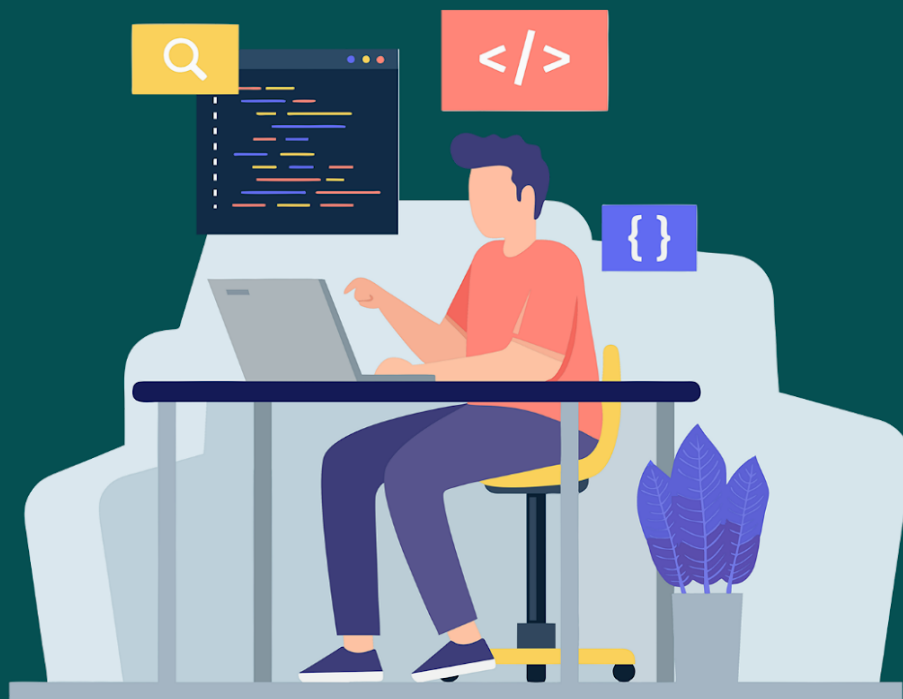


Modul Praktikum
Desain dan Pemrograman Berorientasi Objek

PEWARISAN — BAGIAN #1: ***SINGLE DAN MULTILEVEL INHERITANCE***



TIM ASISTEN PEMROGRAMAN
ANGKATAN 12

Departemen Pendidikan Ilmu Komputer
Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam
Universitas Pendidikan Indonesia
2022

Pewarisan

Pewarisan (*Inheritance*) adalah suatu konsep eksklusif pada pemrograman berorientasi objek di mana suatu kelas mampu menurunkan atau mewariskan sifat-sifatnya—dalam hal ini, **attribute** dan **method**—ke dalam kelas lain.

Ada dua istilah jenis kelas yang menyangkut konsep pewarisan ini, yaitu **superclass** dan **subclass**. Superclass adalah kelas yang mempunyai dan mewariskan sifat-sifat asalnya, sehingga kelas ini juga disebut sebagai kelas induk (*parent class*). Sementara itu, subclass adalah kelas yang “mendapatkan warisan” dari kelas lain, sehingga kelas ini juga disebut sebagai kelas anak (*child class*).

Sintaks dari pewarisan:

- **C++** menggunakan simbol titik dua (**:**) setelah nama kelas. *Modifier* ditulis setelah simbol dan sebelum nama kelas yang diturunkan.

```
class Adult : public Human
{
    // ...
}

class Child : private Human
{
    // ...
}
```

- **Java** dan **PHP** menggunakan kata **extends** setelah nama kelas. *Modifier* ditulis sebelum nama kelas turunan. Jika tidak menggunakan keterangan, maka kelas default **Java** adalah **private**, sementara untuk **PHP** adalah **public**.

```
public class Adult extends Human
{
    // ...
}

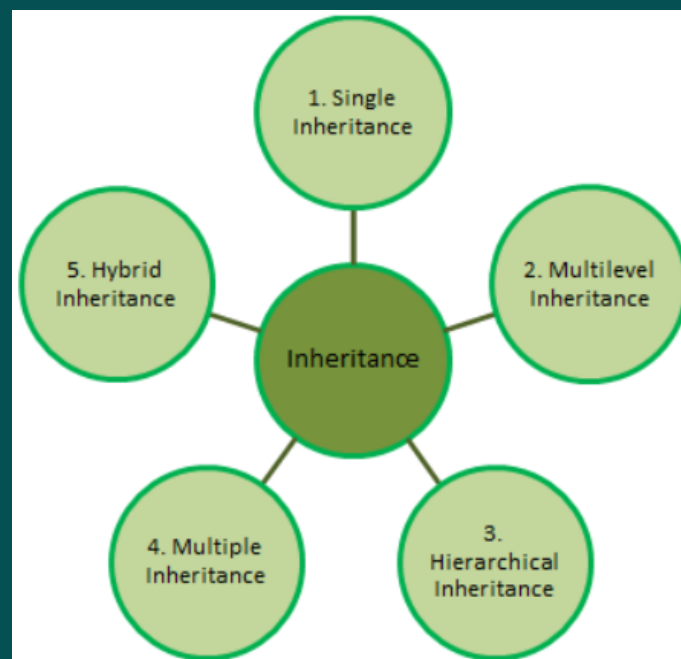
class Child extends Human
{
    // ...
}
```

- **Python**, menggunakan simbol kurung ((...)) setelah nama kelas, seperti penulisan parameter untuk method. *Modifier* ditulis mengikuti aturan garis bawah (_) bahasa ini.

```
class Adult(Human):  
    // ...  
  
class _Child(Human):  
    // ...
```

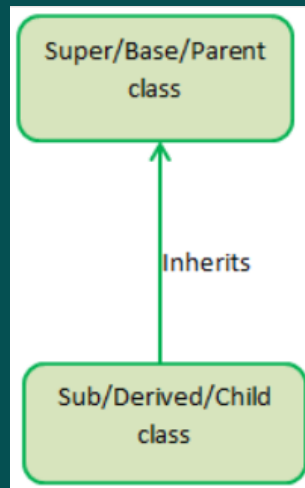
Ada lima jenis pewarisan, yaitu:

- Single
- Multilevel
- Hierarchical
- Multiple
- Hybrid



Jenis-Jenis Pewarisan: Single

Pewarisan tunggal merupakan salah satu konsep dasar dari pewarisan, sehingga penggunaannya di kehidupan nyata hanya untuk beberapa kasus. Pada pewarisan ini, **satu kelas induk** mewarisi **satu kelas anak**.



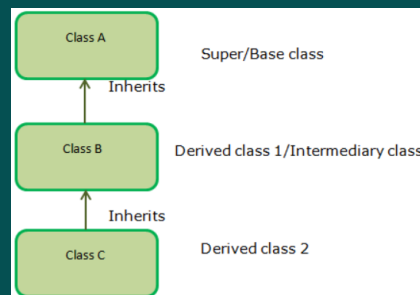
Contoh Program

```
1 // Base class. I don't have to explain all this, do I...?
2 class Human
3 {
4     void eat()
5     {
6         System.out.println(x: "It is eating!");
7     }
8
9     void sleep()
10    {
11        System.out.println(x: "It is sleeping... (zzz)");
12    }
13 }
14
15 // Derived class. It "extends" the existing property.
16 class Child extends Human
17 {
18     void play()
19     {
20         System.out.println(x: "It is playing!~");
21     }
22 }
23
24 public class Main
25 {
26     Run | Debug
27     public static void main(String[] args)
28     {
29         Child child = new Child();
30
31         // Let's try all behaviors.
32         child.eat();
33         child.play();
34         child.sleep();
35     }
36 }
```

Jenis-Jenis Pewarisan: Multilevel

Pewarisan bertingkat merupakan pengembangan dari pewarisan tunggal, di mana **satu kelas induk** mewarisi **satu kelas lain (anak)**, untuk kemudian diwariskan lagi

ke **satu kelas lain (cucu)**. Dengan kata lain, kelas cucu mempunyai sifat dari kelas induk dan kelas anak/tengah.



Contoh Program

```
1 // Base class. I don't have to explain all this, do I...?
2 class Human
3 {
4     void eat()
5     {
6         System.out.println(x: "It is eating!");
7     }
8
9     void sleep()
10    {
11        System.out.println(x: "It is sleeping... (zzz)");
12    }
13 }
14
15 // Derived class. It "extends" the existing property.
16 class Child extends Human
17 {
18     void play()
19     {
20         System.out.println(x: "It is playing!~");
21     }
22 }
23
24 // Another class, derived from the already derived class.
25 class Toddler extends Child
26 {
27     void crawl()
28     {
29         System.out.println(x: "It is crawling!!");
30     }
31 }
32
```

```
33 public class Main
34 {
35     Run | Debug
36     public static void main(String[] args)
37     {
38         Toddler toddler = new Toddler();
39
40         // Let's try all behaviors.
41         toddler.eat();
42         toddler.crawl();
43         toddler.play();
44         toddler.sleep();
45     }
46 }
```

Latihan

Buatlah program berbasis OOP menggunakan bahasa pemrograman C++, Java, Python, dan PHP yang mengimplementasikan konsep **Multi-level Inheritance** pada kelas - kelas tersebut:

- **Mahasiswa:** NIM, nama, jenis_kelamin, fakultas, prodi
- **Human:** NIK, nama, jenis_kelamin
- **SivitasAkademik:** asal_universitas, email_edu

Note.

- Boleh menambahkan properti/atribut baru
- Tampilkan data selengkap-lengkapnya dalam bentuk list/tabel
- Program dikumpulkan pada repository GitHub yang dibuat public dengan nama "LATIHAN2DPBO2023"
 - Hanya program pada branch Main yang akan dinilai dan diperiksa
 - Jika waktu pengumpulan sudah habis dan ingin mengupdate kode program, update pada branch lain karena mengupdate branch Main setelah waktu pengumpulan terlewat maka program tidak akan dinilai
- Struktur folder
 - CPP
 - program
 - screenshot
 - + Java
 - + Python
 - + PHP
 - README.md
- File README berisi desain program, penjelasan alur, dan dokumentasi saat program dijalankan (screenshot/screen record, pilih salah satu bahasa sebagai contoh)
- Submit link repository pada form berikut:
<https://forms.gle/rvb1hKxbQVuYNbhKA>



Penutup

Terima kasih atas kerja sama seluruh pihak yang membantu dalam penyusunan modul ini, semoga apa yang telah didapatkan bisa bermanfaat di masa yang akan datang.

Daftar Pustaka

Sukamto, Rosa A. (2018). *Pemrograman Berorientasi Objek*. Bandung: Modula.

Asisten Pemrograman 11. (2022). *Modul Desain dan Pemrograman Berorientasi Objek*.