

Pemrograman Berorientasi Objek – Quiz 1

Nama: Muhammad Irsyad Dany

Kelas : 2C/16

Quiz 1

1. Class dan Object :

- Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?
 - Dalam pemrograman berorientasi objek (OOP), "class" adalah sebuah template untuk membuat objek. Class mendefinisikan variabel dan method pada seluruh objek tertentu, Class berisi atribut atau tipe data dan method untuk menjalankan suatu program
- Bagaimana Anda mendefinisikan objek dari suatu class dalam bahasa pemrograman Java?
 - Untuk mendefinisikan objek dari suatu class dalam bahasa pemrograman Java, kita perlu melakukan instansiasi atau pembuatan objek dari class tersebut. Instansiasi dilakukan dengan menggunakan kata kunci "new" diikuti dengan nama class dan tanda kurung.
- 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 dapat melakukan instansiasi objek dengan menggunakan kata kunci "new" diikuti dengan nama class "Barang" dan tanda kurung. Kemudian, Kita dapat mengatur atribut dari objek "laptop" tersebut dengan menggunakan method atau setter yang telah didefinisikan dalam class "Barang".
 - Contoh kode untuk membuat objek "laptop" dari class "Barang" adalah sebagai berikut :

```
Barang laptop = new Barang();  
laptop.setNama("Laptop Asus");  
laptop.setJumlah(10);  
laptop.setKondisi("Baik");
```
 - Dalam contoh di atas, "laptop" adalah nama objek yang dibuat, "Barang" adalah nama class yang digunakan untuk membuat objek, dan "new" adalah kata kunci yang digunakan untuk melakukan instansiasi objek. Kemudian, atribut dari objek "laptop" diatur menggunakan method "setNama", "setJumlah", dan "setKondisi" yang telah didefinisikan dalam class "Barang"

2. Encapsulation:

- Jelaskan konsep encapsulation dalam pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan sistem informasi inventaris barang.
 - Encapsulation adalah prinsip dalam pemrograman berorientasi objek yang berguna untuk menyembunyikan rincian implementasi dari objek dan melindungi kelas agar tidak mudah diakses secara sembarangan oleh program lain.
 - Dalam pengembangan sistem informasi inventaris barang, encapsulation sangat penting karena dapat membantu melindungi data inventaris dari akses yang tidak sah dan memudahkan pengelolaan data inventaris secara modular.
- Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.
 - Beberapa contoh atribut (variabel) yang harus di-encapsulate dalam sistem informasi inventaris barang antara lain:
 - Nama barang: atribut ini harus di-encapsulate agar tidak dapat diubah secara sembarangan oleh program lain dan hanya dapat diakses melalui antarmuka publik yang telah ditentukan.
 - Jumlah: atribut ini harus di-encapsulate agar tidak dapat diubah secara sembarangan oleh program lain dan hanya dapat diakses melalui antarmuka publik yang telah ditentukan.
 - Kategori: atribut ini harus di-encapsulate agar tidak dapat diubah secara sembarangan oleh program lain dan hanya dapat diakses melalui antarmuka publik yang telah ditentukan.
 - Harga: atribut ini harus di-encapsulate agar tidak dapat diubah secara sembarangan oleh program lain dan hanya dapat diakses melalui antarmuka publik yang telah ditentukan.
 - Status: atribut ini harus di-encapsulate agar tidak dapat diubah secara sembarangan oleh program lain dan hanya dapat diakses melalui antarmuka publik yang telah ditentukan.
 - Mengapa harus di encapsulate? Hal ini penting dilakukan untuk melindungi data inventaris dari akses yang tidak sah dan memudahkan pengelolaan data inventaris secara modular. Dengan encapsulation, atribut dari class "Barang" dapat disembunyikan dari program lain dan hanya dapat diakses melalui antarmuka publik yang telah ditentukan, sehingga data inventaris dapat terlindungi dengan baik.

3. Relasi Kelas :

- Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?
 - Dalam pemrograman berorientasi objek, relasi antara kelas adalah hubungan yang terjadi antara kelas satu dengan kelas lainnya. Relasi antara kelas digunakan untuk menggambarkan bagaimana kelas-kelas tersebut saling berhubungan dan bekerja sama dalam suatu sistem.
- Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?
 - Dalam sistem informasi inventaris barang, kelas "Barang" dan kelas "Kategori" memiliki relasi antara kelas yang disebut dengan relasi "Aggregation". Relasi ini menunjukkan bahwa sebuah objek dari kelas "Kategori" dapat terdiri dari beberapa objek dari kelas "Barang".
 - Dengan menggunakan relasi "Aggregation", data inventaris barang dapat dikelompokkan berdasarkan kategori atau jenis barang, sehingga memudahkan dalam pengelolaan dan pemeliharaan sistem informasi inventaris barang.

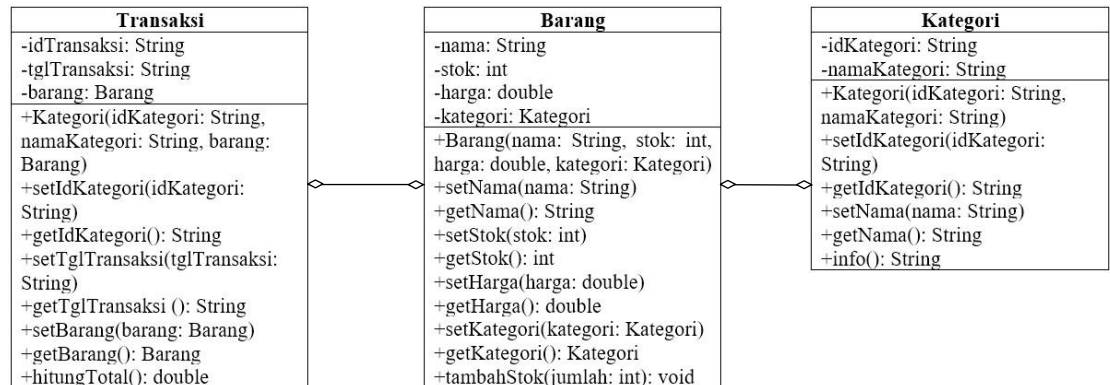
4. PBL :

- 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").
 - Class Barang

Barang
<ul style="list-style-type: none">➤ nama: String➤ stok: int➤ harga: double
<ul style="list-style-type: none">+ Barang(nama: String, stok: int, harga: double)+ setNama(nama: String)+ getNama(): String+ setStok(stok: int)+ getStok(): int+ setHarga(harga: double)+ getHarga(): double+ tambahStok(jumlah: int): void

- Bagaimana Anda akan menggunakan encapsulation untuk melindungi atribut-atribut dalam class tersebut?
 - Untuk menggunakan encapsulation dalam melindungi atribut dalam class "Barang" dalam sistem informasi inventaris barang, Anda dapat melakukan langkah-langkah berikut:

- Menyembunyikan atribut dari akses langsung dari program lain dengan menggunakan access modifier "private".
 - Memberikan akses ke atribut melalui method atau setter dan getter yang telah didefinisikan dalam class "Barang".
 - Memvalidasi nilai atribut yang dimasukkan melalui method atau setter untuk memastikan bahwa nilai tersebut valid dan sesuai dengan tipe data yang telah ditentukan.
- Dengan menggunakan encapsulation, atribut dari class "Barang" dapat terlindungi dari akses yang tidak sah dan hanya dapat diakses melalui method atau setter dan getter yang telah didefinisikan.
- 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.
 - Relasi antar Class Transaksi, Barang, dan Kategori



- a) Class "Transaksi": Mewakili transaksi jual beli barang dengan atribut seperti tanggal transaksi, jumlah, dan total harga.
 - b) Class "Barang": Mewakili barang dengan atribut nama, stok, dan harga.
 - c) Class "Kategori": Mewakili kategori barang dengan atribut seperti nama kategori.
- Contoh relasinya association yaitu setiap transaksi dalam class "Transaksi" dapat terkait dengan beberapa objek "Barang." Hal ini menciptakan relasi asosiasi antara class "Transaksi" dan class "Barang." Kemudian Setiap barang dapat terkait dengan satu atau lebih kategori.

