

PEMROGRAMAN

BERORIENTASI

OBJEK

Pertemuan 6
Polymorphism

Konsep PBO

- Terdapat 3 konsep utama PBO pada python, yaitu:
 - Encapsulation
 - Inheritance
 - Polymorphism

Polymorphism

- *Polymorphism* berasal dari bahasa Yunani yang berarti banyak bentuk.
- Dalam konsep PBO memungkinkan digunakannya suatu interface yang sama untuk memerintah objek agar melakukan aksi atau tindakan yang mungkin secara prinsip sama namun secara proses berbeda.
- Polymorphism merupakan kemampuan suatu method untuk bekerja lebih dari satu tipe argumen, konsep ini sering disebut *overloading*.

Polymorphism

- *Polymorphism* adalah suatu object dapat memiliki berbagai bentuk, sebagai object dari class sendiri atau object dari superclassnya.
- Overloading: penggunaan satu nama untuk beberapa method yang berbeda (beda parameter).
- Overriding: terjadi ketika deklarasi method subclass dengan nama dan parameter yang sama dengan method dari superclass.

Keuntungan Polymorphism

- Dapat menggunakan kelas-kelas yang dibuat (sebagai super kelas) dan membuat kelas kelas baru berdasar superkelas tersebut dengan karakteristik yang lebih khusus dari behavior umum yang dimiliki superkelas.
- Dapat membuat super kelas yang hanya mendefinisikan behavior namun tidak memberikan implementasi dari metode-metode yang ada.
- Menghindari duplikasi object, yang dapat menciptakan class baru dari class yang sudah ada, sehingga tidak perlu menuliskan code dari nol ataupun mengulangnya, namun tetap bisa menambahkan attribute dan atau method unik dari class itu sendiri.

Metode Penggunaan Polymorphism

- Polymorphism dengan function

```
1  # Polymorphism
2  # simple example using len function
3
4  print(len("Polymorphism"))
5  print(len([0,1,2,3]))
6
7  ...
8  Menggunakan Fungsi len
9  Output :
10 12 (Tipe Data String)
11 4 (Tipe Data List)
12 ...
13
14 # using class
15 class jerman:
16     def ibukota(self):
17         print("Berlin adalah ibukota negara Jerman")
18
19 class jepang:
20     def ibukota(self):
21         print("Tokyo adalah ibukota negara Jepang")
22
23 negara1 = jerman()
24 negara2 = jepang()
25
26 for country in (negara1,negara2):
27     country.ibukota()
```

Metode Penggunaan Polymorphism

- Polymorphism dengan class

```
1  # Polymorphism dengan Class
2
3  class Kucing:
4      def __init__(self, nama, umur):
5          self.nama = nama
6          self.umur = umur
7
8      def bersuara(self):
9          print("Meow")
10
11
12 class Dog:
13     def __init__(self, nama, umur):
14         self.nama = nama
15         self.umur = umur
16
17     def bersuara(self):
18         print("Guk..guk...")
19
20 kucing1 = Kucing("Tom", 3)
21 anjing1 = Dog("Spike", 4)
22
23 for hewan in (kucing1, anjing1):
24     hewan.bersuara()
```

Metode Penggunaan Polymorphism

- Polymorphism dengan inheritance

```
1  # Polymorphism dengan Inheritance
2
3  class Burung:
4      def intro(self):
5          print("Di dunia ini ada beberapa type berbeda dari spesies burung")
6
7      def terbang(self):
8          print("Hampir semua burung dapat terbang, namun ada beberapa yang tidak dapat terbang")
9
10
11 class Elang(Burung):
12     def terbang(self):
13         print("Elang dapat terbang")
14
15
16 class BurungUnta(Burung):
17     def terbang(self):
18         print("Burung unta tidak dapat terbang")
19
20 obj_burung = Burung()
21 obj_elang = Elang()
22 obj_burung_unta = BurungUnta()
23
24 obj_burung.intro()
25 obj_burung.terbang()
26
27 obj_elang.intro()
28 obj_elang.terbang()
29
30 obj_burung_unta.intro()
31 obj_burung_unta.terbang()
```


Terima Kasih