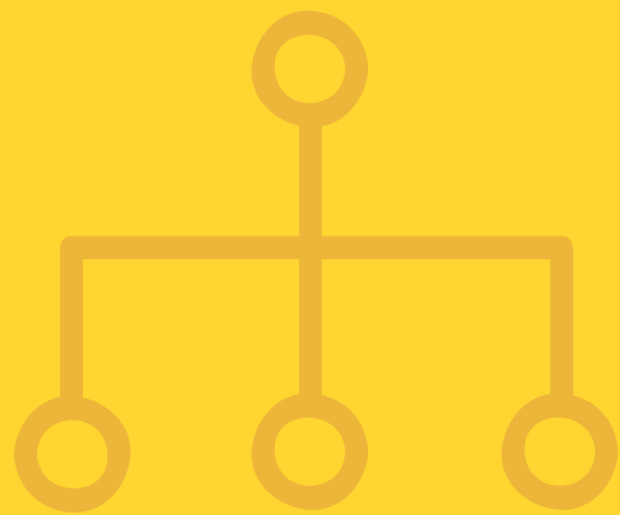




INHERITANCE & POLYMORFISME PYTHON

PEMROGRAMAN BERORIENTASI OBJEK

Apa itu Inheritance ?



Inheritance adalah konsep OOP dimana sebuah class dapat menurunkan property dan method yang dimilikinya kepada class lain.

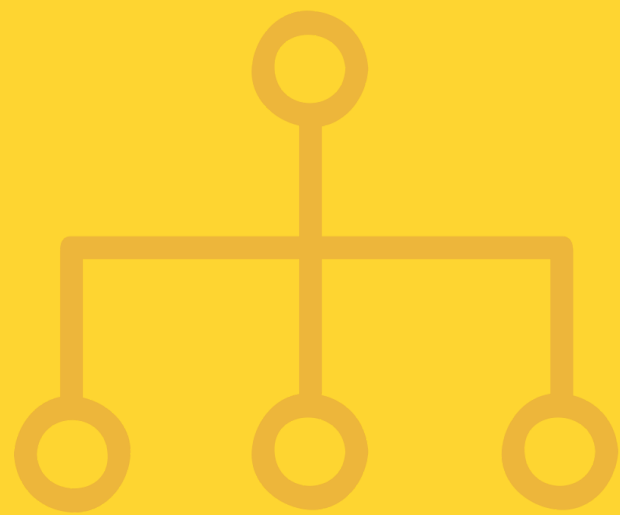
Konsep inheritance dipakai untuk memanfaatkan fitur code reuse, yakni menghindari terjadinya duplikasi kode program.

Dalam bahasa Indonesia, inheritance ini disebut juga sebagai pewarisan atau penurunan. Konsep inheritance membuat sebuah struktur atau hierarchy class dalam kode program. Class yang akan diturunkan bisa disebut sebagai class induk (parent class), super class, atau base class.

Sedangkan class yang menerima penurunan bisa disebut sebagai class anak (child class), sub class, derived class atau heir class.

Tidak semua property dan method class induk akan diturunkan. Property dan method dengan hak akses private tidak akan diturunkan kepada class anak. Hanya property dan method dengan hak akses protected dan public saja yang bisa diakses dari class anak.

Apa itu Inheritance ?



- **Kita dapat mendefinisikan** suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.
- **Penurunan sifat** ini bisa dilakukan secara bertingkat tingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.
- **Sub kelas** memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.
- **Dengan konsep pewarisan**, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.

```
graph TD; A[SUPERCLASS(PARENT)] --> B[SUBCLASS I]; A --> C[SUBCLASS II]
```

SUPERCLASS(PARENT)

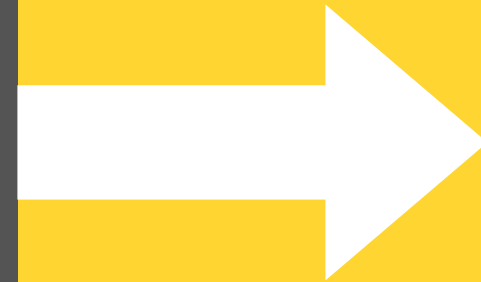
SUBCLASS I

SUBCLASS II

```
class Superclass:  
    def __init__(self):
```

```
class Subclass1(Superclass):  
    Pass
```

```
class Subclass2(Superclass):  
    Pass
```



```
class Superclass:  
    def __init__(self):  
        print ('Memanggil konstruktor super class')  
    def superclassMethod(self):  
        print ("Memanggil model super class")  
    def setAttr(self, attr):  
        Superclass.superclassAttr = attr  
    def getAttr(self):  
        print ('Atribut superclass : ', Superclass.superclassAttr)
```

```
class Subclass(Superclass):  
    def __init__(self):  
        print ('Memanggil konstruktor sub class')
```

```
    def subclassMethod(self):  
        print ('Memanggil metod sub class')
```

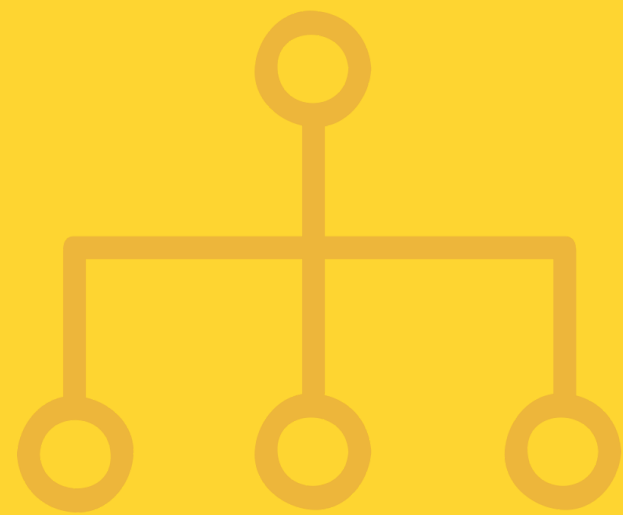
```
c = Subclass()  
c.subclassMethod()  
c.superclassMethod()  
c.setAttr(200)  
c.getAttr()
```

```
c = Superclass()
```

```
#check sub class  
if (issubclass(Subclass, Superclass)):  
    print ('true')
```



Apa itu Polimorfisme?



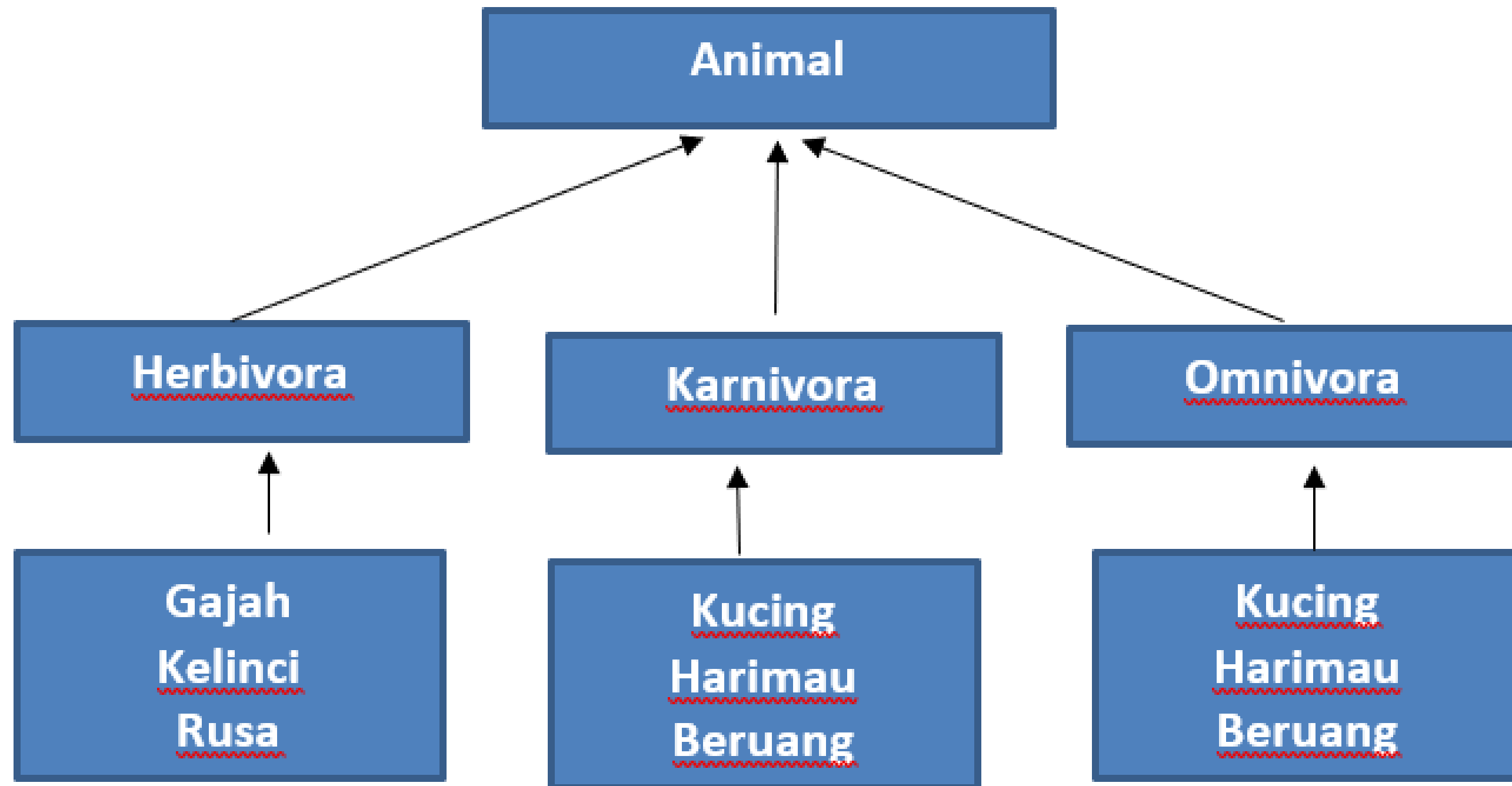
Poly artinya banyak, morfisme artinya bentuk.

Polimorfisme (bahasa inggris polymorphism) adalah sebuah prinsip dalam biologi di mana organisme atau spesies dapat memiliki banyak bentuk atau tahapan (stages).

Prinsip ini juga diadopsi pada pemrograman berorientasikan objek. Sehingga kita dapat definisikan sebagai berikut:

Polimorfisme dalam OOP adalah sebuah prinsip di mana class dapat memiliki banyak “bentuk” method yang berbeda-beda meskipun namanya sama.

“Bentuk” di sini dapat kita artikan: isinya berbeda, parameternya berbeda, dan tipe datanya berbeda.





Thank You!