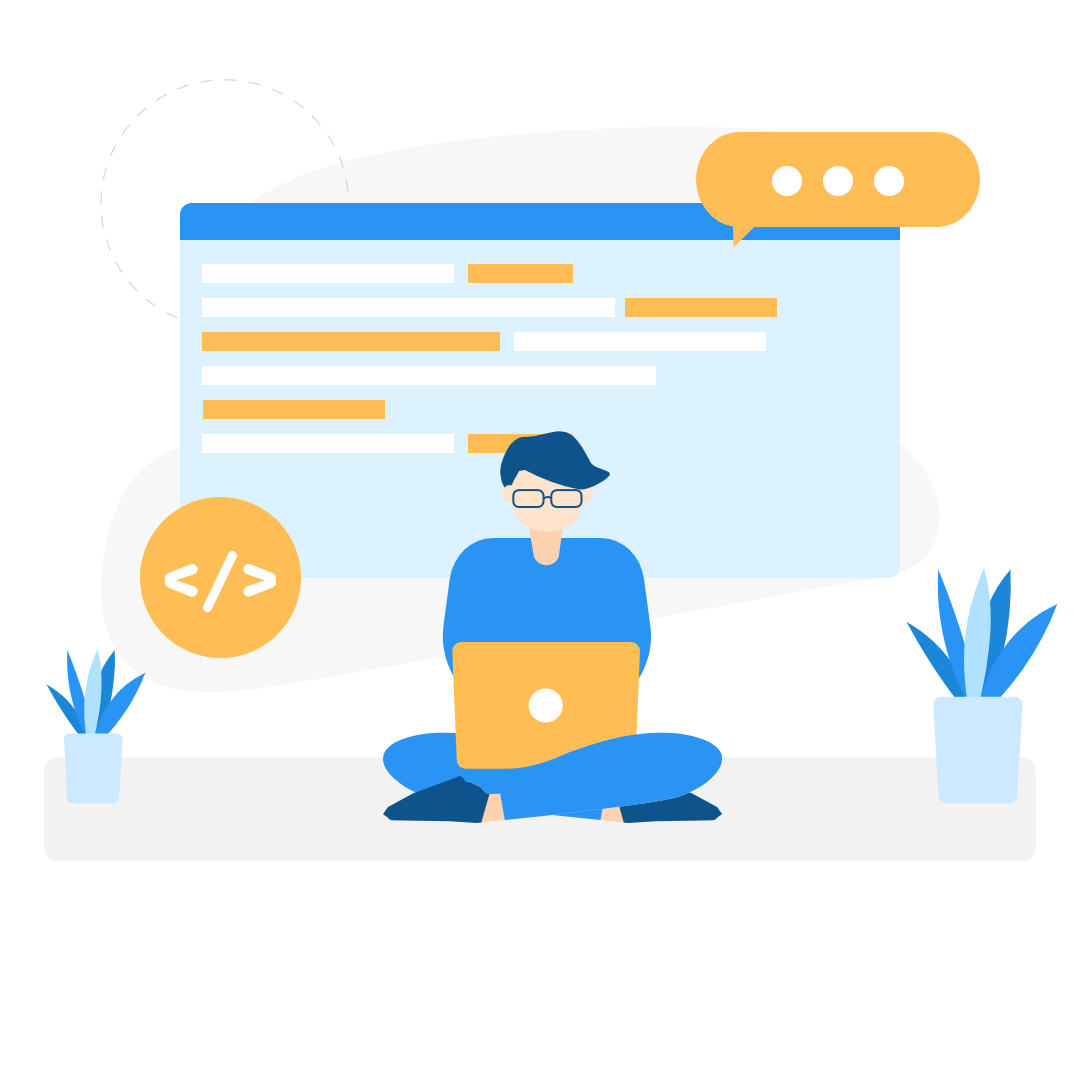
# BAB 10

# Inheritence (Pewarisan)



## Tujuan

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

## Pengantar

Inheritence atau pewarisan merupakan salah satu konsep dasar dalam Pemrograman Berorientasi Object (PBO). Dalam inheritence 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.

### Konsep Inheritence

Untuk mempermudah memahami konsep tentang inheritence, 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** | **Attribut** | **Class Turunan** | **Attribut 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. def keterangan(self):
8. print(f"Mobil baru saya {self.merk} {self.model} tahun {self.tahun} kilometernya masih {self.odometer}")
9. class ElectricCar(Car):
10. def \_\_init\_\_(self, merk, model, tahun):
11. super().\_\_init\_\_(merk, model, tahun)
12. teslaku = ElectricCar('tesla', 'model X', '2022')
13. 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.

## Kegiatan Praktikum

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

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini
2. class Car:
3. def \_\_init\_\_(self, merk, model, tahun):
4. self.merk = merk
5. self.model = model
6. self.tahun = tahun
7. self.odometer = 0
8. def keterangan(self):
9. print(f"Mobil baru saya {self.merk} {self.model} tahun {self.tahun} kilometernya masih {self.odometer}")
10. class ElectricCar(Car):
11. def \_\_init\_\_(self, merk, model, tahun, baterai):
12. super().\_\_init\_\_(merk, model, tahun)
13. self.baterai = baterai
14. def daya(self):
15. print(f"Mobil ini memiliki daya {self.baterai} -kWh")
17. teslaku = ElectricCar('tesla', 'model X', 2022, 7500)
18. print(teslaku.daya())
19. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 1 ini

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

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini
2. class Car:
3. def \_\_init\_\_(self, merk, model, tahun, odometer):
4. self.merk = merk
5. self.model = model
6. self.tahun = tahun
7. self.odometer = odometer
8. def keterangan(self):
9. print(f"Mobil baru saya {self.merk} {self.model} tahun {self.tahun} kilometernya masih {self.odometer}")
10. def gantioli(self):
11. print(f"Mobil {self.merk} ini perlu ganti oli ketika odometer {self.odometer}")
12. class ElectricCar(Car):
13. def \_\_init\_\_(self, merk, model, tahun, odometer, baterai):
14. super().\_\_init\_\_(merk, model, tahun, odometer)
15. self.baterai = baterai
16. def daya(self):
17. print(f"Mobil ini memiliki daya {self.baterai} -kWh")
18. def gantioli(self):
19. print(f"Mobil listrik tidak memerlukan ganti oli")
20. alphardku = Car('toyota', 'alphard', 2022, 10000)
21. print(alphardku.gantioli())
22. teslaku = ElectricCar('tesla', 'model X', 2022, 10000, 7500)
23. print(teslaku.gantioli())
24. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 2 ini

### Kegiatan Praktikum 3 : Multiple Class Inheritence

1. Buat sebuah file program baru, kemudian tuliskan kode berikut ini
2. class Penjumlahan:
3. def \_\_init\_\_(self, angka1, angka2):
4. self.angka1 = angka1
5. self.angka2 = angka2
6. def jumlah(self):
7. return self.angka1 + self.angka2
8. class Perkalian():
9. def \_\_init\_\_(self, angka1, angka2):
10. self.angka1 = angka1
11. self.angka2 = angka2
12. def kali(self):
13. return self.angka1 \* self.angka2
14. class Pembagian(Penjumlahan, Perkalian):
15. def bagi(self):
16. return self.angka1 / self.angka2
17. hitung = Pembagian(8, 2)
18. print(hitung.kali())
19. print(hitung.bagi())
20. print(hitung.jumlah())
21. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan 3 ini

## Tugas

1. A. Buatlah sebuah class dengan nama “Orang”. Buat 2 atribute 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.