

Nama : D. Ihsan Maulana

Nim : 20220040069

Kelas : TI 22 C

TUGAS PEMROGRAMAN BERBASIS OBJEK

SESI 2

Catatan :

1. Jelaskan apa yang dimaksud dengan class dan object!
2. Sebutkan dan jelaskan jenis-jenis method !
3. Berdasarkan gambar berikut ini, jelaskan masing-masing bagian sesuai dengan nomor yang ada!

```
public class Komputer {  
    String jenis_komputer;  
    private String merk;  
  
    public void setDataKomputer(String jenis, String merk){  
        jenis_komputer = jenis;  
        this.merk = merk;  
    }  
  
    public String getJenis(){  
        return jenis_komputer;  
    }  
  
    public String getMerk(){  
        return merk;  
    }  
  
    public static void main(String[] args){  
        Komputer mykom = new Komputer();  
        mykom.setDataKomputer("LAPTOP", "MACBOOK");  
        System.out.println(mykom.getJenis());  
        System.out.println(mykom.getMerk());  
    }  
}
```

The image shows a Java code snippet for a class named 'Komputer'. The code is annotated with red boxes and numbers 1 through 8. Annotation 1 points to the class declaration 'public class Komputer'. Annotation 2 points to the private field 'private String merk;'. Annotation 3 points to the 'setDataKomputer' method. Annotation 4 points to the 'getJenis' method. Annotation 5 points to the 'getMerk' method. Annotation 6 points to the object creation 'Komputer mykom = new Komputer();'. Annotation 7 points to the method call 'mykom.setDataKomputer("LAPTOP", "MACBOOK");'. Annotation 8 points to the method call 'System.out.println(mykom.getMerk());'.

4. Tuliskan perbaikan pada kode program berikut ini, agar kode program tersebut dapat dicompile sehingga program akan berjalan dengan benar!

```
2 public class HandPhone {  
3     String jenis_hp;  
4     int tahun_pembuatan;  
5  
6     String public setDataHP(String jenis_hp, int tahun_pembuatan){  
7         jenis_hp = jenis_hp;  
8         tahun_pembuatan = tahun_pembuatan;  
9     }  
10  
11     String getJenisHP(){  
12  
13     }  
14  
15     String getTahunPembuatan(){  
16  
17     }  
18  
19     public static main void(String args[]){  
20         HandPhone hp = new HandPhone();  
21         hp.setDataHP(jenis_hp, tahun_pembuatan);  
22         hp.getJenisHP()  
23         hp.getTahunPembuatan()  
24     }  
25 }
```

The image shows a Java code snippet for a class named 'HandPhone'. The code contains several syntax errors. Line 6 has 'String public' instead of 'public String'. Line 12 has a closing brace '}' without an opening brace. Line 22 has 'hp.getJenisHP()' without a semicolon. Line 23 has 'hp.getTahunPembuatan()' without a semicolon. Line 19 has 'public static main void' instead of 'public static void main'.

Jawaban :

1. **“class”** adalah sebuah blueprint atau cetak biru untuk membuat objek. Class mendefinisikan properti (atribut) dan perilaku (metode) yang akan dimiliki oleh objek yang dibuat berdasarkan blueprint tersebut.

Contoh :

jika kita memiliki sebuah class "Mobil", maka class tersebut akan mendefinisikan atribut-atribut seperti "warna", "merek", "tahun pembuatan", dan metode-metode seperti "maju", "mundur", atau "berhenti".

✚ **“object”** adalah instance atau instansi konkret dari sebuah class. Ketika kita membuat objek dari sebuah class, kita menciptakan sebuah salinan dari blueprint tersebut dengan nilai-nilai yang spesifik untuk atribut-atributnya.

Coba kita gunakan contoh di atas, jika kita membuat sebuah objek mobil dengan warna merah, merek Toyota, dan tahun pembuatan 2020, maka objek tersebut adalah sebuah instansi dari class "Mobil". Kita bisa memiliki banyak objek mobil dengan warna, merek, dan tahun pembuatan yang berbeda, namun semuanya dibuat berdasarkan class "Mobil" yang sama.

Class dan object merupakan konsep fundamental dalam pemrograman berorientasi objek. Class mendefinisikan struktur dan perilaku suatu objek, sedangkan object adalah instance nyata dari class. Penggunaan class dan object membantu programmer untuk membuat program yang lebih modular, reusable, dan scalable.

2. Dalam pemrograman berorientasi objek, terdapat beberapa jenis metode yang sering digunakan dalam sebuah class. Berikut adalah beberapa jenis metode umum:

a. Metode Instance

- Metode ini adalah metode yang beroperasi pada instance dari sebuah class.
- Metode ini memiliki parameter pertama yang biasanya disebut self, yang merujuk pada instance objek itu sendiri.
- Metode ini dapat mengakses dan memanipulasi atribut-atribut dari objek yang sedang diproses.

Contoh :

Metode untuk mengubah nilai atribut objek, metode untuk menghitung atau memanipulasi data di dalam objek.

b. Metode Kelas (Class Methods)

- Metode ini adalah metode yang beroperasi pada level class, bukan instance objek individu.
- Metode ini diidentifikasi dengan dekorator `@classmethod`.
- Metode ini memiliki parameter pertama yang biasanya disebut `cls`, yang merujuk pada class itu sendiri.
- Metode ini dapat digunakan untuk melakukan operasi pada level class, seperti mengakses atau memanipulasi atribut-atribut kelas.

Contoh :

Metode untuk menghitung statistik dari seluruh instance yang dibuat dari sebuah class.

c. Metode Statis (Static Methods)

- Metode ini adalah metode yang tidak terkait dengan instance objek maupun class, namun masih terkait dengan konteks dari class tersebut.
- Metode ini diidentifikasi dengan dekorator `@staticmethod`.
- Metode ini tidak memiliki parameter khusus seperti `self` atau `cls`, sehingga tidak dapat mengakses atribut atau metode dari class atau instance objek.
- Metode ini sering digunakan untuk menyediakan fungsionalitas yang berkaitan dengan class namun tidak bergantung pada state instance atau class.

Contoh :

Metode untuk melakukan operasi matematika sederhana, fungsi bantuan yang digunakan dalam class.

d. Metode Khusus (Special Methods atau Magic Methods)

- Metode-metode ini memiliki nama yang diawali dan diakhiri dengan dua garis bawah (double underscore), seperti `__init__`, `__str__`, `__repr__`, dll.
- Metode-metode ini memberikan perilaku spesial kepada objek yang berkaitan dengan operasi-operasi khusus dalam bahasa Python.

Contoh :

`__init__` digunakan untuk inisialisasi objek, `__str__` digunakan untuk mengonversi objek menjadi string, `__eq__` digunakan untuk mendefinisikan logika kesetaraan objek, dll.

3. Berdasarkan gambar berikut ini, jelaskan masing-masing bagian sesuai dengan nomor yang ada !

```
public class Komputer { - 1
    String jenis_komputer; - 2
    private String merk;

    public void setDataKomputer(String jenis, String merk){
        jenis_komputer = jenis;
        this.merk = merk;
    } - 3

    public String getJenis(){
        return jenis_komputer; - 4
    }

    public String getMerk(){
        return merk; - 5
    }

    public static void main(String[] args){
        Komputer mykom = new Komputer(); - 6
        mykom.setDataKomputer("LAPTOP", "MALBUCK"); - 7
        System.out.println(mykom.getJenis());
        System.out.println(mykom.getMerk()); - 8
    }
}
```

- a. Nama Class
 - b. Attribute
 - c. Method Declaration (Setter)
 - d. Method Declaration (Getter)
 - e. Method Declaration (Getter)
 - f. Instantiante
 - g. Method setter (prosedur)
 - h. Method getter (fungsi)
- Method
- Object

4. “Mohon izin pak untuk Scriptnya saya sertakan di GitHub”