

BAB 5 INHERITANCE

5.1 Maksud dan Tujuan

5.1.1 Maksud

Bab ini akan membahas tentang inheritance pada pemrograman berorientasi obyek (PBO) dan penerapannya menggunakan bahasa pemrograman Python.

5.1.2 Tujuan

1. Mahasiswa dapat memahami tentang inheritance dalam PBO
2. Mahasiswa dapat melakukan PBO menggunakan inheritance

5.2 Dasar Teori Inheritance

Inheritance dalam PBO dapat diartikan sebagai pewarisan atribut dari kelas induk (*parent class*) ke kelas anak (*child class*). Dibandingkan dengan membuat kelas baru dengan atribut yang sama, sebaiknya kelas tersebut didefinisikan sebagai kelas anak dari kelas yang sudah ada.

Kelas anak mewarisi atribut dari kelas induk sehingga bisa menggunakan atribut kelas induk seolah atribut tersebut didefinisikan di kelas anak. Satu kelas anak bisa mewarisi karakteristik dari satu atau lebih kelas induk.

```
class Polygon:
    def __init__(self, banyak_sisi):
        self.n = banyak_sisi
        self.sisi = [0 for i in range(banyak_sisi)]

    def inputSisi(self):
        print("Masukkan panjang sisi (dalam cm):")
        self.sisi = [float(input("Sisi "+str(i+1)+" : ")) for i in range(self.n)]
        print()

    def dispSisi(self):
        for i in range(self.n):
            print("Panjang sisi",i+1,"adalah",self.sisi[i],"cm")
```

Walaupun kelas anak mewarisi karakteristik dari kelas induk, namun kelas anak dapat merubah data dan metode dari kelas induk dengan data dan metodenya sendiri. Hal ini disebut dengan *overriding*.

Script sebelumnya merupakan contoh sebuah kelas induk bernama Polygon yang mempunyai atribut **banyak_sisi** untuk menyimpan nilai dari banyaknya sisi suatu bangun datar. Kelas Polygon mempunyai method **inputSisi** dan **dispSisi**. Method **inputSisi** digunakan untuk menerima masukkan panjang sisi, sedangkan method **dispSisi** digunakan untuk menampilkan data panjang sisi dari obyek yang ada di kelas Polygon.

Sebuah kelas bernama segiEmpat ditambahkan sebagai kelas anak dari Polygon. Kelas segiEmpat mempunyai method bernama **hitungKeliling** untuk menghitung dan menampilkan keliling dari segiempat.

```
class segiEmpat(Polygon):
    #menyatakan bahwa segiEmpat merupakan kelas anak dari Polygon

    def __init__(self):
        Polygon.__init__(self,4)
        print("Segiempat")

    def hitungKeliling(self):
        a,b,c,d = self.sisi
        k = a+b+c+d
        print("Keliling Segiempat adalah",k,"cm.")
```

Baris pertama pada script menyatakan bahwa kelas segiEmpat merupakan kelas anak dari kelas Polygon. Sehingga dapat dilihat bahwa obyek s4 pada kelas segiEmpat memiliki atribut **banyak_sisi** yang diturunkan dari kelas induk Polygon. Obyek s4 juga bisa mengakses method **inputSisi** dan **dispSisi** yang berada pada kelas Polygon.

Selain bisa mengakses method yang dimiliki oleh kelas induk, kelas anak juga bisa mempunyai method sendiri. Sehingga obyek s4 bisa mengakses method **hitungKeliling** yang hanya dimiliki oleh kelas segiEmpat. Hal ini tidak bisa dilakukan oleh obyek poligon (pada script sebelumnya) karena poligon merupakan obyek dari kelas induk sedangkan **hitungKeliling** merupakan method dari kelas anak.

5.3 Latihan

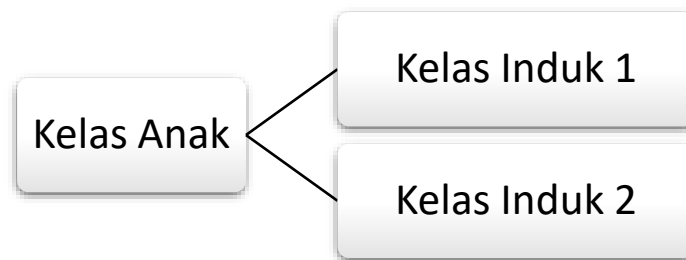
1. Buatlah satu kelas induk Buku dengan atribut judul buku, tahun terbit, nama pengarang, dan penerbit.
2. Buatlah kelas anak novel, komik, majalah, atau buku pelajaran.

3. Tambahkan method untuk memasukkan data pada kelas induk.
4. Buatlah satu method yang hanya bisa dimiliki oleh kelas anak.
5. Buatlah obyek dari kelas anak dan tampilkan datanya.

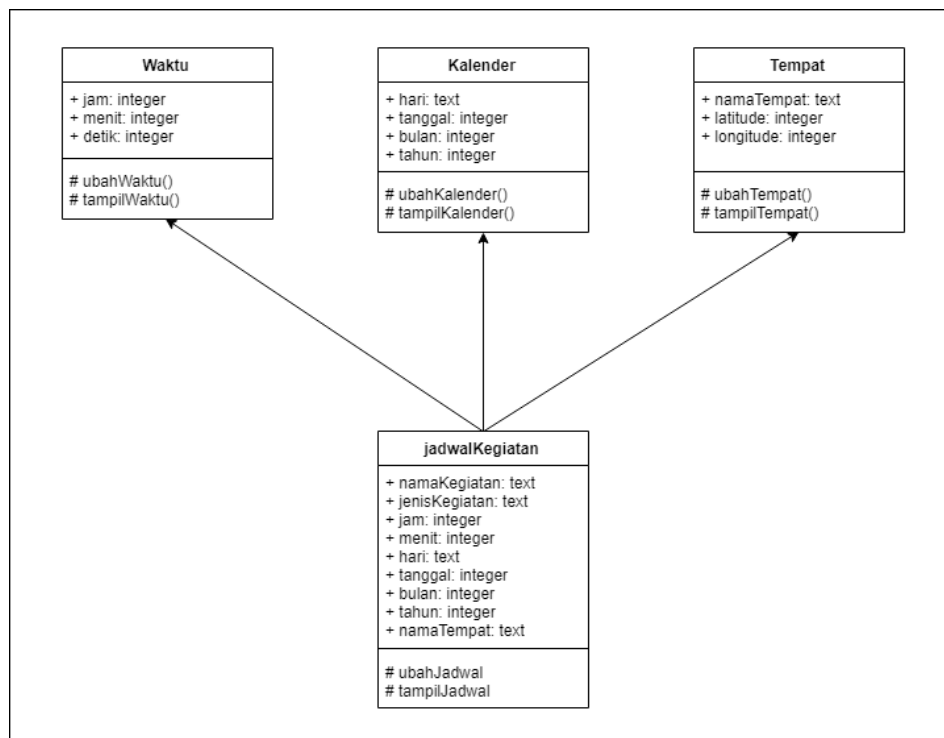
5.4 Multiple Inheritance

Inheritance adalah pewarisan karakteristik dari kelas induk ke kelas anak. Pada inheritance biasa, kelas anak hanya mewarisi karakteristik dari satu kelas induk hal ini dikenal dengan nama *single inheritance*.

Dalam PBO, kelas anak bisa mewarisi karakteristik dari beberapa kelas induk sekaligus, hal ini dikenal dengan nama *multiple inheritance*. Sehingga obyek yang berasal dari kelas anak akan memiliki karakteristik dari semua kelas induk ditambah dengan karakteristik yang dimiliki oleh kelas anak.



Berikut ini adalah contoh penerapan *multiple inheritance* pada PBO menggunakan Python. Jadwal kegiatan merupakan kelas anak dari kelas waktu, tanggal, dan tempat. Untuk lebih jelasnya dapat dilihat pada gambar berikut.



Berikut ini adalah contoh kodingan PBO untuk contoh kasus jadwal kegiatan.

```
class waktu:
    # def __init__(self, jam=0, menit=0, detik=0):
    def __init__(self, jam=0, menit=0, detik=0):
        self.jam = jam
        self.menit = menit
        self.detik = detik
    def ubahWaktu(self, jam, menit, detik=0):
        self.jam = jam
        self.menit = menit
        self.detik = detik

class tempat:
    # def __init__(self, nama, latitude, longitude):
    def __init__(self):
        self.namaTempat = namaTempat
        self.lattitude = latitude
        self.longitude = longitude

class kalender:
    # def __init__(self, hari, tanggal, bulan, tahun):
    def __init__(self):
        self.hari=hari
        self.tanggal=tanggal
        self.bulan=bulan
        self.tahun=tahun
    def ubahKalender(self, hari, tanggal, bulan, tahun):
        self.hari=hari
        self.tanggal=tanggal
        self.bulan=bulan
        self.tahun=tahun

class jadwal(waktu,tempat,kalender):
    def __init__(self):
        self.namaKegiatan = 'Kuliah PBO'
        self.jenisKegiatan = 'Kuliah Teori'
        self.hari='Rabu'
        self.tanggal=6
        self.bulan=3
        self.tahun=2019
        self.jam = 11
        self.menit = 30
        self.namaTempat="Kampus 1 UNJANI"
```

```
def tampilJadwal(self):
    print('Jadwal')
    print('-----')
    print('Nama Kegiatan :', self.namaKegiatan)
    print('Jenis Kegiatan  :', self.jenisKegiatan)
    print('Hari/tgl       :', self.hari,',',self.tanggal,'/',self.bulan,'/',self.tahun)
    print('Waktu         :', self.jam,':',self.menit,'WIB')
    print('Tempat         :', self.namaTempat)
```

5.5 Latihan

1. Ubahlah kodingan pada bab 5.4 supaya inputan bisa dilakukan melalui keyboard!
2. Buatlah method ubahJadwal pada kelas **jadwal**!
3. Ubahlah nilai dari sebuah obyek pada kelas jadwal melalui method ubahJadwal!