

Quiz 1

@ Name	Davis Maulana Hermanto
🏠 Class	TI 2i
# NIM	2241720255
📖 Subject	Object Oriented Programming
📁 Type	Quiz
👤 Semester	Semester 3
📅 Time	@October 8, 2023

1. Class and Object

- Apa yang dimaksud dengan "class" dalam pemrograman berorientasi objek?
 - Class adalah sebuah pengelompokan kode dalam OOP yang memungkinkan pembuatan object dengan karakteristik tertentu.
- Bagaimana Anda mendefinisikan objek dari suatu class dalam bahasa pemrograman Java?
 - Pertama tama membuat class terlebih dahulu. contoh :

```

public class Mobil {
    private String merk;
    private int biaya;

    public Mobil(){
        this.merk = "kosong";
        this.biaya = 0;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setMerk(String merk) {
        this.merk = merk;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getMerk() {
        return merk;
    }

```

- Selanjutnya membuat objectnya

```
Mobil m = new Mobil();
```

- 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 adalah sebuah ide untuk menyembunyikan detail implementasi internal suatu objek dan hanya mengekspos fungsionalitas yang diperlukan
 - Mengapa penting?

- Dalam konteks pengembangan sistem informasi inventaris barang, ini dapat menciptakan antarmuka yang jelas untuk berinteraksi dengan objek inventaris barang dan pada saat yang sama melindungi data dan logika internalnya.
- Dalam konteks sistem informasi inventaris, sebutkan contoh atribut (variabel) yang harus di-encapsulate dan mengapa
 - Jumlah Barang
 - Agar kita dapat menetapkan aturan dan validasi khusus untuk memastikan bahwa jumlah barang selalu positif dan menghindari manipulasi yang tidak diinginkan.
 - Nilai Barang
 - Agar dapat membantu dalam memberlakukan kebijakan terkait harga dan nilai barang dan memastikan bahwa perubahan nilai dilakukan dengan cara yang terkontrol.
 - Status Barang
 - Agar dapat mengendalikan perubahan status sesuai dengan alur kerja inventaris dan mencegah perubahan status yang tidak sah.

3. Relasi Kelas

- Apa yang dimaksud dengan relasi antara kelas dalam pemrograman berorientasi objek?
 - Relasi antara kelas mengacu pada hubungan atau keterkaitan antara dua atau lebih kelas yang mencerminkan cara objek dari satu kelas berinteraksi dengan objek dari kelas lainnya.
- Dalam sistem informasi inventaris barang, bagaimana Anda akan menggambarkan relasi antara kelas "Barang" dan kelas "Kategori"?
 - Hubungan antara Item dan Kategori adalah asosiasi karena item sering dikategorikan ke dalam kategori yang berbeda dan mengelola inventaris secara efektif.

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").

```

package ac.relasticlass.quiz;

public class Barang {
    public String nama;
    public double harga;
    public int jumlah;

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

    Codeium: Refactor | Explain | Generate Javadoc
    public void tambahJumlah(int jumlah){
        this.jumlah += jumlah;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void kurangJumlah(int jumlah){
        if (jumlah < 0) {
            System.out.println(x:"Barang habis");
        }else{
            this.jumlah -= jumlah;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void print(){
        System.out.println("Nama Barang : " + nama);
        System.out.println("Harga Barang : " + harga);
        System.out.println("Jumlah Barang : " + jumlah);
    }
}

```

- Bagaimana Anda akan menggunakan encapsulation untuk melindungi atribut-atribut dalam class tersebut?

- Dengan mengubah modifier atribut dari public ke private dan menambahkan method setter getter untuk setiap atribut.

```
package ac.relasticlass.quiz;

public class Barang {
    private String nama;
    private double harga;
    private int jumlah;

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

    Codeium: Refactor | Explain | Generate Javadoc
    public void setNama(String nama){
        this.nama = nama;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getNama(){
        return nama;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setHarga(double harga){
        this.harga = harga;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public double getHarga(){
        return harga;
    }
}
```

```
Codeium: Refactor | Explain | Generate Javadoc
public void setJumlah(int jumlah){
    this.jumlah = jumlah;
}

Codeium: Refactor | Explain | Generate Javadoc
public int getJumlah(){
    return jumlah;
}

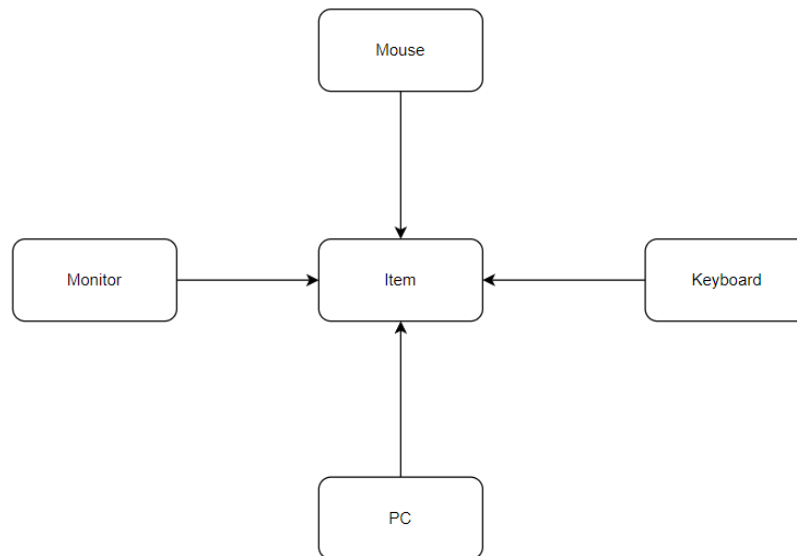
Codeium: Refactor | Explain | Generate Javadoc
public void tambahJumlah(int jumlah){
    this.jumlah += jumlah;
}

Codeium: Refactor | Explain | Generate Javadoc
public void kurangJumlah(int jumlah){
    if (jumlah < 0) {
        System.out.println("Barang habis");
    }else{
        this.jumlah -= jumlah;
    }
}

Codeium: Refactor | Explain | Generate Javadoc
public void print(){
    System.out.println("Nama Barang : " + nama);
    System.out.println("Harga Barang : " + harga);
    System.out.println("Jumlah Barang : " + jumlah);
}
}
```

- 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

- Hierarki Class



- Class Item

```
package ac.relasticlass.quiz;

public class Item {
    private String nama;
    private double harga;
    private int jumlah;
    private PC pc;
    private Monitor monitor;
    private Keyboard keyboard;
    private Mouse mouse;

    public Item(){

    }

    public Item(String nama, double harga, int jumlah, PC pc, Monitor monitor, Keyboard keyboard, Mouse mouse){
        this.nama = nama;
        this.harga = harga;
        this.jumlah = jumlah;
        this.pc = pc;
        this.monitor = monitor;
        this.keyboard = keyboard;
        this.mouse = mouse;
    }
}
```


Codeium: Refactor | Explain | Generate Javadoc
`public void setNama(String nama){
 this.nama = nama;
}`

Codeium: Refactor | Explain | Generate Javadoc
`public String getNama(){
 return nama;
}`

Codeium: Refactor | Explain | Generate Javadoc
`public void setHarga(double harga){
 this.harga = harga;
}`

Codeium: Refactor | Explain | Generate Javadoc
`public double getHarga(){
 return harga;
}`

Codeium: Refactor | Explain | Generate Javadoc
`public void setJumlah(int jumlah){
 this.jumlah = jumlah;
}`

Codeium: Refactor | Explain | Generate Javadoc

```
public int getJumlah(){  
    return jumlah;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public void tambahJumlah(int jumlah){  
    this.jumlah += jumlah;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public void kurangJumlah(int jumlah){  
    if (jumlah < 0) {  
        System.out.println(x:"Barang habis");  
    }else{  
        this.jumlah -= jumlah;  
    }  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public void print(){  
    System.out.println("Nama Barang : " + nama);  
    System.out.println("Harga Barang : " + harga);  
    System.out.println("Jumlah Barang : " + jumlah);  
    pc.print();  
    monitor.print();  
    keyboard.print();  
    mouse.print();  
}  
}
```

- Class PC

```

package ac.relasticlass.quiz;

public class PC {
    private String cpu;
    private int ram;
    private int storage;

    public PC(){

    }

    public PC(String cpu, int ram, int storage) {
        this.cpu = cpu;
        this.ram = ram;
        this.storage = storage;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setCpu(String cpu) {
        this.cpu = cpu;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getCpu() {
        return cpu;
    }

```

```

Codeium: Refactor | Explain | Generate Javadoc
public void setRam(int ram) {
    this.ram = ram;
}

Codeium: Refactor | Explain | Generate Javadoc
public int getRam() {
    return ram;
}

Codeium: Refactor | Explain | Generate Javadoc
public void setStorage(int storage) {
    this.storage = storage;
}

Codeium: Refactor | Explain | Generate Javadoc
public int getStorage() {
    return storage;
}

Codeium: Refactor | Explain | Generate Javadoc
public void print() {
    System.out.println("CPU : " + cpu);
    System.out.println("RAM : " + ram);
    System.out.println("Storage : " + storage);
}

```

- Class Monitor

```

package ac.relasiclass.quiz;

public class Monitor {
    private String nama;
    private int hz;
    private String resolusi;

    public Monitor(){

    }

    public Monitor(String nama, int hz, String resolusi){
        this.nama = nama;
        this.hz = hz;
        this.resolusi = resolusi;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setNama(String nama){
        this.nama = nama;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getNama(){
        return nama;
    }

```

Codeium: Refactor | Explain | Generate Javadoc

```
public void setHz(int hz){  
    this.hz = hz;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public int getHz(){  
    return hz;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public void setResolusi(String resolusi){  
    this.resolusi = resolusi;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public String getResolusi(){  
    return resolusi;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public void print(){  
    System.out.println("Monitor : " + nama);  
    System.out.println("Hz : " + hz);  
    System.out.println("Resolusi : " + resolusi);  
}
```

```
}
```

- Class Keyboard

```

package ac.relasticlass.quiz;

public class Keyboard {
    private String nama;
    private String jenis;

    public Keyboard(){

    }

    public Keyboard(String nama, String jenis){
        this.nama = nama;
        this.jenis = jenis;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setNama(String nama) {
        this.nama = nama;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getNama() {
        return nama;
    }

```

Codeium: Refactor | Explain | Generate Javadoc

```
public void setJenis(String jenis) {  
    this.jenis = jenis;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public String getJenis() {  
    return jenis;  
}
```

Codeium: Refactor | Explain | Generate Javadoc

```
public void print(){  
    System.out.println("Nama Keyboard : " + nama);  
    System.out.println("Jenis Keyboard : " + jenis);  
}
```

```
}
```

- Class Mouse


```

package ac.relasticlass.quiz;

public class Mouse {
    private String nama;
    private String jenis;

    public Mouse(){

    }

    public Mouse(String nama, String jenis){
        this.nama = nama;
        this.jenis = jenis;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setName(String nama){
        this.nama = nama;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public String getName(){
        return nama;
    }

    Codeium: Refactor | Explain | Generate Javadoc
    public void setJenis(String jenis){
        this.jenis = jenis;
    }

```

```
Codeium: Refactor | Explain | Generate Javadoc
public String getJenis(){
    return jenis;
}

Codeium: Refactor | Explain | Generate Javadoc
public void print(){
    System.out.println("Nama Mouse : " + nama);
    System.out.println("Jenis Mouse : " + jenis);
}
}
```

- Class Main

```
package ac.relasticlass.quiz;

public class Main {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc
    public static void main(String[] args) {
        PC pc = new PC();
        Monitor monitor = new Monitor();
        Keyboard keyboard = new Keyboard();
        Mouse mouse = new Mouse();

        pc.setCpu(cpu:"intel i9");
        pc.setRam(ram:16);
        pc.setStorage(storage:512);
        monitor.setNama(nama:"KOORUI 24E4");
        monitor.setHz(hz:165);
        monitor.setResolusi(resolusi:"1920 x 1080");
        keyboard.setNama(nama:"Logitech");
        keyboard.setJenis(jenis:"Wireless");
        mouse.setNama(nama:"Logitech");
        mouse.setJenis(jenis:"Wireless");

        Item item = new Item(nama:"PC Lab 1", harga:9000000, jumlah:30, pc, monitor, keyboard, mouse);

        item.print();
    }
}
```

- Hasil

```
Nama Barang : PC Lab 1
Harga Barang : 9000000.0
Jumlah Barang : 30
CPU : intel i9
RAM : 16
Storage : 512
Monitor : KOORUI 24E4
Hz : 165
Resolusi : 1920 x 1080
Nama Keyboard : Logitech
Jenis Keyboard : Wireless
Nama Mouse : Logitech
Jenis Mouse : Wireless
PS E:\document\POLINEMA\Sem 3\OOP prak\week 6\Relasi Class>
```