

Nama : Renathan AAW

NIM : 2241720239/2I

1. Class dan object:

- Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?

Class adalah entitas yang mendefinisikan atribut (variabel) dan metode (fungsi) yang akan dimiliki oleh objek yang dibuat berdasarkan class tersebut. Konsep class memungkinkan kita untuk mengorganisasi dan mengelompokkan kode ke dalam unit yang lebih besar.

- Bagaimana Anda mendefinisikan objek dari suatu class dalam Bahasa pemrograman Java?

Pertama, Anda harus mendefinisikan class dengan menggunakan kata kunci "class". Contoh:

```
public class Barang {  
    // Atribut dan metode akan didefinisikan di sini.  
}
```

Selanjutnya, Anda dapat membuat objek dari class tersebut dengan menggunakan pernyataan "new". Contoh:

```
public class Barang {  
    // Atribut dan metode akan didefinisikan di sini.  
    Barang laptop = new Barang();  
}
```

- Misalkan Anda memiliki class "Barang" dalam sistem informasi inventaris. Bagaimana Anda akan membuat objek "laptop" dari class tersebut?

```
public class Barang {  
    String nama;  
    int jumlah;  
    public Barang(String nama, int jumlah) {  
        this.nama = nama;  
        this.jumlah = jumlah;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Barang laptop = new Barang("Laptop Acer", 5);  
    }  
}
```

2. Encapsulation:

- Jelaskan konsep encapsulation dalam pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan sistem informasi inventaris barang.

Dalam pemrograman, ini berarti bahwa atribut (variabel) dan metode (fungsi) yang terkait dengan suatu objek harus disimpan dalam suatu class dan dapat diakses melalui antarmuka yang telah ditentukan, yaitu metode yang didefinisikan dalam class tersebut.

1. Keamanan Data: Encapsulation memungkinkan Anda menghindari akses langsung ke fitur tertentu. Ini membantu mencegah perubahan data inventaris yang tidak sah atau tidak diinginkan, yang dapat mengganggu integritas dan konsistensi sistem, dengan menunjukkan bahwa data objek hanya dapat diubah dengan cara yang telah ditentukan.

2. Abstraksi: Encapsulation membuat antarmuka lebih mudah berinteraksi dengan objek. Pengguna sistem informasi inventaris hanya perlu tahu bagaimana menggunakan metode yang telah ditentukan untuk berinteraksi dengan objek, dan mereka tidak perlu mengetahui detail internal seperti struktur data dan algoritma.

3. Fleksibilitas: Encapsulation memungkinkan Anda mengubah implementasi internal kelas atau memperbaikinya tanpa mengganggu penggunaan kelas di bagian lain sistem. Ini memungkinkan peningkatan sistem tanpa mengganggu penggunaan objek yang sudah ada.

- Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.

Contoh atribut yang harus di-encapsulate dalam sistem informasi inventaris adalah:

1. Harga Barang: Harga barang adalah informasi sensitif yang harus dilindungi, jadi sebaiknya di-encapsulate. Dengan encapsulation, Anda dapat menerapkan validasi untuk memastikan bahwa harga selalu memiliki nilai yang tepat dan mengontrol siapa yang dapat mengakses atau mengubahnya.

2. Jumlah Stok: Fitur ini harus di-encapsulate karena perubahan jumlah stok barang harus dilakukan dengan hati-hati. Anda harus memastikan bahwa penambahan atau pengurangan stok hanya dilakukan dengan cara yang telah ditentukan untuk menghindari masalah penghitungan yang salah atau stok negatif.

3. Nama dan Deskripsi Barang: Untuk mencegah perubahan yang tidak sah atau tidak diinginkan, informasi ini juga harus di-encapsulate. Dengan memberikan metode akses yang tepat, Anda dapat memastikan bahwa data selalu konsisten.

3. Relasi Kelas:

- Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?

Relasi antara kelas dalam pemrograman berorientasi objek mengacu pada bagaimana dua atau lebih kelas berinteraksi atau terkait satu sama lain dalam suatu sistem.

- Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?

"Kategori" adalah kelas yang mewakili jenis atau kategori barang, seperti "Elektronik", "Pakaian", atau "Alat Dapur".

Barang adalah kelas yang mewakili barang individu yang ada dalam inventaris. Setiap barang harus terkait dengan satu kategori tertentu untuk mengelompokkannya ke dalam kategori yang relevan.

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 nama;
    private double harga;
    private int stok;
    public Barang(String kodeBarang, String nama, double harga,
int stok) {
        this.kodeBarang = kodeBarang;
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }
    public String getKodeBarang() {
        return kodeBarang;
    }
    public String getNama() {
        return nama;
    }
    public double getHarga() {
        return harga;
    }
    public int getStok() {
        return stok;
    }
    public void tambahStok(int jumlah) {
        stok += jumlah;
    }
    public void kurangiStok(int jumlah) {
        if (stok >= jumlah) {
            stok -= jumlah;
        } else {
            System.out.println("Stok tidak mencukupi.");
        }
    }
}
```

- Bagaimana Anda akan menggunakan encapsulation untuk melindungi atribut-atribut dalam class tersebut?
 - Atribut "kodeBarang," "nama," "harga," dan "stok" dideklarasikan sebagai private, sehingga hanya dapat diakses dari dalam class "Barang" itu sendiri. Ini melindungi atribut-atribut tersebut dari perubahan yang tidak sah dari luar class.
 - Metode-metode "getKodeBarang," "getNama," "getHarga," dan "getStok" digunakan untuk memberikan akses read-only ke atribut-atribut tersebut. Ini memungkinkan pengguna class untuk melihat nilainya tanpa dapat mengubahnya langsung.

- c. Metode-metode "tambahStok" dan "kurangiStok" digunakan untuk mengatur cara perubahan stok dilakukan, dan mereka memeriksa apakah perubahan tersebut valid sebelum mengubah nilai stok. Ini memastikan bahwa stok tidak dapat menjadi negatif.
- 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.
 - a. Class "Barang" yang mewakili barang individual dalam inventaris.
 - b. Class "KategoriBarang" yang dapat digunakan untuk mengelompokkan barang-barang ke dalam kategori tertentu.
 - c. Class "Transaksi" yang dapat digunakan untuk merekam transaksi penjualan atau pembelian barang.
 - d. Class "Pengguna" yang mewakili pengguna sistem yang dapat melakukan operasi terkait inventaris, seperti menambah, mengubah, atau menghapus barang.

Contoh relasi antar class:

- a. Class "Barang" dan "KategoriBarang" dapat memiliki relasi asosiasi di mana setiap barang terkait dengan satu kategori barang tertentu.
- b. Class "Transaksi" dapat memiliki relasi asosiasi dengan "Barang" untuk merekam barang-barang yang terlibat dalam transaksi tersebut.
- c. Class "Pengguna" dapat memiliki relasi asosiasi dengan "Barang" untuk mengidentifikasi operasi yang diizinkan oleh pengguna terhadap barang-barang dalam inventaris.