

QUIZ 1

Object Oriented Programming



**Arranged by :
Shofwah Kanaka Ebsa Anargya
2241720254 / 22
2I**

**INFORMATION TECHNOLOGY
D-IV INFORMATICS ENGINEERING
MALANG STATE POLYTECHNIC**

2023

Quiz 1

1. Class dan Object:

- Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?
- Bagaimana Anda mendefinisikan objek dari suatu class dalam bahasa pemrograman Java?
- Misalkan Anda memiliki class "Barang" dalam sistem informasi inventaris. Bagaimana Anda akan membuat objek "laptop" dari class tersebut?

Class dalam pemrograman berorientasi objek adalah blueprint atau cetak biru untuk menciptakan objek. Itu mendefinisikan atribut (variabel) dan perilaku (metode) yang akan dimiliki oleh objek.

Contohnya, jika terdapat class "Barang" dapat mendefinisikan atribut seperti "nama," "harga," dan "stok," serta metode seperti "tambahStok" atau "kurangiStok."

Untuk membuat objek "laptop" dari class "Barang," terdapat instansiasi, yaitu:

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

2. Encapsulation:

- Jelaskan konsep encapsulation dalam pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan sistem informasi inventaris barang.
- Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.

Encapsulation adalah konsep dalam pemrograman berorientasi objek di mana atribut (variabel) objek dibungkus bersama dengan metode yang mengoperasikannya. Ini penting karena melindungi data dari akses langsung dan modifikasi yang tidak sah.

Dalam sistem informasi inventaris barang, atribut seperti "harga," "stok," dan "kode barang" harus di-encapsulate. Sebagai contoh, atribut "harga" mungkin perlu memiliki pembatasan akses dan validasi untuk memastikan nilainya selalu positif.

3. Relasi Kelas:

- Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?
- Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?

Relasi antara kelas dalam pemrograman berorientasi objek menggambarkan hubungan antara objek dari kelas yang berbeda. Ini mencakup association (hubungan asosiasi), aggregation (komposisi), inheritance (pewarisan), dan dependency (ketergantungan).

Dalam sistem informasi inventaris barang, kelas "Barang" dan "Kategori" bisa memiliki relasi asosiasi di mana barang-barang tertentu terkait dengan kategori tertentu.

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

Contoh class Barang:

```
public class Barang {
    private String nama;
    private double harga;
    private int stok;
    private String kodeBarang;

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public double getHarga() {
        return harga;
    }

    public void setHarga(double harga) {
        if (harga > 0) {
            this.harga = harga;
        }
    }

    public int getStok() {
        return stok;
    }

    public void setStok(int stok) {
        if (stok >= 0) {
            this.stok = stok;
        }
    }

    public String getKodeBarang() {
        return kodeBarang;
    }

    public void setKodeBarang(String kodeBarang) {
        this.kodeBarang = kodeBarang;
    }
}
```

Hierarki class atau hubungan antar class dalam sistem inventaris barang bisa terdiri dari:

- Class "Kategori" mungkin berhubungan dengan class "Barang" dalam hubungan asosiasi.
- Class "Barang" bisa menjadi superclass yang diwariskan oleh class "BarangElektronik," "BarangPakaian," dll., jika ada spesialisasi barang.
- Class "Transaksi" bisa memiliki relasi komposisi dengan class "Barang," karena transaksi terdiri dari beberapa barang.

Contoh relasi asosiasi:

```
public class Kategori {  
    // ...  
}  
  
public class Barang {  
    // ...  
    private Kategori kategori;  
}
```

Dalam contoh di atas, class "Barang" dan "Kategori" memiliki hubungan asosiasi di mana setiap barang memiliki kategori tertentu.