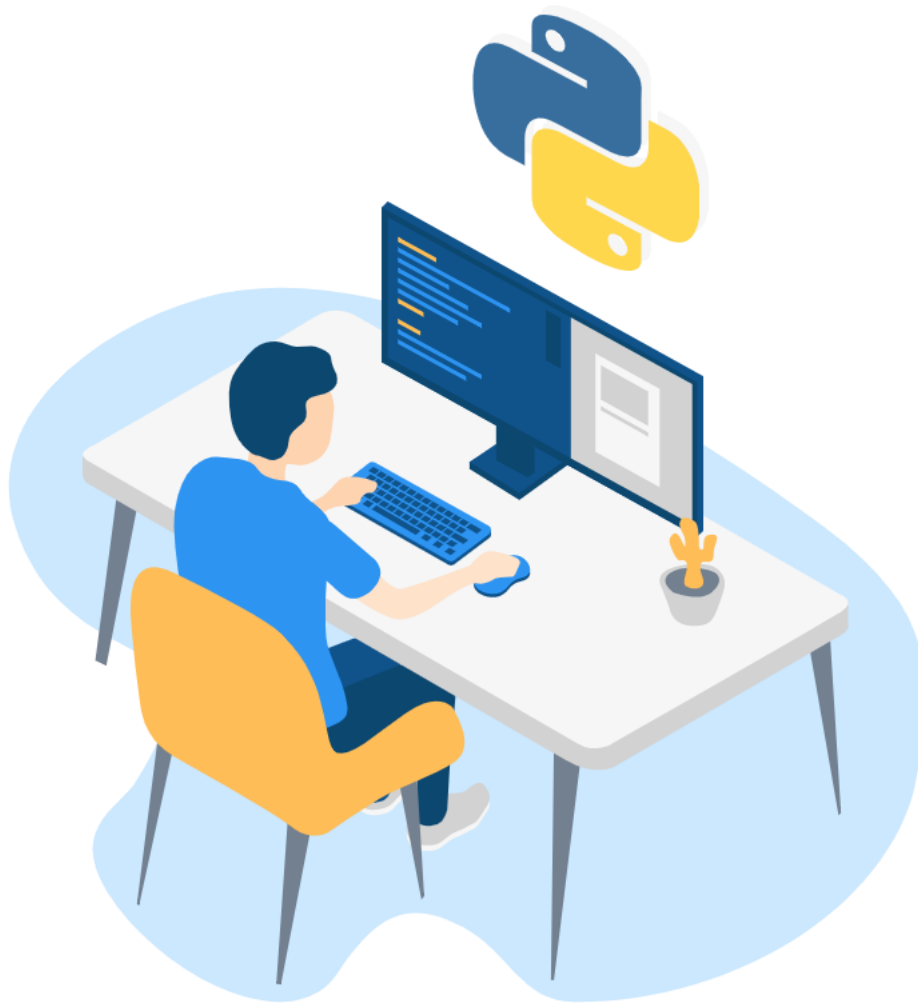


## BAB 5

### Inheritance



#### 5.1 Tujuan

1. Dapat memahami konsep inheritance.
2. Dapat mengimplementasikan konsep inheritance menggunakan Bahasa pemrograman python.

## **5.2 Pengantar**

### **5.2.1 Konsep Inheritance**

Konsep inheritance pada OOP (Object-Oriented Programming) adalah sebuah konsep di mana sebuah kelas baru (subclass) dibuat dengan mewarisi sifat atau perilaku dari kelas yang sudah ada (superclass). Kelas yang mewarisi sifat dari superclass disebut sebagai subclass atau turunan, sedangkan kelas yang memberikan sifat atau perilaku disebut sebagai superclass atau induk.

Dengan menggunakan konsep inheritance, kita dapat membuat sebuah kelas baru dengan mewarisi sifat dan perilaku dari kelas yang sudah ada, sehingga mempermudah pembuatan program dan meningkatkan fleksibilitas dan modularitas pada program. Selain itu, kita juga dapat menghindari duplikasi kode dan membuat perubahan pada superclass secara otomatis diterapkan pada semua subclass yang mewarisinya.

Contoh inheritance dalam kehidupan sehari-hari adalah warisan atau turunan yang diterima oleh anak dari orang tua atau keluarga lainnya. Konsep inheritance dalam OOP mirip dengan konsep warisan dalam kehidupan nyata, dimana sifat atau ciri-ciri tertentu dapat diwariskan dari generasi sebelumnya ke generasi selanjutnya.

Sebagai contoh, jika seorang ayah memiliki bisnis restoran, maka anaknya dapat mewarisi bisnis tersebut dan melanjutkan usaha yang telah dimulai oleh ayahnya. Anak tersebut dapat mengembangkan bisnis restoran dengan menambahkan menu baru atau melakukan perubahan yang diperlukan, tetapi tetap mempertahankan sifat atau ciri khas yang dimiliki oleh bisnis restoran yang diwarisi dari ayahnya.

Konsep inheritance juga dapat diterapkan pada ilmu biologi, dimana sifat-sifat atau ciri-ciri biologis dapat diwariskan dari generasi sebelumnya ke generasi selanjutnya. Misalnya, sifat tertentu seperti warna mata atau tinggi badan dapat diwariskan dari orang tua ke anaknya. Seorang anak juga dapat mewarisi kecenderungan terhadap penyakit tertentu dari orang tua atau kakek neneknya.

### 5.2.2 Superclass dan Subclass

Perbedaan antara subclass dan superclass pada konsep inheritance di OOP adalah:

- Superclass adalah kelas yang memberikan sifat atau perilaku yang dapat diwarisi oleh subclass, sedangkan subclass adalah kelas yang mewarisi sifat atau perilaku dari superclass.
- Superclass adalah kelas yang lebih umum atau generik, sedangkan subclass adalah kelas yang lebih khusus atau spesifik.
- Superclass bisa memiliki beberapa subclass yang berbeda, sedangkan subclass hanya memiliki satu superclass.
- Subclass dapat menambahkan atribut atau method yang unik untuk dirinya sendiri, sementara superclass tidak dapat mempengaruhi subclass dengan cara apapun.

### 5.2.3 Inheritance Single Level

Single level inheritance adalah ketika sebuah kelas turunan hanya memiliki satu kelas induk atau superclass. Dalam single level inheritance, subclass mewarisi semua metode dan properti dari superclassnya.

```
1. class Manusia:
2.     def __init__(self, nama, umur):
3.         self.nama = nama
4.         self.umur = umur
5.
6.     def info(self):
7.         print(f>Nama: {self.nama}")
8.         print(f"Umur: {self.umur}")
9.
10. class Pelajar(Manusia):
11.     def __init__(self, nama, umur, kelas):
12.         super().__init__(nama, umur)
13.         self.kelas = kelas
14.
15.     def info(self):
16.         super().info()
17.         print(f"Kelas: {self.kelas}")
```

### 5.2.4 Inheritance Multi Level

Multi level inheritance adalah ketika sebuah kelas turunan memiliki lebih dari satu kelas induk atau superclass. Dalam multi level inheritance, sebuah subclass mewarisi semua metode dan properti dari superclassnya, termasuk dari kelas induk.

```
1. class Kendaraan:
2.     def __init__(self, jenis):
3.         self.jenis = jenis
4.
5. class Sepeda(Kendaraan):
6.     def __init__(self, jenis, warna):
7.         super().__init__(jenis)
8.         self.warna = warna
9.
10. class SepedaMotor(Sepeda):
11.     def __init__(self, jenis, warna, merk):
12.         super().__init__(jenis, warna)
13.         self.merk = merk
```

### 5.2.5 Inheritance Hierarki

Hierarki inheritance adalah ketika beberapa kelas turunan berasal dari satu kelas induk atau superclass yang sama. Dalam hierarki inheritance, kelas-kelas turunan memiliki metode dan properti yang sama dari superclassnya.

```
1. class Hewan:
2.     def __init__(self, jenis):
3.         self.jenis = jenis
4.
5. class Kucing(Hewan):
6.     def __init__(self, jenis, warna):
7.         super().__init__(jenis)
8.         self.warna = warna
9.
10. class Anjing(Hewan):
11.     def __init__(self, jenis, ras):
12.         super().__init__(jenis)
13.         self.ras = ras
14.
15. class Singa(Hewan):
16.     def __init__(self, jenis, habitat):
17.         super().__init__(jenis)
```

```
18.         self.habitat = habitat
```

## 5.3 Kegiatan Praktikum

### 5.3.1 Kegiatan 1 : Inheritance Single Level

1. Buat sebuah file program baru kemudian tulis kode program berikut ini

```
1. class Person:
2.     def __init__(self, name, age):
3.         self.name = name
4.         self.age = age
5.
6.     def introduce(self):
7.         print(f"My name is {self.name} and I am {self.age} years old.")
8.
9. class Student(Person):
10.    def __init__(self, name, age, student_id):
11.        super().__init__(name, age)
12.        self.student_id = student_id
13.
14.    def introduce(self):
15.        super().introduce()
16.        print(f"My student ID is {self.student_id}.")
17.
18. person1 = Person("John", 30)
19. person1.introduce()
20.
21. student1 = Student("Mary", 20, "12345")
22. student1.introduce()
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan ini

### 5.3.2 Kegiatan 2 : Inheritance Multi Level

1. Buat sebuah file program baru kemudian tulis kode program berikut ini

```
1. class Kendaraan:
2.     def __init__(self, jenis):
3.         self.jenis = jenis
4.
5. class Sepeda(Kendaraan):
6.     def __init__(self, jenis, warna):
7.         super().__init__(jenis)
8.         self.warna = warna
9.
10. class SepedaMotor(Sepeda):
11.     def __init__(self, jenis, warna, merk):
12.         super().__init__(jenis, warna)
13.         self.merk = merk
14.
15. class Mobil(SepedaMotor):
16.     def __init__(self, jenis, warna, merk, tahun):
17.         super().__init__(jenis, warna, merk)
18.         self.tahun = tahun
19.
20.     def info(self):
21.         print(f"Jenis: {self.jenis}")
22.         print(f"Warna: {self.warna}")
23.         print(f"Merk: {self.merk}")
24.         print(f"Tahun: {self.tahun}")
25.
26. mobil1 = Mobil("Sedan", "Merah", "Toyota", 2022)
27. mobil1.info()
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan ini

### 5.3.3 Kegiatan 3 : Inheritance Hierarki

1. Buat sebuah file program baru kemudian tulis kode program berikut ini

```
1. class Hewan:
2.     def __init__(self, nama, jenis):
3.         self.nama = nama
4.         self.jenis = jenis
5.
6. class Mamalia(Hewan):
7.     def __init__(self, nama, jenis, mamalia_bertelur):
8.         super().__init__(nama, jenis)
9.         self.mamalia_bertelur = mamalia_bertelur
10.
11. class Unggas(Hewan):
12.     def __init__(self, nama, jenis, unggas_terbang):
13.         super().__init__(nama, jenis)
14.         self.unggas_terbang = unggas_terbang
15.
16. class Gajah(Mamalia):
17.     def __init__(self, nama, jenis, mamalia_bertelur, berat_badan):
18.         super().__init__(nama, jenis, mamalia_bertelur)
19.         self.berat_badan = berat_badan
20.
21. class Ayam(Unggas):
22.     def __init__(self, nama, jenis, unggas_terbang, jumlah_telur):
23.         super().__init__(nama, jenis, unggas_terbang)
24.         self.jumlah_telur = jumlah_telur
25.
26.     def info(self):
27.         print(f>Nama: {self.nama}")
28.         print(f>Jenis: {self.jenis}")
29.         print(f>Unggas Terbang: {self.unggas_terbang}")
30.         print(f>Jumlah Telur: {self.jumlah_telur}")
31.
32. ayam1 = Ayam("Ayam Serama", "Ayam", True, 20)
33. ayam1.info()
34.
35. gajah1 = Gajah("Gajah Sumatera", "Gajah", False, 4000)
36. print(f>Nama: {gajah1.nama}")
37. print(f>Jenis: {gajah1.jenis}")
38. print(f>Mamalia Bertelur: {gajah1.mamalia_bertelur}")
39. print(f>Berat Badan: {gajah1.berat_badan}")
```

2. Amati hasilnya kemudian tulis analisis singkat mengenai kegiatan ini

## 5.4 Tugas

1. Buatlah sebuah program untuk menghitung harga tiket bioskop yang menerapkan konsep inheritance. Buatlah superclass 'Tiket' dan subclass 'TiketBiasa', 'TiketVIP', dan 'TiketGold'. Setiap subclass memiliki atribut dan method berisi harga tiket yang berbeda-beda. Output yang diharapkan dari program ini adalah seperti berikut

Masukkan jenis tiket (biasa/vip/gold) : *(input dari user)*

Masukkan jumlah tiket : *(input dari user)*

Total Harga Tiket : Rp *(Output Hasil perhitungan sistem)*