

Quiz 1

Nama : Chyntia Santi Nur Trisnawati

Kelas : 2C

Absen : 08 (Delapan)

1. Class dan Object

- Apa yang dimaksud dengan “class” dalam pemrograman berorientasi objek?

Jawab

Class dalam PBO merupakan blueprint (template) atau suatu tempat yang digunakan untuk membuat suatu objek. Contohnya Ketika Kita membuat sebuah class yang berisikan atribut dari mobil dan terdapat pula fungsi mobil di dalamnya. Maka class tersebut akan menjadi class objek untuk mobil.

- Bagaimana anda mendefinisikan objek dalam suatu class dalam Bahasa pemrograman java

Jawab

Dalam Bahasa pemrograman pendefinisian objek dalam suatu class dapat dilakukan dengan beberapa cara sebagai berikut:

1. Buatlah class terlebih dahulu. Kali ini saya akan membuat sebuah class Mahasiswa

```
Public class Mahasiswa{  
    String nama;  
    String Alamat;  
    String nim;  
}
```

2. Kemudian mulai tambahkan fungsi di dalamnya

```
public void biodata{  
    System.out.println(“Nama : ”);  
    System.out.println(“Alamat : ”);  
    System.out.println(“Nim : ”);  
}
```

3. Kemudian kita akan membuat objek dengan kode program seperti dibawah ini
Mohaswa mhs = new Mahasiswa();

Kemudian akses atribut dan methode

```
mhs.nama = “Reyhan Iqbal”;  
mhs.alamat = “Jln Sanagrejo”;  
mhs.nim = “223123123”;
```

4. Lalu panggil fungsi yang sudah dibuat pada main
mhs.biodata();

- Misalkan anda memiliki class “Barang” dalam system informasi investaris. Bagaimana anda akan membuat objek “Laptop” dari class tersebut?

Jawab

1. Yang perlu dilakukan yaitu membuat class barang terlebih dahulu dengan memasukkan atribut yang dimiliki oleh objek seperti berikut

```
Public class Barang{  
    private String nama;
```

```

private String jenis;
private String kode;
private int jumlah;
private double harga;
}

```

2. Kemudian buatlah konstruktor di dalam class yang sudah dibuat


```

Public Barang(String nama, String jenis, String kode, int jumlah, double harga){
    this.nama = nama;
    this.jenis = jenis;
    this.kode = kode;
    this.jumlah = jumlah;
    this.harga = harga;
}

```

3. Kemudian buatlah method get dan set pada masing masing atribut ini opsional ya
Saya akan memberikan salah satu contoh

```

public String getNama(){
    return nama;
}

```

```

Public void setNama(String nam){
    this.nama;
}

```

4. Kemudian mari kita membuat objek Laptop, sebelumnya kita membuat konstruktor terlebih dahulu seperti berikut:

```

Barang laptop = new Barang("Laptop Asus", "Tuf Gaming", "A9I97YG", 1,
9.999.999,99);

```

5. Kemudian pada main tulis kode program untuk menampilkan output program dengan kode program seperti di bawah ini:

```

System.out.println("Informasi Laptop:");
System.out.println("Nama: " + laptop.getNama());
System.out.println("Jenis: " + laptop.getJenis());
System.out.println("Kode Barang: " + laptop.getKode());
System.out.println("Jumlah: " + laptop.getJumlah());
System.out.println("Harga: $" + laptop.getHarga());

```

2. Encapsulation

- Jelaskan konsep encapsulation pemrograman berorientasi objek dan mengapa hal ini penting dalam pengembangan system investaris barang

Jawab

Konsep encapsulation adalah sebuah konsep yang terlihat seperti tabung. Dimana konsep ini dapat membungkus atribut dan metode dalam suatu class sehingga metode-metode yang ditentukan yang memiliki akses langsung ke atribut tersebut. Dalam hal ini terdapat tiga kata kunci yaitu, public, private dan protected.

Alasan mengapa Encapsulation penting yaitu:

1. Pengendalian Akses, Kita dapat mengendalikan akses menggunakan kata tiga kata kunci yang sudah disebutkan tadi. Hal ini berfungsi untuk mencegah terjadinya manipulasi yang tidak sah atau tidak terkendali

2. Abstraksi, Pengguna hanya tau cara untuk menggunakan class tanpa perlu tau bagaimana proses dalam class dikerjakan dan diimplementasi. Ini dapat meningkatkan abstraksi dan memudahkan dalam proses pemeliharaan kode di masa depan tanpa mempengaruhi pengguna class.
 3. Validasi Data, validasi data dapat dilakukan dengan menggunakan metode getter dan setter. Kita dapat memastikan nilai pada atribut valid sebelum mengubahnya.
 4. Fleksibilitas, Ketika sistem di ubah atau melakukan perbaikan maka dalam sistem inventaris tanpa memaksa pengguna class lain untuk memodifikasi kode.
 5. Keamanan, membantu untuk melindungi data dari modifikasi yang tidak sah atau tidak sengaja sehingga data lebih aman.
- Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa.
 - a. Harga: Karena merupakan atribut yang penting dalam proses penghitungan. Sehingga Ketika terjadi perubahan harga dapat dilakukan dengan metode yang tepat dan sudah tervalidasi. Contohnya harga tidak boleh negative
 - b. Tanggal Pembelian: Tanggal pembelian digunakan untuk melacak Riwayat barang sehingga Ketika tanggal tersebut diatur dengan benar maka mencegah terjadinya manipulasi yang tidak sah
 - c. Lokasi: Lokasi penyimpanan sangat penting sehingga dengan mengencapsulasi lokasi dapat memastikan bahwa barang yang dipindahkan atau diubah lokasinya sudah melalui prosedur yang benar
3. Relasi Kelas
- Apa yang dimaksud dengan relasi antar kelas dalam pemrograman berbasis objek?
Jawab
 Ketergantungan yang terjadi antara satu class dengan kelas lain dimana class tersebut saling berinteraksi satu sama lain dalam suatu objek
 - Dalam system informasi inventaris barang, bagaimana anda akan menggambarkan relasi antara kelas “Barang” dan “Kategori”?
Jawab
 - a. Asosiasi, relasi ini berarti setiap barang memiliki ikatan dengan satu kategori, dan setiap kategori memiliki banyak barang yang terkait. Kategori tersebut digunakan untuk mengelompokkan barang-barang yang serupa.
 - b. Akses ke Kategori, Class Barang dapat memiliki atribut yang bisa mengacu ke dalam kategori. Ini memudahkan untuk menentukan kita memilih kategori yang cocok untuk barang yang kita kelompokkan.
 - c. Mengakses barang dalam kategori, Dalam class kategori kita bisa memiliki atribut atau metode yang bisa mengizinkan kita mengakses semua barang yang masuk dalam kategori. Contohnya daftar barang yang ada pada kategori tersebut.
 - d. Perubahan dan Pembaruan, Ini dapat dilakukan dengan cara memanipulasi barang atau kategori sehingga dapat membantu membangun sebuah system inventaris yang efisien.
4. PBL
- Berdasarkan system kasus informasi inventaris barang, coba buat class sederhana beserta atribut dan metodenya yang menggambarkan suatu entitas dalam system tersebut (misalnya, class Barang)

Jawab

a. Membuat Class Kategori

public class Kategori {
//Atribut
private String jenisKategori;
private int kodeKategori;
private String deskripsiKategori;
//Konstruktor
public Kategori(String jenisKategori, String deskripsiKategori, int kodeKategori){
this.jenisKategori = jenisKategori;
this.deskripsiKategori = deskripsiKategori;
this.kodeKategori = kodeKategori;
}
//Metode Get & Set Jenis Kategori
public String getJenisKategori(){
return jenisKategori;
}
public void setJenisKategori(String jenisKategori){
this.jenisKategori = jenisKategori;
}
//Matode Get & Set Kode Kategori
public int getKodeKategori(){
return kodeKategori;
}
public void setKodeKategori(int kodeKategori){
this.kodeKategori = kodeKategori;
}
//Matode Get & Set Deskripsi Kategori
public String getDeskripsiKategori(){
return deskripsiKategori;
}
public void setDeskripsiKategori(String deskripsiKategori){
this.deskripsiKategori = deskripsiKategori;
}
}

b. Membuat Class Barang

public class Barang{
//Atribut
private String namaBarang;
private int kodeBarang;
private int jumlah;
private double harga;
private Kategori kategori;
//Constructor
public Barang(String namaBarang, int kodeBarang, int jumlah, double harga, Kategori kategori){
this.namaBarang = namaBarang;
this.kodeBarang = kodeBarang;
this.jumlah = jumlah;
this.harga = harga;
this.kategori = kategori;
}
//Method Get & Set Nama Barang
public String getNamaBarang(){
return namaBarang;
}
public void setNamaBarang(String namaBarang){
this.namaBarang = namaBarang;
}
//Method Get & Set Kode Barang
public int getKodeBarang(){
return kodeBarang;
}
public void setKodeBarang(int kodeBarang){
this.kodeBarang = kodeBarang;
}
//Method Get & Set Jumlah
public int getJumlah(){
return jumlah;
}
public void setJumlah(int jumlah){
this.jumlah = jumlah;
}
//Method Get & Set Harga
public double getHarga(){
return harga;
}
public void setHarga(double harga){

this.harga = harga;
}
//Method Get & Set Kategori
public Kategori getKategori(){
return kategori;
}
public void setKategori(Kategori kategori){
this.kategori = kategori;
}
//Metode Hitung Total
public double total(){
return harga*jumlah;
}
//Metode Tambah Stok Barang
public void tambahStok(int jumlahTambah){
this.jumlah += jumlahTambah;
}
//Metode Kurang Stok Barang
public void kurangStok(int jumlahKurang){
if (jumlahKurang <= this.jumlah){
this.jumlah -= jumlahKurang;
} else{
System.out.println("Maaf stok tidak mencukupi");
}
}
}

c. Class Main

import java.util.Scanner;
public class Hasil {
public static void main(String[] args) {
//Membuat Objek Kategori
Kategori makanan0 = new Kategori("Minuman", "Susu untuk anak usia 9 - 15 Tahun", 987);
Kategori makanan1 = new Kategori("Makanan", "Snak singkong rasa balado", 223);
Kategori makanan2 = new Kategori("Makanan", "Mie instan rasa sambal ijo", 663);
//Membuat Objek Barang
Barang barang0 = new Barang("Susu Ultramilk", 1198, 5, 25000, makanan0);

Barang barang1 = new Barang("Kusuka", 2928, 3, 8500, makanan1);
Barang barang2 = new Barang("Mie Goreng", 3214, 4, 2500, makanan2);
//Membuat Menu Tampilan
Scanner sc = new Scanner(System.in);
int menu;
System.out.println("-----");
System.out.println(" Selamat Datang ");
System.out.println(" Data Informasi Toko Sejahterah ");
System.out.println("-----");
System.out.println("Pilihan Menu : ");
System.out.println("1. Tampil Barang ");
System.out.println("2. Tambah Barang ");
System.out.println("3. Kurang Barang ");
System.err.print("Pilih Menu [1/2/3] : ");
menu = sc.nextInt();
//Pilihan Menu
if(menu == 1){
//Menampilkan Barang dan Kategori
System.out.println("Informasi Barang 1:");
System.out.println("Kode Barang : " + barang0.getKodeBarang());
System.out.println("Nama Barang : " + barang0.getNamaBarang());
System.out.println("Harga : Rp " + barang0.getHarga());
System.out.println("Jumlah : " + barang0.getJumlah());
System.out.println("Kategori : " + barang0.getKategori().getJenisKategori());
System.out.println("Deskripsi : " + barang0.getKategori().getDeskripsiKategori());
System.out.println(" ");
System.out.println("Informasi Barang 2:");
System.out.println("Kode Barang : " + barang1.getKodeBarang());
System.out.println("Nama Barang : " + barang1.getNamaBarang());
System.out.println("Harga : Rp " + barang1.getHarga());
System.out.println("Jumlah : " + barang1.getJumlah());
System.out.println("Kategori : " + barang1.getKategori().getJenisKategori());
System.out.println("Deskripsi : " + barang1.getKategori().getDeskripsiKategori());
System.out.println(" ");
System.out.println("Informasi Barang 3:");
System.out.println("Kode Barang : " + barang2.getKodeBarang());
System.out.println("Nama Barang : " + barang2.getNamaBarang());
System.out.println("Harga : Rp " + barang2.getHarga());
System.out.println("Jumlah : " + barang2.getJumlah());
System.out.println("Kategori : " + barang2.getKategori().getJenisKategori());
System.out.println("Deskripsi : " + barang2.getKategori().getDeskripsiKategori());
System.out.println(" ");

} else if(menu == 2){
//Tambah Barang
int kode;
int jumlah;
Scanner ab = new Scanner(System.in);
Scanner bc = new Scanner(System.in);
System.out.print("Masukkan Kode Barang : ");
kode = ab.nextInt();
System.out.println("Masukkan Jumlah Barang yang di tambahkan");
jumlah = bc.nextInt();
//Pemilihan barang
if(kode == barang0.getKodeBarang()){
barang0.tambahStok(jumlah);
System.out.println("Barang dengan kode " +kode + " berjumlah : " +
barang0.getJumlah());
} else if(kode == barang1.getKodeBarang()){
barang1.tambahStok(jumlah);
System.out.println("Barang dengan kode " +kode + " berjumlah : " +
barang0.getJumlah());
} else if (kode == barang2.getKodeBarang()){
barang2.tambahStok(jumlah);
System.out.println("Barang dengan kode " +kode + " berjumlah : " +
barang0.getJumlah());
}
} else if(menu == 3){
//Kurang Barang
int kode;
int jumlah;
Scanner ab = new Scanner(System.in);
Scanner bc = new Scanner(System.in);
System.out.print("Masukkan Kode Barang : ");
kode = ab.nextInt();
System.out.println("Masukkan Jumlah Barang yang di tambahkan");
jumlah = bc.nextInt();
//Pemilihan barang
if(kode == barang0.getKodeBarang()){
barang0.kurangStok(jumlah);
System.out.println("Barang dengan kode " +kode + " berjumlah : " +
barang0.getJumlah());
} else if(kode == barang1.getKodeBarang()){

barang1.kurangStok(jumlah);
System.out.println("Barang dengan kode " +kode + " berjumlah : " + barang0.getJumlah());
} else if (kode == barang2.getKodeBarang()){
barang2.kurangStok(jumlah);
System.out.println("Barang dengan kode " +kode + " berjumlah : " + barang0.getJumlah());
}
}
}
}

- Bagaimana anda akan menggunakan encapsulation untuk melindungi atribut-atribut dalam class tersebut

Jawab

Dengan melakukan modifier access pada kode program sehingga data yang masuk dapat disesuaikan dengan kebutuhan. Modifier access yang dimaksud antara lain public, private dan protected.

- 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.

Jawab

