

TUGAS

Nama : MUHAMAD FADHLI NURDIANSYAH RANGKUTI
Kelas : TI 22 C
NIM : 20220040259
Mata kuliah : Pemrograman Berorientasi Objek

apa yang dimaksud dengan class dan object!

Class

Dalam pemrograman berorientasi objek (Object Oriented Programming), class dapat didefinisikan sebagai blueprint atau rancangan dari sebuah objek. Class menjelaskan apa saja property dan method yang dimiliki oleh sebuah objek. Class ini tidak memiliki nilai atau data sendiri, melainkan hanya menjelaskan tipe-tipe data dan tindakan yang dapat dilakukan oleh sebuah objek.

Object

Sedangkan object dapat didefinisikan sebagai instans dari sebuah class. Setelah class didefinisikan, kita dapat membuat beberapa instans dari class tersebut. Setiap instans tersebut disebut dengan objek. Objek memiliki nilai atau data sendiri, yang disimpan dalam properti atau attribute. Objek juga dapat melakukan tindakan, yang dispesifikasikan oleh method yang terdapat dalam class.

Contohnya, class Mobil dapat kita definisikan sebagai berikut:

```
1 class Mobil:
2     def __init__(self, warna, kecepatan_max):
3         self.warna = warna
4         self.kecepatan_max = kecepatan_max
5
6     def naik_kecepatan(self, increment):
7         self.kecepatan_max += increment
8
9     def turun_kecepatan(self, decrement):
10        self.kecepatan_max -= decrement
```

Dalam contoh di atas, Mobil adalah class yang memiliki beberapa properti, yaitu warna dan kecepatan_max. Class ini juga memiliki dua method, yaitu naik_kecepatan dan turun_kecepatan. Setelah class ini didefinisikan, kita dapat membuat beberapa instans dari class tersebut, seperti berikut:

```
1 mobil_saya = Mobil('merah', 120)
2 mobil_anda = Mobil('biru', 150)
```

Dalam contoh di atas, mobil_saya dan mobil_anda adalah dua objek yang berbeda, tetapi keduanya merupakan instans dari class Mobil. Setiap objek ini memiliki nilai atau data sendiri, yang disimpan dalam properti. Objek mobil_saya memiliki warna merah dan kecepatan maksimum 120, sedangkan objek mobil_anda memiliki warna biru dan kecepatan maksimum 150. Objek ini juga dapat melakukan tindakan, yang dispesifikasikan oleh method yang terdapat dalam class.

jenis-jenis method !

Dalam pemrograman berorientasi objek (OOP), ada beberapa jenis method yang umum digunakan. Berikut adalah beberapa jenis method yang umum digunakan:

- Method konstruktor (Constructor): Method ini digunakan untuk menginisialisasi objek. Constructor ini akan dipanggil saat objek dibuat. Contoh:

```
1 class Mobil:
2     def __init__(self, warna, kecepatan_max):
3         self.warna = warna
4         self.kecepatan_max = kecepatan_max
```

- Method getter: Method ini digunakan untuk mengambil nilai dari properti objek. Contoh:

```
1 class Mobil:
2     def __init__(self, warna, kecepatan_max):
3         self.warna = warna
4         self.kecepatan_max = kecepatan_max
5
6     def get_warna(self):
7         return self.warna
8
9     def get_kecepatan_max(self):
10        return self.kecepatan_max
```

- Method setter: Method ini digunakan untuk mengatur nilai dari properti objek. Contoh:

```
1 class Mobil:
2     def __init__(self, warna, kecepatan_max):
3         self.warna = warna
4         self.kecepatan_max = kecepatan_max
5
6     def set_warna(self, warna):
7         self.warna = warna
8
9     def set_kecepatan_max(self, kecepatan_max):
10        self.kecepatan_max = kecepatan_max
```

- Method instance: Method ini digunakan untuk melakukan tindakan pada objek yang telah dibuat. Contoh:

```

1 class Mobil:
2     def __init__(self, warna, kecepatan_max):
3         self.warna = warna
4         self.kecepatan_max = kecepatan_max
5
6     def naik_kecepatan(self, increment):
7         self.kecepatan_max += increment
8
9     def turun_kecepatan(self, decrement):
10        self.kecepatan_max -= decrement

```

- Method static: Method ini digunakan untuk melakukan tindakan yang tidak bergantung pada objek yang dibuat. Contoh:

```

1 class Mobil:
2     @staticmethod
3     def hitung_jarak(jarak):
4         return jarak * 1.60934

```

masing-masing bagian sesuai dengan nomor yang ada!

- public class Komputer { - Ini adalah awal dari definisi kelas bernama Komputer. Kelas adalah blueprint untuk membuat objek di Java.
- String jenis_komputer; - Ini adalah deklarasi variabel dengan tipe String bernama jenis_komputer. Ini adalah variabel instans, yang berarti setiap objek yang dibuat dari kelas ini akan memiliki salinan variabel ini.
private String merk; - Ini adalah deklarasi variabel lain dengan tipe String bernama merk. Kata kunci private berarti bahwa variabel ini hanya dapat diakses dari dalam kelas ini.
- public void setDataKomputer (String jenis, String merk) { - Ini adalah awal dari definisi metode bernama setDataKomputer. Ini adalah metode publik, yang berarti bahwa ia dapat dipanggil dari luar kelas ini.
Metode ini menerima dua parameter, kedua-duanya dengan tipe String.
jenis komputer = jenis; - Ini adalah deklarasi variabel dan assign di dalam metode setDataKomputer. Ini mendeklarasikan variabel baru bernama jenis komputer dengan tipe String, dan menetapkan nilai dari parameter jenis ke variabel ini.
this.merk = merk; - Ini adalah assign statement di dalam metode setDataKomputer. Ini menetapkan nilai dari parameter merk ke variabel instans merk, menggunakan kata kunci this untuk merujuk ke objek saat ini.

- `public String getJenis () {` - Ini adalah awal dari definisi metode bernama `getJenis`. Ini adalah metode publik yang mengembalikan nilai dengan tipe `String`.
`return jenis_komputer;` - Ini adalah statement pengembalian di dalam metode `getJenis`. Ini mengembalikan nilai dari variabel instans `jenis_komputer`.
- `public String getMerk () {` - Ini adalah awal dari definisi metode bernama `getMerk`. Ini adalah metode publik yang mengembalikan nilai dengan tipe `String`.
`return merk;` - Ini adalah statement pengembalian di dalam metode `getMerk`. Ini mengembalikan nilai dari variabel instans `merk`.
- `public static void main(String[] args) {` - Ini adalah awal dari metode `main`, yang merupakan titik masuk dari program Java. Ini adalah metode publik, yang berarti bahwa ia dapat dipanggil dari luar kelas ini. Ini adalah metode `static`, yang berarti bahwa ia dapat dipanggil tanpa membuat objek dari kelas ini. Metode ini menerima parameter berupa array dari `string`.
`Komputer mykom = new Komputer ();` - Ini adalah statement yang membuat objek baru dari kelas `Komputer`, dan menetapkan nilainya ke variabel bernama `mykom`.
`mykom.setDataKomputer ("LAPTOP", "MACBOOK");` - Ini adalah statement yang memanggil metode `setDataKomputer` pada objek `mykom`, dan mengirim `string` `"LAPTOP"` dan `"MACBOOK"` sebagai argumen. 14.
`System.out.printin (mykom.getJenis ());` - Ini adalah statement yang memanggil metode `getJenis` pada objek `mykom`, dan mencetak nilai yang dikembalikan ke `console`. Namun, ada kesalahan di sini - harusnya `mykom.getJenis()` bukan `mykom.getJenis()`.
`System.out.println (mykom.getMerk ()) ;` - Ini adalah statement yang memanggil metode `getMerk` pada objek `mykom`, dan mencetak nilai yang dikembalikan ke `console`.