



468



0



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

Polimorfisme

Pemrograman Berorientasi Objek
muhamad.soleh@iti.ac.id



START

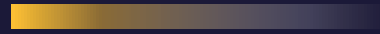


Polimorfisme

- Dari segi bahasa, *Polimorfisme* (bahasa inggris: *Polymorphism*) berasal dari dua kata bahasa Latin yakni *poly* dan *morph*.
- *Poly* berarti banyak, dan *morph* berarti bentuk. *Polimorfisme* berarti banyak bentuk.
- Sehingga dalam pemrograman berorientasi objek, *polimorfisme* merupakan kode program yang memiliki satu entitas tapi berbagai bentuk.



Energy



Weapons





468



0



01

Overriding

02

Overloading

03

Method
Resolution
Order

04

Magic
Method





368



2



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

01 Overriding





Polimorfisme Overriding

- Polimorfisme sangat erat kaitannya dengan konsep inheritance / pewarisan. Polimorfisme hanya akan terjadi jika dalam program terdapat konsep pewarisan.
- Polimorfisme terdiri dari 2 konsep, yaitu Overriding dan Overloading.
- **Overriding** biasa juga disebut sebagai **True polimorfisme**



Polimorfisme Overriding adalah Polimorfisme yang terjadi jika terdapat atribut atau fungsi dengan nama yang sama dalam superclass dan subclassnya atau sesama sub class nya.

Pemrograman Berorientasi Objek



Home



Game



Tools






Facts



Team



polimorfisme >  item.py >  Item >  __init__

```
1  #membuat kelas item
2  #superclass
3  class Item:
4      jumlahItem = 0 #inisialisasi jumlah hero
5      saveItem = [] #inisialisasi save hero
6      def __init__(self,n,h,d):
7          self.nama = n
8          self.harga = int(h)
9          self.deskripsi = d
10         Item.jumlahItem += 1
11
```



polimorfisme > item.py > Item > __init__




```
12 #subclass
13 class Movement(Item):
14     def efek1(self,nama):
15         self.efek0 = 40 #movement speed
16         self.efek1 = 0
17         self.efek2 = 0
18         self.pasif = False
19         if (nama == "Magic Shoes"):
20             self.efek1 = 0.1 #Cooldown Reduction
21         elif(nama == "Rapid Boots"):
22             self.efek0 += 40 #movement speed
23         else:
24             print("Item Movement yang lainnya")
```



polimorfisme > item.py > Defense

```
26 #subclass lain-nya
27 class Magic(Item):
28     def efek1(self,nama):
29         self.efek0 = 0
30         self.efek1 = 0
31         self.efek2 = 0
32         self.pasif = False
33         if (nama == "Holy Crystal"):
34             self.efek1 = 100 #Magic Power
35         else:
36             print("Item Magic yang lainnya")
```



polimorfisme >  item.py >  Item >  __init__

```
26  #subclass lain-nya
27  class Magic(Item):
28      |      pass
29
30  class Attack(Item):
31      |      pass
32
33  class Defense(Item):
34      |      pass
35
36  class Jungle(Item):
37      |      pass
38
```







368



2



STORE



LUCK ROYALE



CHARACTER



VAULT

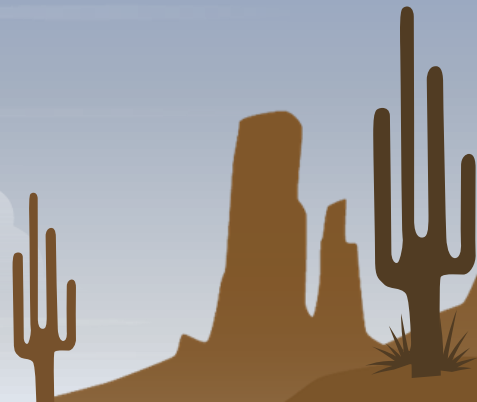


PET



COLLECTION

02 Overloading





368



2



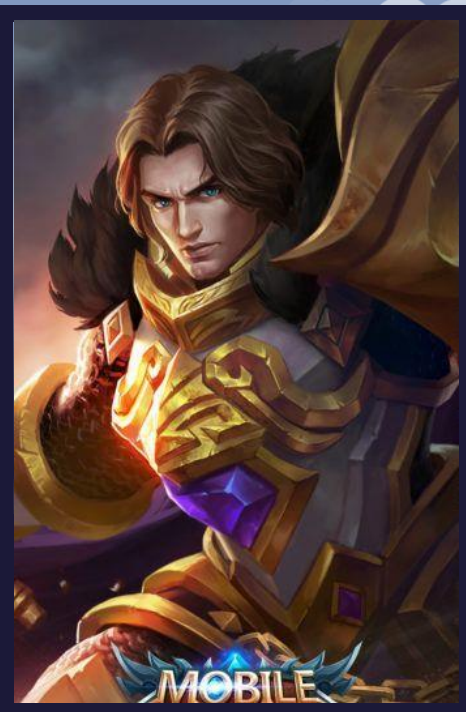
Overloading

- Polimorfisme Overloading terjadi ketika kita memiliki **method** dengan nama yang **sama**, namun jumlah **argumen** nya **berbeda** pada dua kelas class yang menerapkan konsep inheritance.
- Tujuan dari polimorfisme overloading adalah untuk memudahkan penggunaan/pemanggilan method dengan fungsionalitas yang mirip.





Overloading





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

polimorfisme > item.py > Item > __init__

```
1  #membuat kelas item
2  #superclass
3  class Item:
4      jumlahItem = 0 #inisialisasi jumlah hero
5      saveItem = [] #inisialisasi save hero
6      def __init__(self,n,h,d):
7          self.nama = n
8          self.harga = int(h)
9          self.deskripsi = d
10         Item.jumlahItem += 1
11
```





998



22



 STORE

 LUCK ROYALE

 CHARACTER

 VAULT

 PET

 COLLECTION

polimorfisme > item.py > Item > __init__

```
12  #subclass
13  class Movement(Item):
14      def efek1(self,nama):
15          self.efe0 = 40 #movement speed
16          self.efe1 = 0
17          self.efe2 = 0
18          self.pasif = False
19          if (nama == "Magic Shoes"):
20              self.efe1 = 0.1 #Cooldown Reduction
21          elif(nama == "Rapid Boots"):
22              self.efe0 += 40 #movement speed
23          else:
24              print("Item Movement yang lainnya")
```











998



22



-  STORE
-  LUCK ROYALE
-  CHARACTER
-  VAULT
-  PET
-  COLLECTION

polimorfisme >  item.py >  Defense

```
26 #subclass lain-nya
27 class Magic(Item):
28     def efek1(self,nama):
29         self.efek0 = 0
30         self.efek1 = 0
31         self.efek2 = 0
32         self.pasif = False
33         if (nama == "Holy Crystal"):
34             self.efek1 = 100 #Magic Power
35         else:
36             print("Item Magic yang lainnya")
```





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

polimorfisme > item.py > Defense

```
38 class Attack(Item):
39     def efek1(self,nama,harga):
40         self.efek0 = 0
41         self.efek1 = 0
42         self.efek2 = 0
43         self.pasif = False
44         if (nama == "Demond Hunter Sword" and harga == 2180):
45             self.efek1 = 35 #physical attack
46             self.efek2 = 0.25 #attack speed
47         else:
48             print("Item Attack yang lainnya")
```





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET

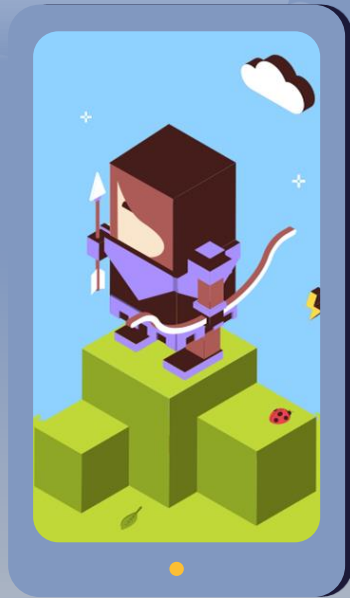


COLLECTION

polimorfisme >  item.py > ...

```
49
50 class Defense(Item):
51     pass
52
53 class Jungle(Item):
54     pass
55
56 class Roaming(Item):
57     pass
58
```





Ingat!

**OOP Python tidak boleh menggunakan
nama fungsi yang sama pada satu
class.**







368



2



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

Method 03 Resolution Order





368



2



MRO

- Method Resolution Order adalah sebuah property yang disediakan oleh python untuk menampilkan urutan pemanggilan fungsi pada kasus multiple inheritance.
- Method Resolution Order menggunakan fungsi `help(namaObjek)` untuk menampilkan urutan pemanggilan fungsi nya.





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

polimorfisme > hybrid.py > ...

```
1 class Marksman:
2     lateDamage = True
3     def __init__(self,n):
4         self.nama = n
5
6     def tampilkanNama(self):
7         print("nama", self.nama)
8
9 class Assassin:
10     highMobility = True
11     def __init__(self,n):
12         self.nama = n
13
```





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

polimorfisme > hybrid.py > ...

```
13     def __init__(self,n):
14         self.nama = n
15
16     def tampilkanNama(self):
17         print("nama", self.nama)
18
19 class Lesley(Marksman,Assassin):
20     hp = 1000
21
22 lesley = Lesley("Lesley")
23 print(lesley.highMobility)
24 print(lesley.lateDamage)
25 lesley.tampilkanNama()
```





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

polimorfisme > hybrid.py > ...

```
26
27 os.system('pause')
28 help(lesley)
29
30
31
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Help on Lesley in module __main__ object:

class Lesley(Marksman, Assassin)

| Lesley(n)

| Method resolution order:

| Lesley

| Marksman

| Assassin

| builtins.object

| Data and other attributes defined here:

-- More --





368



2



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

04 Magic Method





368



2



Magic Method

- Magic method adalah method yang diawali dan diakhiri dengan double underscore.
- Kita akan mempelajari magic method lainnya yang terdapat dalam Bahasa pemrograman python:
 - `def __repr__(self):`
 - `def __add__(self):`





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET

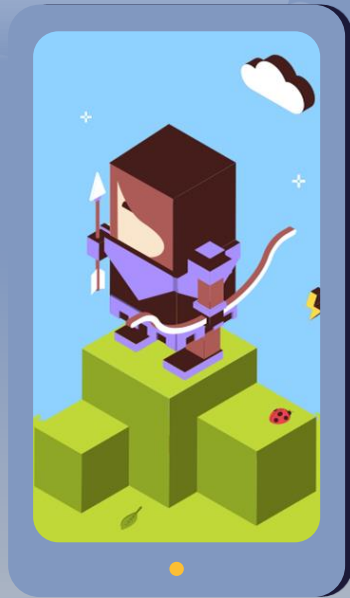


COLLECTION

polimorfisme > magic.py > ...

```
1 class Hero:
2     def __init__(self, namaHero, roleHero, hp):
3         self.nama = namaHero
4         self.role = roleHero
5         self.hp = hp
6
7     def __repr__(self):
8         print("ini adalah sebuah objek dengan atribut: ")
9         print("nama: ", self.nama)
10        return ("objek berhasil dibuat")
11
12 kagura = Hero("Kagura", "Mage", 5000)
13 print(kagura.nama)
14 print(kagura)
```





Ingat!

**Magif `__repr__`(self) harus
mengembalikan nilai string
(return string)**





998



22



STORE



LUCK ROYALE



CHARACTER



VAULT



PET

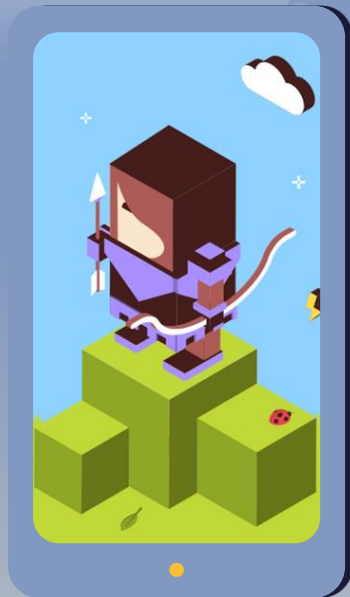


COLLECTION

polimorfisme > magic.py > ...

```
1 class Hero:
2     def __init__(self,namaHero,roleHero,hp):
3         self.nama = namaHero
4         self.role = roleHero
5         self.hp = hp
6
7     def __add__(self, obj):
8         print("penjumlahan atribut hp: ")
9         return (self.hp + obj.hp)
10
11 kagura = Hero("Kagura","Mage",5000)
12 uranus = Hero("Uranus","Tank",10000)
13 print(kagura + uranus)
```





Output:

```
e/polimorfisme/magic.py"  
penjumlahan atribut hp:  
15000  
PS C:\Users\DPKM\OneDrive\0.  
k\Code> 
```





854



10



THANKS!



Do you have any questions?

youremail@freepik.com

+91 620 421 838

yourcompany.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**

PLEASE KEEP THIS SLIDE FOR ATTRIBUTION



STORE



LUCK ROYALE



CHARACTER



VAULT



PET



COLLECTION

