dan memastikan bahwa atribut-atribut ini hanya diakses atau diubah melalui metode-metode yang telah ditetapkan. Misalnya, hanya metode tertentu yang dapat mengubah harga barang.

### 3. Relasi Kelas :

- Dalam pemrograman berorientasi objek (OOP), relasi antara kelas mengacu pada hubungan yang dibentuk antara satu atau lebih kelas dalam suatu sistem. Hubungan ini mendefinisikan bagaimana satu kelas berinteraksi dengan kelas lainnya dan bagaimana informasi atau perilaku dapat berpindah antara kelas-kelas tersebut.



### 4. PBL :

```

1  public class Barang {
2      private String namaBarang;
3      private Kategori kategori;
4
5      public Barang(String namaBarang, Kategori kategori) {
6          this.namaBarang = namaBarang;
7          this.kategori = kategori;
8      }
9
10     public String getNamaBarang() {
11         return namaBarang;
12     }
13
14     public void setNamaBarang(String namaBarang) {
15         this.namaBarang = namaBarang;
16     }
17
18     public Kategori getKategori() {
19         return kategori;
20     }
21
22     public void setKategori(Kategori kategori) {
23         this.kategori = kategori;
24     }
25 }

```

- 
- Menggunakan prinsip akses private pada atribut
- Dalam sebuah sistem informasi inventaris barang di jurusan Teknologi Informasi, hierarki kelas atau hubungan antar kelas mungkin melibatkan beberapa entitas terkait seperti Barang, Kategori, Peminjaman, dan Mahasiswa. Berikut ini adalah contoh hubungan antara kelas-kelas tersebut:
  1. Barang
 

Atribut: namaBarang, jumlah, deskripsi, dll.

Hubungan:

    - Association dengan Kategori: Setiap barang terkait dengan satu kategori barang.
    - Association dengan Peminjaman: Barang-barang bisa dipinjam oleh mahasiswa.
  2. Kategori
 

Atribut: namaKategori, deskripsiKategori, dll.

Tidak memiliki hubungan langsung dengan kelas lain dalam hierarki ini, kecuali melalui hubungan dengan Barang.
  3. Peminjaman
 

Atribut: tanggalPinjam, tanggalKembali, dll.

Hubungan:

    - Association dengan Mahasiswa: Peminjaman dilakukan oleh mahasiswa.

- Association dengan Barang: Peminjaman melibatkan barang-barang tertentu.
4. Mahasiswa
- Atribut: nama, NIM, alamat, dll.
- Hubungan:
- Association dengan Peminjaman: Mahasiswa dapat melakukan peminjaman barang.

Contoh relasi antar class dalam konteks ini:

- a. Association: Barang memiliki asosiasi dengan Kategori dan Peminjaman. Peminjaman memiliki asosiasi dengan Mahasiswa dan Barang.
- b. Inheritance (Pewarisan): Tidak ada contoh yang jelas untuk inheritance dalam konteks ini, kecuali jika ada hierarki lebih lanjut seperti kelas turunan dari Barang yang lebih spesifik seperti Laptop, Proyektor, dll.