Quiz 1

1. Class dan Object:
   - Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?
     Class dalam pemrograman berorientasi objek adalah sebuah blueprint atau cetakan untuk menciptakan objek. Class mendefinisikan atribut (variabel) dan metode (fungsi) yang akan dimiliki oleh objek yang akan dibuat dari class tersebut.
   - 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?
     Untuk membuat objek dari suatu class dalam bahasa pemrograman Java adalah 1.Membuat sebuah class dengan mendefinisikan atribut dan metodenya. 2.Untuk membuat objek dari class tersebut, gunakan kata kunci "new" diikuti oleh nama class dan simpan objek tersebut dalam sebuah variabel.
     Contoh:

```
// Definisi class "Barang"
class Barang {
    String nama;
    int harga;
}
public class Main {
    public static void main(String[] args) {
        // Membuat objek "laptop" dari class "Barang"
        Barang laptop = new Barang();
        laptop.nama = "Laptop ASUS";
        laptop.harga = 800;
    }
}
```

2. Encapsulation:
   - Jelaskan konsep encapsulation dalam pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan sistem informasi inventaris barang.
     Encapsulation adalah konsep dalam pemrograman berorientasi objek yang mengacu pada pembungkusan (encapsulation) atribut dan method dalam suatu class, sehingga hanya mesthod dalam class tersebut yang dapat mengakses dan mengubah nilai atribut. Hal ini penting dalam pengembangan sistem informasi inventaris barang karena melindungi data atribut dari perubahan yang tidak diinginkan dan memungkinkan pengendalian yang lebih baik terhadap data tersebut.
   - Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.
     - Harga barang: Untuk mencegah perubahan harga secara langsung dari luar class.
     - Jumlah stok: Agar hanya metode dalam class yang dapat mengelola stok barang.
     - Kategori barang: Untuk memastikan kategori yang valid sebelum mengubahnya.
3. Relasi Kelas:
   - Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?
     Relasi antara kelas dalam pemrograman berorientasi objek menggambarkan bagaimana kelas-kelas saling berinteraksi atau berhubungan satu sama lain.

- Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?
  Dalam sistem informasi inventaris barang, kita dapat menggambarkan relasi antara kelas "Barang" dan "Kategori" dengan menggunakan asosiasi. Setiap objek barang dapat memiliki referensi ke objek kategori yang menggambarkan kategori barang tersebut. Ini akan memungkinkan kita untuk mengelompokkan barang-barang berdasarkan kategori.

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 nama;
    private int harga;
    private int stok;

    public Barang(String nama, int harga, int stok) {
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }

    public String getNama() {
        return nama;
    }

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

    public int getHarga() {
        return harga;
    }

    public void setHarga(int harga) {
        this.harga = harga;
    }

    public int getStok() {
        return stok;
    }

    public void setStok(int stok) {
        this.stok = stok;
    }
}
```

- Bagaimana Anda akan menggunakan encapsulation untuk melindungi atributatribut dalam class tersebut?
    - Private Atribut: Deklarasikan atribut sebagai private. Ini akan mencegah akses langsung dari luar class ke atribut tersebut.
    - Public Metode Getter: Buat metode-metode getter public untuk mengakses nilai atribut dari luar class. Getter memungkinkan penggunaan atribut tanpa mengaksesnya langsung.
    - Public Metode Setter: Jika ingin memungkinkan perubahan nilai atribut dari luar class, dapat membuat metode setter yang memvalidasi dan mengubah nilai atribut.
- 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.
    1. Class "Barang":
        - Class ini mewakili entitas barang dengan atribut seperti nama, stok, dan kategori.
        - Relasi antar class:
          Association dengan class "Kategori": Setiap barang termasuk dalam satu kategori.
    2. Class "Kategori":
        - Class ini mewakili entitas kategori barang dengan atribut seperti nama kategori.
        - Relasi antar class:
          Association dengan class "Barang": Setiap kategori memiliki banyak barang yang termasuk dalam kategori tersebut.
    3. Class "Transaksi":
        - Class ini mewakili entitas transaksi, mencatat peminjaman atau pengembalian barang dengan atribut seperti tanggal, jumlah, dan jenis transaksi (peminjaman / pengembalian).
        - Relasi antar class:
          Association dengan class "Barang": Transaksi mencakup barang-barang yang terlibat dalam transaksi.
          Agregasi/Composition dengan class "Pelanggan": Transaksi dapat berhubungan dengan pelanggan yang melakukan transaksi.
    4. Class "Mahasiswa":
        - Class ini mewakili entitas mahasiswa dengan atribut seperti nim, nama, dan nomor telepon.
        - Relasi antar class:
          Agregasi/Composition dengan class "Transaksi": Mahasiswa dapat melakukan banyak transaksi.
    5. Class "Laporan":
        - Class ini digunakan untuk menghasilkan laporan inventaris, seperti laporan stok barang, laporan peminjaman, dan laporan pengembalian.
        - Relasi antar class:
          Association dengan class "Barang": Laporan dapat memerlukan data dari class "Barang".
          Association dengan class "Transaksi": Laporan dapat memerlukan data transaksi untuk , laporan peminjaman dan laporan pengembalian.