

PEMROGRAMAN

BERORIENTASI

OBJEK

Pertemuan 5 (Praktikum)

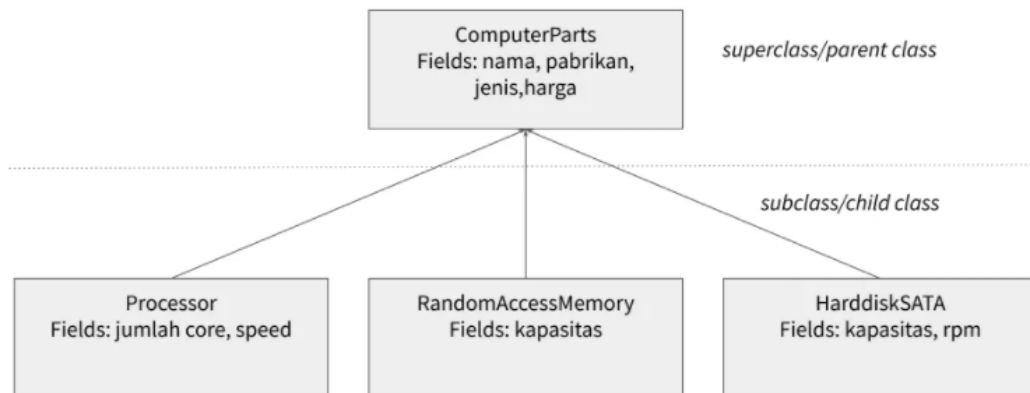
INHERITANCE

(PEWARISAN)

KONSEP INHERITANCE

- » Pewarisan adalah proses mewariskan karakteristik yang ada di kelas induk ke kelas anak tanpa mengalami perubahan.
- » Kelas induk dikenal dengan parent class, super class, kelas utama
- » Kelas anak dikenal dengan child class, sub class, kelas turunan
- » Manfaat yang didapatkan jika menggunakan pewarisan, yaitu: *code reuse* (kode tidak perlu ditulis ulang) dan tersedianya informasi struktur klasifikasi hierarki antar kelas.

Berikut ini adalah ilustrasi untuk memahami konsep pewarisan:



Kode program class ComputerPart:

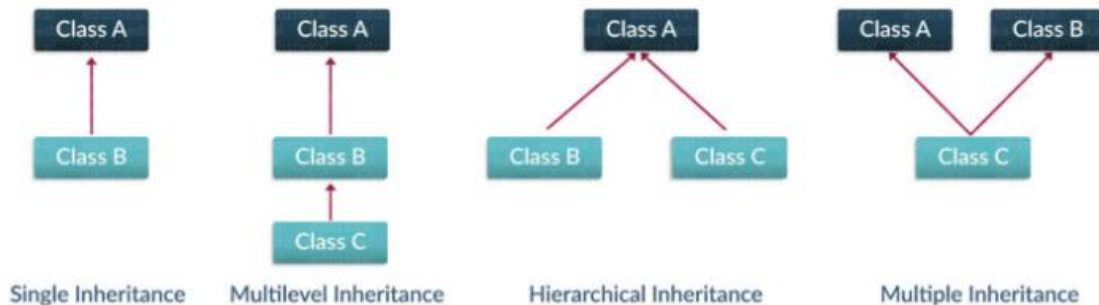
```
1 class ComputerPart:
2     def __init__(self, pabrikan, nama, jenis, harga):
3         self.pabrikan = pabrikan
4         self.nama = nama
5         self.jenis = jenis
6         self.harga = harga
7
8 class Processor(ComputerPart):
9     def __init__(self, pabrikan, nama, harga, jumlah_core, speed):
10        super().__init__(pabrikan, nama, 'processor', harga)
11        self.jumlah_core = jumlah_core
12        self.speed = speed
13
14 class RandomAccessMemory(ComputerPart):
15     def __init__(self, pabrikan, nama, harga, kapasitas):
16        super().__init__(pabrikan, nama, 'RAM', harga)
17        self.kapasitas = kapasitas
18
19 class HardDiskSATA(ComputerPart):
20     def __init__(self, pabrikan, nama, harga, kapasitas, rpm):
21        super().__init__(pabrikan, nama, 'SATA', harga)
22        self.kapasitas = kapasitas
23        self.rpm = rpm
```

```

1 p = Processor('Intel','Core i7 7740X',4350000,4,'4.3GHz')
2 m = RandomAccessMemory('V-Gen','DDR4 SODimm PC19200/2400MHz',328000,'4GB')
3 hdd = HardDiskSATA('Seagate','HDD 2.5 inch',295000,'500GB',7200)
4
5 parts = [p,m,hdd]
6 for part in parts:
7     print('{} {} produksi {}'.format(part.jenis,part.nama,part.pabrikan))

```

Python mempunyai empat tipe pewarisan yaitu:



Berikut ini merupakan contoh-contoh program dari keempat tipe pewarisan:

Single Inheritance, dengan kode program class Mahasiswa:

```

1 # Single Inheritance
2
3 class Mahasiswa:
4     def __init__(self, nama, nim):
5         self.nama=nama
6         self.nim=nim
7
8     def detailMhs(self):
9         return self.nim, self.nama
10
11    def cekMhs(self):
12        if self.nim < 150000:
13            return "Mahasiswa aktif"
14        else:
15            return "Mahasiswa tidak terdaftar"
16
17    class Siswa(Mahasiswa):
18        def End(self):
19            print("Mahasiswa belum melakukan daftar ulang")
20
21    mahasiswa1 = Mahasiswa("Mahasiswa 1", 135103)
22    print(mahasiswa1.detailMhs(), mahasiswa1.cekMhs())
23    mahasiswa2 = Siswa("Mahasiswa 2", 150503)
24    print(mahasiswa2.detailMhs(), mahasiswa2.cekMhs())
25    mahasiswa2.End()

```

Multilevel Inheritance, dengan kode program class Mahasiswa:

```
1  # Multilevel Inheritance
2
3  class Mahasiswa():
4      def __init__(self,nama,nim):
5          self.nama=nama
6          self.nim=nim
7
8  class Siswa1(Mahasiswa):
9      def __init__(self,nama,nim):
10         self.nama=nama
11         self.nim=nim
12
13 class Siswa2(Siswa1):
14     def __init__(self,nama,nim):
15         self.nama=nama
16         self.nim=nim
17
18 mhs1=Mahasiswa('Mikasa',135105)
19 mhs2=Siswa1('Nezuko',135117)
20 mhs3=Siswa2('Hancock',134079)
21
22 print(mhs1.nama, mhs1.nim)
23 print(mhs2.nim)
24 print(mhs3.nama)
```

Hierarchical Inheritance, dengan kode program class Mahasiswa:

```
1  # Hierarchical Inheritance
2
3  class Mahasiswa():
4      def __init__(self,nama,nim):
5          self.nama=nama
6          self.nim=nim
7
8  class Siswa1(Mahasiswa):
9      def __init__(self,nama,nim):
10         self.nama=nama
11         self.nim=nim
12
13         def detSiswa1(self):
14             print (self.nama,'alamat: wall rose')
15
16 class Siswa2(Mahasiswa):
17     def __init__(self,nama,nim):
18         self.nama=nama
19         self.nim=nim
20
21     def detSiswa2(self):
22         print (self.nama,'jurusan: Informatika')
23
24 mhs1=Siswa1('Mikasa',135105)
25 mhs2=Siswa2('Nezuko',135117)
26
27 print(mhs1.nim)
28 mhs1.detSiswa1()
29 print(mhs2.nim)
30 mhs2.detSiswa2()
```

Multiple Inheritance, dengan kode program class Mahasiswa:

```
1 # Multiple Inheritance
2
3 class Mahasiswa1():
4     def __init__(self,nama,nim):
5         self.nama=nama
6         self.nim=nim
7
8 class Mahasiswa2():
9     def __init__(self,alamat,jurusan):
10         self.alamat=alamat
11         self.jurusan=jurusan
12
13 class Siswa(Mahasiswa1, Mahasiswa2):
14     def __init__(self,nama,nim,alamat,jurusan):
15         Mahasiswa1.__init__(self,nama,nim)
16         Mahasiswa2.__init__(self,alamat,jurusan)
17
18 s=Siswa('Mikasa',135105,'wall rose','Informatika')
19 print('nim:',s.nim, 'nama:',s.nama, 'alamat:',s.alamat, 'jurusan:',s.jurusan)
```

TUGAS

Membuat laporan dengan format:

- ❖ Cover
- ❖ Isi (Teori+Tujuan+Diagram class+3 Kode program+Penjelasan+Output)
 - 1) Gambarkan diagram class untuk class ComputerPart
 - 2) Buatlah program untuk *Multilevel Inheritance* pada class ComputerPart
 - 3) Buatlah program untuk *Hierarchical Inheritance* pada class ComputerPart
 - 4) Buatlah program untuk *Multiple Inheritance* pada class ComputerPart
- ❖ Kesimpulan

Laporan dalam bentuk PDF (dikumpulkan di e-learning)

Program yang diupload ke github berjumlah 12 + Diagram class