



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C
QUIS 1

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?

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.

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"?

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.

Jawaban :

1 a. kelas" adalah cetak biru atau cetak biru yang digunakan untuk membuat objek.

Misalnya, jika Anda ingin membuat program yang mengelola data pelanggan dalam suatu aplikasi, Anda dapat membuat kelas dengan nama "Pelanggan" yang mendefinisikan atribut seperti nama, alamat, nomor telepon, dan perilaku seperti metode menambah atau menghapus pelanggan.

b. Untuk mendefinisikan objek dari suatu kelas dalam bahasa pemrograman Java, meliputi pendefinisian kelas dan pembuatan objek.

c. dengan membuat code seperti ini

```

1  public class Barang {
2      // Atribut-atribut class "Barang"
3      String nama;
4      String merek;
5      double harga;
6
7      // Constructor untuk class "Barang"
8      public Barang(String nama, String merek, double harga) {
9          this.nama = nama;
10         this.merek = merek;
11         this.harga = harga;
12     }
13
14     // Metode untuk menampilkan informasi barang
15     public void displayInfo() {
16         System.out.println("Nama: " + nama);
17         System.out.println("Merek: " + merek);
18         System.out.println("Harga: $" + harga);
19     }
20 }
21

```

2

a. Relasi antara kelas mengacu pada hubungan atau koneksi yang dapat ada antara dua atau lebih kelas dalam sebuah program.

b. Dalam konteks sistem informasi inventaris, beberapa atribut (variabel) harus dienkapsulasi untuk menjamin keamanan data, abstraksi yang baik, dan pengendalian modifikasi data. Berikut adalah beberapa contoh properti yang harus dienkapsulasi dan alasannya:

- **Harga Barang (harga):** Harga barang adalah informasi sensitif yang sebaiknya di-encapsulate sebagai atribut privat.

- **Jumlah item (kuantitas):** Jumlah item dalam stok juga harus ditentukan.

Nama Barang (nama): Nama barang dapat di-encapsulate karena mungkin ada aturan-aturan bisnis terkait dengan format nama atau validasi lainnya.

- **Nama barang (Nama):**

Nama elemen dapat dienkapsulasi karena mungkin terdapat aturan bisnis mengenai pemformatan nama atau validasi lainnya.

- Dan lain-lain.

a. Jadi Hubungan kelas dalam pemrograman berorientasi objek (OOP) mengacu pada hubungan atau interaksi yang terjadi antara kelas yang berbeda dalam sistem berorientasi objek.

b. Biasanya akan dijelaskan dengan menggunakan relasi asosiasi atau relasi agregasi
Hubungan afiliasi:

Dalam hubungan asosiasi, kelas “Item” dan kelas “Kategori” merupakan dua kelas yang berdiri sendiri dan tidak sepenuhnya bergantung satu sama lain.

```
1 public class Barang {
2     private String nama;
3     private double harga;
4     private Kategori kategori;
5
6     // Konstruktor dan metode lain untuk Barang
7 }
8
9 public class Kategori {
10    private String nama;
11    private String deskripsi;
12
13    // Konstruktor dan metode lain untuk Kategori
14 }
```

Hubungan afiliasi:

Dalam hubungan asosiasi, kelas “Item” dan kelas “Kategori” merupakan dua kelas yang berdiri sendiri dan tidak sepenuhnya bergantung satu sama lain.

```
1 public class Kategori {
2     private String nama;
3     private String deskripsi;
4     private List<Barang> daftarBarang;
5
6     public Kategori(String nama, String deskripsi) {
7         this.nama = nama;
8         this.deskripsi = deskripsi;
9         this.daftarBarang = new ArrayList<>();
10    }
11
12    // Metode untuk menambah, menghapus, dan mengelola Barang dalam Kat
13 }
14
15 public class Barang {
16     private String nama;
17     private double harga;
18
19     // Konstruktor dan metode lain untuk Barang
20 }
```

4.

```
a. public class Barang {
    // Atribut atau variabel instance
    private String kodeBarang;
    private String nama;
    private double harga;
    private int stok;

    // Constructor (konstruktor) untuk inisialisasi objek Barang
    public Barang(String kodeBarang, String nama, double harga, int stok) {
        this.kodeBarang = kodeBarang;
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }

    // Metode untuk mendapatkan kode barang
    public String getKodeBarang() {
        return kodeBarang;
    }

    // Metode untuk mendapatkan nama barang
    public String getNama() {
        return nama;
    }

    // Metode untuk mendapatkan harga barang
    public double getHarga() {
        return harga;
    }

    // Metode untuk mengatur harga barang
    public void setHarga(double harga) {
        if (harga >= 0) {
            this.harga = harga;
        } else {
            System.out.println("Harga tidak valid.");
        }
    }

    // Metode untuk mendapatkan stok barang
    public int getStok() {
        return stok;
    }

    // Metode untuk menambah stok barang
    public void tambahStok(int jumlah) {
        if (jumlah > 0) {
            stok += jumlah;
        } else {
            System.out.println("Jumlah stok tidak valid.");
        }
    }
}
```

```
// Metode untuk mengurangi stok barang
public void kurangiStok(int jumlah) {
    if (jumlah > 0 && jumlah <= stok) {
        stok -= jumlah;
    } else {
        System.out.println("Jumlah stok tidak valid.");
    }
}
}
```

b.

Ubah properti menjadi pribadi:

Jadikan semua properti bersifat pribadi sehingga tidak dapat diakses langsung dari luar kelas.

Menggunakan metode pengakses (pengambil dan penyetel):

Gunakan metode pengakses (pengambil dan penyetel) untuk mengakses dan mengubah nilai properti. Metode pengambil digunakan untuk membaca nilai properti, sedangkan metode penyetel digunakan untuk memodifikasinya.

c.

Pertama buat class dulu lalu relasi :

Dalam hierarki class di atas, terdapat lima class, yaitu:

- Barang adalah class induk yang merupakan representasi dari semua barang inventaris.
- PeralatanKomputer adalah class turunan dari class Barang yang merupakan representasi dari peralatan komputer.
- AlatTulis adalah class turunan dari class Barang yang merupakan representasi dari alat tulis.
- BarangPribadi adalah class turunan dari class Barang yang merupakan representasi dari barang pribadi.
- Pengguna adalah class yang mewakili pengguna sistem informasi inventaris.
- Transaksi adalah class yang mewakili transaksi yang terjadi pada barang inventaris.

Berikut adalah contoh relasi antar class dalam konteks tersebut:

- Inheritance
 - PeralatanKomputer dan AlatTulis mewarisi semua atribut dan metode dari class Barang.
 - BarangPribadi mewarisi semua atribut dan metode dari class Barang, serta memiliki atribut tambahan berupa pemilik.

- Association
 - Barang memiliki asosiasi dengan Pengguna melalui atribut pemilik pada class BarangPribadi.
 - Barang memiliki asosiasi dengan Transaksi melalui atribut barang pada class Transaksi.

Berikut adalah penjelasan lebih lanjut mengenai relasi antar class tersebut:

- Inheritance
 - Inheritance digunakan untuk merepresentasikan hubungan antara class induk dan class turunan. Dalam konteks ini, class PeralatanKomputer dan AlatTulis adalah class turunan dari class Barang. Hal ini berarti bahwa PeralatanKomputer dan AlatTulis memiliki semua atribut dan metode yang dimiliki oleh class Barang.
 - Inheritance juga dapat digunakan untuk merepresentasikan hubungan antara class induk dan class turunan yang memiliki atribut tambahan. Dalam konteks ini, class BarangPribadi adalah class turunan dari class Barang yang memiliki atribut tambahan berupa pemilik.
- Association
 - Association digunakan untuk merepresentasikan hubungan antara dua class yang berbeda. Dalam konteks ini, Barang memiliki asosiasi dengan Pengguna melalui atribut pemilik pada class BarangPribadi. Hal ini berarti bahwa setiap BarangPribadi memiliki Pengguna yang menjadi pemiliknya.
 - Barang juga memiliki asosiasi dengan Transaksi melalui atribut barang pada class Transaksi. Hal ini berarti bahwa setiap Transaksi memiliki Barang yang menjadi objek transaksinya.