

Quiz 1 OOP



Name: Steven Christian Susanto

NIM/Number: 2241720003/24

Class: 2I

Major: Information Technology

Study Program: D-IV Informatics Engineering

1. Class dan Object:

- Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?
 - Class adalah sebuah template yang digunakan untuk membentuk sebuah objek.
- Bagaimana Anda mendefinisikan objek dari suatu class dalam bahasa pemrograman Java?
 - `Nama_class nama_objek = new Nama_class(constructor);`
- Misalkan Anda memiliki class "Barang" dalam sistem informasi inventaris. Bagaimana Anda akan membuat objek "laptop" dari class tersebut?
 - `Barang laptop = new Barang();`

2. Encapsulation:

- Jelaskan konsep encapsulation dalam pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan sistem informasi inventaris barang.
 - Encapsulation disebut juga **information-hiding** yang berarti menyembunyikan informasi, semisal kita memiliki sistem inventaris barang, kita tentu tidak ingin informasi penting terkait barang barang tersebut diketahui atau bahkan dimanipulasi oleh pihak yang tidak memiliki kewenangan, disini encapsulation berfungsi untuk memastikan detail detail tersebut agar tidak bocor, dan agar tidak dapat dimanipulasi oleh sembarang orang.
- Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.
 - Atribut yang perlu di-encapsulate adalah seperti jumlah, nomor seri, harga, nama barang agar tidak dapat dimanipulasi dan disalahgunakan untuk korupsi.

3. Relasi Kelas:

- Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?
 - Suatu class yang saling memiliki keterkaitan antara class satu dengan yang lainnya.
- Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?
 - Agregasi (*has-a*)

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").

- ```
public class Barang {
 private String kodeBarang;
 private String namaBarang;
 private int jumlahBarang;

 public Barang(String kodeBarang, String namaBarang, int jumlahBarang) {
 this.kodeBarang = kodeBarang;
 this.namaBarang = namaBarang;
 this.jumlahBarang = jumlahBarang;
 }

 public String getKodeBarang() {
 return kodeBarang;
 }

 public String getNamaBarang() {
 return namaBarang;
 }

 public int getJumlahBarang() {
 return jumlahBarang;
 }

 public void setJumlahBarang(int jumlah) {
 this.jumlahBarang = jumlah;
 }
}
```

- Bagaimana Anda akan menggunakan encapsulation untuk melindungi atribut atribut dalam class tersebut?
  - Dalam program di atas, atribut kodeBarang, namaBarang, dan jumlahBarang dienkapsulasi dengan menggunakan akses modifier private. Hal ini membuat atribut-atribut tersebut tidak dapat diakses langsung dari luar class.
- 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
  - ```
public class BarangElektronik extends Barang {  
    private String jenis;
```

```

    public BarangElektronik(String kodeBarang, String namaBarang, int
jumlahBarang, String jenis) {
        super(kodeBarang, namaBarang, jumlahBarang);
        this.jenis = jenis;
    }

    public String getJenis() {
        return jenis;
    }
}

public class BarangNonElektronik extends Barang {
    private String kategori;

    public BarangNonElektronik(String kodeBarang, String namaBarang, int
jumlahBarang, String kategori) {
        super(kodeBarang, namaBarang, jumlahBarang);
        this.kategori = kategori;
    }

    public String getKategori() {
        return kategori;
    }
}

public class Peminjaman {
    private Barang barang;
    private String peminjam;
    private Date tanggalPeminjaman;

    public Peminjaman(Barang barang, String peminjam, Date
tanggalPeminjaman) {
        this.barang = barang;
        this.peminjam = peminjam;
        this.tanggalPeminjaman = tanggalPeminjaman;
    }

}

public class Pengembalian {
    private Barang barang;
    private String peminjam;
    private Date tanggalPengembalian;

```

```

        public Pengembalian(Barang barang, String peminjam, Date
tanggalPengembalian) {
            this.barang = barang;
            this.peminjam = peminjam;
            this.tanggalPengembalian = tanggalPengembalian;
        }
    }
}

```

```

public class Inventaris {
    private List<Barang> daftarBarang;

    public Inventaris() {
        daftarBarang = new ArrayList<>();
    }

    public void tambahBarang(Barang barang) {
        daftarBarang.add(barang);
    }

    public void hapusBarang(Barang barang) {
        daftarBarang.remove(barang);
    }

    public List<Barang> getDaftarBarang() {
        return daftarBarang;
    }
}

```

- Class BarangElektronik dan BarangNonElektronik merupakan subclass dari Barang, yang menggunakan kata kunci extends untuk mewarisi atribut dan metode dari class Barang.
- Class Peminjaman dan Pengembalian memiliki asosiasi dengan class Barang, yang berarti mereka menggunakan objek dari class Barang dalam proses peminjaman dan pengembalian. Dalam contoh ini, class Peminjaman dan Pengembalian memiliki atribut barang yang merupakan objek dari class Barang.
- Class Inventaris memiliki asosiasi dengan class Barang, yang berarti class Inventaris menggunakan objek dari class Barang dalam daftar inventaris barang. Dalam contoh ini, class Inventaris memiliki atribut daftarBarang yang merupakan List of Barang.