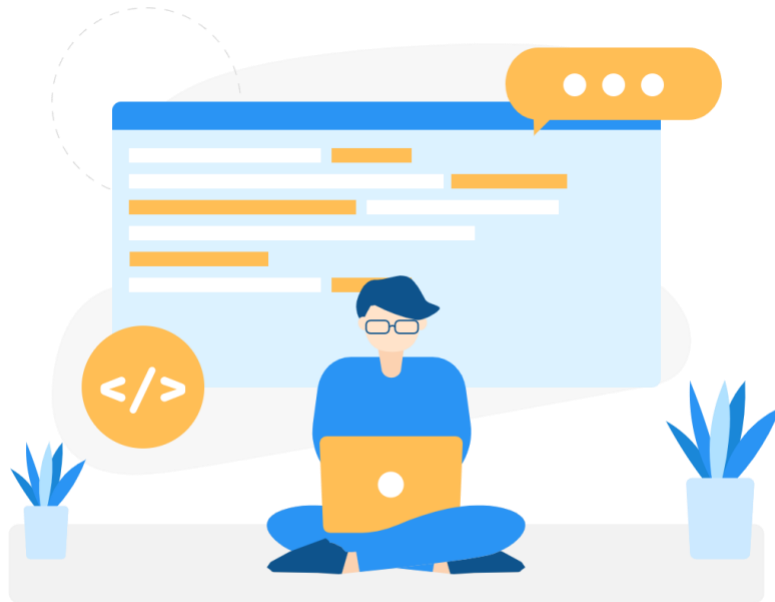


BAB 10

Inheritance (Pewarisan)



10.1 Tujuan

1. Dapat menjelaskan konsep inheritance pada bahasa python.
2. Dapat mengimplementasikan konsep inheritance pada pemrograman bahasa python

10.2 Pengantar

Inheritance atau pewarisan merupakan salah satu konsep dasar dalam Pemrograman Berorientasi Object (PBO). Dalam inheritance sebuah class atau objek dapat mewariskan sifat dan perilaku kepada class yang menjadi turunannya. Class yang menjadi pemberi waris disebut dengan class induk atau class basis sedangkan class yang menjadi ahli waris dinamakan dengan class turunan. Class turunan dapat mewarisi semua atribut dan method dari class induknya, dan juga bisa menambah atribut dan method baru. Sedangkan class induk belum tentu memiliki semua atribut dan method dari class turunannya.

10.2.1 Konsep Inheritance

Untuk mempermudah memahami konsep tentang inheritance, kita dapat mengambil studi kasus dari sebuah mobil. Sebagai contoh, mobil elektrik merupakan salah satu tipe mobil yang spesifik. Sehingga kita dapat membuat class mobil elektrik turunan dari sebuah class mobil. Untuk memahami secara lebih jelas perhatikan tabel berikut ini.

Tabel 10.1 Contoh Class dan Object

Class	Atribut	Class Turunan	Atribut Class Turunan
Car	merk, model, tahun	Electric Car	merk, model, tahun, baterai

Untuk membuat sebuah class kita dapat menggunakan kode berikut ini. Nama sebuah kelas harus diawali dengan huruf kapital.

```
1. class NamaClassInduk:
2.     Isi dari class induk
3. class NamaClassTurunan(NamaClassInduk):
4.     Isi dari class turunan
```

Contoh dalam pembuatan class Car dan turunannya yaitu electricCar :

```
1. class Car:
2.     def __init__(self, merk, model, tahun):
```

```

3.         self.merk = merk
4.         self.model = model
5.         self.tahun = tahun
6.         self.odometer = 0
7.
8.     def keterangan(self):
9.         print(f"Mobil baru saya {self.merk} {self.model} tahun {s
elf.tahun} kilometernya masih {self.odometer}")
10.
11.
12. class ElectricCar(Car):
13.     def __init__(self, merk, model, tahun):
14.         super().__init__(merk, model, tahun)
15.
16.
17. teslaku = ElectricCar('tesla', 'model X', '2022')
18. print(teslaku.keterangan())

```

Baris ke 1-9 merupakan kode program python untuk class Car yang sudah kita pelajari di praktikum bab sebelumnya. Baris ke 12 merupakan deklarasi class ElectricCar yang merupakan pewaris dari class Car. Baris ke 13 merupakan method `__init__` yang digunakan untuk mendeklarasikan semua atribut yang ada pada class Car. Baris ke 14 merupakan `super()` function yang berfungsi untuk memanggil semua method yang ada pada Class induk. Sehingga jika kode diatas dijalankan akan menghasilkan output seperti berikut :

```
Mobil baru saya tesla model X tahun 2022 kilometernya masih 0
```

Dapat dilihat di baris ke 12-14 bahwa Class ElectricCar tidak memiliki method keterangan, namun di baris ke-18 kita dapat memanggil method keterangan yang merupakan method dari Class Car yang merupakan Class Induknya.

10.3 Kegiatan Praktikum

10.3.1 Kegiatan Praktikum 1 : Atribut dan Method baru untuk Class Turunan

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. class Car:
2.     def __init__(self, merk, model, tahun):
3.         self.merk = merk
4.         self.model = model
5.         self.tahun = tahun
6.         self.odometer = 0
7.
8.     def keterangan(self):
9.         print(f"Mobil baru saya {self.merk} {self.model} tahun {s
elf.tahun} kilometranya masih {self.odometer}")
10.
11.
12. class ElectricCar(Car):
13.     def __init__(self, merk, model, tahun, baterai):
14.         super().__init__(merk, model, tahun)
15.         self.baterai = baterai
16.
17.     def daya(self):
18.         print(f"Mobil ini memiliki daya {self.baterai} -kWh")
19.
20.
21. teslaku = ElectricCar('tesla', 'model X', 2022, 7500)
22. print(teslaku.dayu())
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 1 ini

10.3.2 Kegiatan Praktikum 2 : Override (Menimpa) method Class Induk

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```
1. class Car:
2.     def __init__(self, merk, model, tahun, odometer):
3.         self.merk = merk
4.         self.model = model
5.         self.tahun = tahun
6.         self.odometer = odometer
7.
```

```

8.     def keterangan(self):
9.         print(f"Mobil baru saya {self.merk} {self.model} tahun {s
elf.tahun} kilometernya masih {self.odometer}")
10.
11.     def gantioli(self):
12.         print(f"Mobil {self.merk} ini perlu ganti oli ketika odom
eter {self.odometer}")
13.
14.
15. class ElectricCar(Car):
16.     def __init__(self, merk, model, tahun, odometer, baterai):
17.         super().__init__(merk, model, tahun, odometer)
18.         self.baterai = baterai
19.
20.     def daya(self):
21.         print(f"Mobil ini memiliki daya {self.baterai} -kWh")
22.
23.     def gantioli(self):
24.         print(f"Mobil listrik tidak memerlukan ganti oli")
25.
26.
27. alphardku = Car('toyota', 'alphard', 2022, 10000)
28. print(alphardku.gantioli())
29.
30. teslaku = ElectricCar('tesla', 'model X', 2022, 10000, 7500)
31. print(teslaku.gantioli())

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 2 ini

10.3.3 Kegiatan Praktikum 3 : Multiple Class Inheritance

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini

```

1. class Penjumlahan:
2.     def __init__(self, angka1, angka2):
3.         self.angka1 = angka1
4.         self.angka2 = angka2
5.
6.     def jumlah(self):
7.         return self.angka1 + self.angka2
8.
9.

```

```

10. class Perkalian():
11.     def __init__(self, angka1, angka2):
12.         self.angka1 = angka1
13.         self.angka2 = angka2
14.
15.     def kali(self):
16.         return self.angka1 * self.angka2
17.
18.
19. class Pembagian(Penjumlahan, Perkalian):
20.
21.     def bagi(self):
22.         return self.angka1 / self.angka2
23.
24.
25. hitung = Pembagian(8, 2)
26. print(hitung.kali())
27. print(hitung.bagi())
28. print(hitung.jumlah())

```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

10.4 Tugas

1. A. Buatlah sebuah class dengan nama “Orang”. Buat 2 atribut dengan nama "nama", "umur" . Buatlah sebuah method dengan nama "kenalan" yang akan menampilkan ucapan " Halo, namaku {nama}, umurku {umur} ".
 B. Buatlah class turunan dengan nama “Mahasiswa” dan “Pekerja”.
 C. Class “Mahasiswa” memiliki tambahan atribut “universitas”, dengan override method “kenalan” yang akan menampilkan ucapan " Halo, namaku {nama}, umurku {umur} dan aku kuliah di {universitas} ".
 D. Class “Pekerja” memiliki tambahan atribut “tempatKerja”, dengan override method “kenalan” yang akan menampilkan ucapan " Halo, namaku {nama}, umurku {umur} dan aku kerja di {tempatKerja} ".
 E. Buatlah 3 object yang mewakili class “Orang”, “Mahasiswa” dan “Pekerja” dan panggil method “kenalan” untuk setiap objectnya.