

Nama : Haikal Muhammad Kurniawan
Nim : 20220040008
Kelas : TI22M

1. Jelaskan apa yang dimaksud dengan class dan object!

Class adalah sebuah blueprint atau template untuk menciptakan objek. Class mendefinisikan atribut (variabel) dan metode (fungsi) yang akan dimiliki oleh objek yang dibuat berdasarkan class tersebut. Class merupakan struktur dasar yang digunakan untuk menciptakan objek dalam pemrograman berorientasi objek.

Object adalah instance atau perwujudan dari sebuah class. Ketika sebuah class diinstansiasi, maka akan tercipta objek yang memiliki atribut dan perilaku sesuai dengan definisi class tersebut. Objek merupakan entitas konkret yang dapat digunakan dalam program untuk melakukan berbagai tugas sesuai dengan fungsionalitas yang telah didefinisikan dalam classnya. Dengan menggunakan class dan object, kita dapat mengorganisir dan mengelola kode program secara lebih terstruktur dan efisien.

2. Sebutkan dan jelaskan jenis-jenis method !

Method adalah blok kode yang disebut juga fungsi atau prosedur yang terdefinisi didalam class pada pemrograman berorientasi objek. Method ini menyediakan cara tertentu untuk interaksi dengan objek yang dibuat dari class tersebut. Berikut adalah beberapa jenis utama dari method yang sering digunakan:

1. **Method Konstruktor (Constructor):** Method ini digunakan untuk membantu membuat objek baru dengan memberikan nilai awal bagi atribut-atribut yang diperlukan. Nama constructor sama dengan nama class tersebut tanpa menggunakan tanda kurung (). Constructor tidak perlu memiliki return type spesifik, namun biasanya tidak mengembalikan nilai kepada pemanggilannya.
2. **Method Getter (Accessor):** Method getter digunakan untuk mengambil nilai suatu atribut. Ini sangat penting agar atribut-atribut dapat dinilai dari luar class. Nama method getter umumnya mulai dengan huruf kapital ("get") followed by the attribute's name.
3. **Method Setter (Mutator):** Method setter digunakan untuk mengubah nilai suatu atribut. Ini juga sangat penting agar atribut-atribut dapat dirubah dari luar class. Nama method setter umumnya mulai dengan huruf kapital ("set") followed by the attribute's name.
4. **Method Aksesori (Property):** Property adalah kombinasi antara method getter dan setter. Property memungkinkan kita untuk mengakses dan

mengubah nilai atribut dengan cara yang lebih singkat dan lebih mirip dengan operasi langsung.

5. **Method Utility** : Method utility adalah method yang digunakan untuk melakukan tugas tersendiri tanpa harus memiliki hubungan langsung dengan objek. Biasanya, method utility ini bukanlah method yang hanya bisa dipanggil dari dalam class tersebut saja, tapi juga dapat dipanggil dari luar class tersebut.
6. **Method Overloading**: Method overloading adalah teknik yang digunakan untuk memiliki beberapa versi dari method dengan nama yang sama, namun dengan parameter berbeda. Pemilihan versi mana yang akan dieksekusi dilakukan oleh compiler berdasarkan tipe data dan jumlah argument yang diterima.
7. **Method Virtual**: Method virtual adalah method yang dapat dioverride oleh subclass. Hal ini sangat berguna untuk mengimplementasikan polymorphism di dalam pemrograman berorientasi objek.
8. **Method Static**: Method static adalah method yang tidak memerlukan instansiasi objek untuk dapat digunakan. Method ini dapat dianggap sebagai method yang "global" dalam class tersebut.
9. **Method Iterator**: Method iterator adalah method yang digunakan untuk mengatur iterasi pada objek. Method ini memungkinkan objek tersebut dapat dioperasikan secara iteratif dengan loop for atau fungsi built-in seperti list comprehension.

Sebagian besar bahasa pemrograman berorientasi objek memiliki semua atau beberapa dari jenis-jenis method ini. Selain itu, ada baiknya untuk mempelajari dokumentasi resmi bahasa pemrograman yang Anda gunakan untuk informasi tentang fitur tambahan yang tersedia dalam bahasa tersebut.

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

1. ****Deklarasi Kelas****:

```
public class Komputer {
```

Ini adalah deklarasi kelas Java yang disebut `Komputer`.

2. ****Deklarasi Variabel Kelas****:

```
String jenis_komputer;  
private String merk;
```

Di sini, dua variabel kelas dideklarasikan: `jenis_komputer` (dengan tipe data `String`) dan `merk` (dengan tipe data `String` juga). Variabel `merk` diberi akses modifier `private`, yang berarti hanya dapat diakses di dalam kelas tersebut.

3. ****Metode untuk Menetapkan Data Komputer****:

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

Ini adalah metode `setDataKomputer` yang menerima dua parameter (`jenis` dan `merk`) dan digunakan untuk menetapkan nilai `jenis_komputer` dan `merk` berdasarkan nilai parameter yang diterima.

4. ****Metode untuk Mendapatkan Jenis Komputer****:

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

Ini adalah metode `getJenis` yang digunakan untuk mengembalikan nilai dari variabel `jenis_komputer`.

5. ****Metode untuk Mendapatkan Merk Komputer****:

```
public String getMerk() {  
    return merk;  
}
```

Ini adalah metode `getMerk` yang digunakan untuk mengembalikan nilai dari variabel `merk`.

6. ****Pembuatan Objek Kelas Komputer****:

```
Komputer mykom = new Komputer();
```

Ini adalah membuat objek dari kelas `Komputer` dengan nama `mykom`.

7. ****Pemanggilan Metode `setDataKomputer`**:**

```
mykom.setDataKomputer("FIAFTOP", "MACBOOK");
```

Ini memanggil metode `setDataKomputer` dari objek `mykom` untuk menetapkan jenis dan merk komputer.

8. ****Pemanggilan Metode `getJenis` dan `getMerk`**:**

```
System.out.println(mykom.getJenis());  
System.out.println(mykom.getMerk());
```

Ini memanggil metode `getJenis` dan `getMerk` dari objek `mykom` untuk mendapatkan nilai jenis dan merk komputer, lalu mencetaknya ke layar.

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



```
Main.java  
1- public class HandPhone {  
2-     String jenis_hp;  
3-     int tahun_pembuatan;  
4-  
5-     public void setDataHP(String jenis_hp, int tahun_pembuatan) {  
6-         this.jenis_hp = jenis_hp;  
7-         this.tahun_pembuatan = tahun_pembuatan;  
8-     }  
9-  
10-    public String getJenisHP() {  
11-        return jenis_hp;  
12-    }  
13-  
14-    public int getTahunPembuatan() {  
15-        return tahun_pembuatan;  
16-    }  
17-  
18-    public static void main(String[] args) {  
19-        HandPhone hp = new HandPhone();  
20-        hp.setDataHP("jenis_hp", 2022);  
21-        System.out.println("Jenis HP: " + hp.getJenisHP());  
22-        System.out.println("Tahun Pembuatan: " + hp.getTahunPembuatan());  
23-    }  
24- }  
25- }
```

Output

```
java -cp /tmp/qNddqfnZU9 HandPhone  
Jenis HP: jenis_hp  
Tahun Pembuatan: 2022  
|
```

Perbaikan-perbaikan yang dilakukan adalah:

1. Metode '**setDataHP**': Perbaikan tanda kurung kurawal menjadi kurung kurawal biasa dan penambahan '**this**' untuk mengarahkan variabel instance kelas.
2. Metode '**getJenisHP**' dan '**getTahunPembuatan**': Penambahan tipe data pengembalian ('**String**' untuk jenis HP dan '**int**' untuk tahun pembuatan) dan penambahan return statement untuk mengembalikan nilai yang sesuai.
3. Penambahan tipe data '**String**' pada parameter metode '**main**'.
4. Memperbaiki pemanggilan metode '**setDataHP**' dan menambahkan output untuk hasil pemanggilan metode '**getJenisHP**' dan '**getTahunPembuatan**' agar tercetak ke layar.