

Pemrograman Berorientasi Objek

muhamad.soleh@iti.ac.id



Enkapsulasi



Home



Game



Tools



Facts



Team



Jenis-jenis Pewarisan

01

Akses Modifier

02

Protected

03

Private

04

Setter dan Getter



Home



Game



Tools



Facts



Team





01

Akses Modifier

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team







Konsep Enkapsulasi

- ❑ Enkapsulasi merupakan sebuah cara untuk mengatur method dan property pada class dengan menyembunyikan alur kerja dari class tersebut.
- ❑ Enkapsulasi berasal dari konsep kapsul. Dimana kapsul melindungi obat yang terdapat di dalam cangkang yang mudah larut.
- ❑ Manfaat dari enkapsulasi:
 1. Penyembunyian informasi
 2. Independen
 3. Perlindungan data-data dan informasi
 4. *Access modifier*



Home



Game



Tools



Facts



Team



Konsep Akses Modifier



- ❑ Access Modifier adalah sebuah “hak akses” yang diberikan kepada sebuah variabel/method
- ❑ Tujuan untuk menjaga integritas dari data tersebut ketika ingin diakses object lain.
- ❑ Hak akses sendiri diberikan oleh pembuat program, dengan adanya Access Modifier,
- ❑ Kita dapat membatasi resource-resource mana saja yang dapat diakses oleh object tertentu, turunannya, ataupun oleh method tertentu.



Home



Game



Tools



Facts



Team





Public, private, dan protected.



Akses Modifier	Sama Class	Sub Class	Sama Package	Beda Package
<i>Public</i>	Yes	Yes	Yes	Yes
<i>Protected</i>	Yes	Yes	Yes	No
<i>Private</i>	Yes	No	No	No

- ☐ Default akses modifier dari python adalah public
- ☐ Jadi, kode program yang sebelum nya kita buat memiliki tipe akses modifier public.



Home



Game



Tools



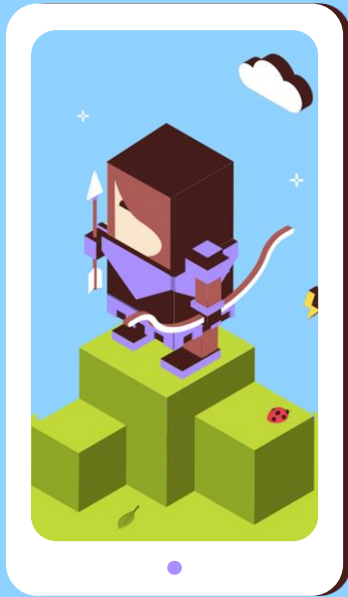
Facts



Team



Protected dan private



- ☐ Protected menggunakan single underscore
- ☐ Private menggunakan double underscore
- ☐ Perbedaan hak akses pada kedua akses modifier tersebut ada pada same package dan sub class



02

Protected

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



hero_protected.py > Hero > __init__

```
1  #membuat prototype Hero
2  class Hero:
3      #atribut kelas
4      jumlahHero = 0 #inisialisasi jumlah hero
5      #magic function konstruktor
6      def __init__ (self, n, r, h, m, ms,pa,ma):
7          self.nama = n
8          self.role = r
9          #protected
10         self._hp = int(h)
11         self.mana = int(m)
12         self.movSpeed = int(ms)
13         self.phyAtt = int(pa)
14         self.maAtt = int(ma)
```



hero_protected.py > Hero > __init__

```
15         self.item5 = []
16         self.gold = 0
17         Hero.jumlahHero += 1
18
19     def beliItem(self, item):
20         if (self.gold >= item.harga):
21             print(self.nama, "Membeli item", item.nama)
22             self.item5.append(item)
23             print(self.item5[0].nama, "sudah masuk slot")
24         else:
25             print(self.nama, "tidak memiliki cukup gold")
26
27     #parsing objek pada argumen fungsi
```



```
26
27     #parsing objek pada argumen fungsi
28     #proteccted
29     def _skill2 (self, musuh):
30         self.cd = 12 #menambahkan atribut cd pada hero,
31         print(self.nama , "skill 2 pada", musuh.nama)
32         efekMusuh = (60 * musuh.movSpeed // 100)
33         musuh.movSpeed -= efekMusuh
34         print("Movement Speed", musuh.nama, "menjadi:", \
35             musuh.movSpeed)
36         efekPlayer = self.mana - 35
37         self.mana = efekPlayer
38         print("Mana ", self.nama, " tersisa:", self.mana)
39
```

main_protected.py > landOfDown

```
1  import hero_protected
2  import item
3
4  def landOfDown():
5      print("Membuat beberapa objek hero sampai berhenti")
6      saveHero = [] #inisialisai tempat menyimpan hero
7      while(True):
8          print("Hero ke-", hero_protected.Hero.jumlahHero+1)
9          print("Masukan atribut hero / berhenti")
10         masukan = input()
11         if (masukan == "berhenti"):
12             break
13         else:
14             listMasukan = masukan.split(";")
```





main_protected.py > landOfDown

```
15         saveHero.append(hero_protected.Hero(listMasukan[0],
16         listMasukan[1],listMasukan[2], \
17         listMasukan[3],listMasukan[4],\
18         listMasukan[5],listMasukan[6]))
19
20     saveHero[0]._skill12(saveHero[1])
21
22     print("Membuat beberapa objek item sampai berhenti")
23     saveItem = [] #inisialisai tempat menyimpan hero
24     while(True):
25         print("Item ke-", item.Item.jumlahItem+1)
26         print("Masukan atribut item / berhenti")
27         masukan = input()
```





main_protected.py > landOfDown

```
28         if(masukan == "berhenti"):
29             break
30         else:
31             listMasukan = masukan.split("#")
32             saveItem.append(item.Item(listMasukan[0],\
33                                     listMasukan[1], listMasukan[2]))
34
35     print(saveItem[0].nama)
36     print("Cek beli item")
37     saveHero[0].beliItem(saveItem[0])
38     print("Parfming dulu dong")
39     print("hero membunuh monster jungle mendapatkan 800 gold")
40     saveHero[0].gold = 800
41     print("Cek beli item")
```



03

Private

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



hero_private.py > Hero

```
1  #membuat prototype Hero
2  class Hero:
3      #atribut kelas
4      jumlahHero = 0 #inisialisasi jumlah hero
5      #magic function konstruktor
6      def __init__ (self, n, r, h, m, ms,pa,ma):
7          self.nama = n
8          self.role = r
9          #private
10         self.__hp = int(h)
11         self.mana = int(m)
12         self.movSpeed = int(ms)
13         self.phyAtt = int(pa)
14         self.maAtt = int(ma)
```

hero_private.py > Hero

```
15         self.item5 = []
16         self.gold = 0
17         Hero.jumlahHero += 1
18
19     def beliItem(self, item):
20         if (self.gold >= item.harga):
21             print(self.nama, "Membeli item", item.nama)
22             self.item5.append(item)
23             print(self.item5[0].nama, "sudah masuk slot")
24         else:
25             print(self.nama, "tidak memiliki cukup gold")
26
27     #parsing objek pada argumen fungsi
28     #private
```





```
26
27     #parsing objek pada argumen fungsi
28     #private
29     def __skill2 (self, musuh):
30         self.cd = 12 #menambahkan atribut cd pada hero, bukan
31         print(self.nama , "skill 2 pada", musuh.nama)
32         efekMusuh = (60 * musuh.movSpeed // 100)
33         musuh.movSpeed -= efekMusuh
34         print("Movement Speed", musuh.nama, "menjadi:", \
35             musuh.movSpeed)
36         efekPlayer = self.mana - 35
37         self.mana = efekPlayer
38         print("Mana ", self.nama, " tersisa:", self.mana)
```



main_private.py > landOfDown

```
1 import hero_private
2 import item
3
4 def landOfDown():
5     print("Membuat beberapa objek hero sampai berhenti")
6     saveHero = [] #inisialisai tempat menyimpan hero
7     while(True):
8         print("Hero ke-", hero_private.Hero.jumlahHero+1)
9         print("Masukan atribut hero / berhenti")
10        masukan = input()
11        if (masukan == "berhenti"):
12            break
13        else:
14            listMasukan = masukan.split(";")
```



```
15         saveHero.append(hero_private.Hero(listMasukan[0],
16             listMasukan[1],listMasukan[2], \
17             listMasukan[3],listMasukan[4],\
18             listMasukan[5],listMasukan[6]))
19
20     saveHero[0].__skill2(saveHero[1])
21
22     print("Membuat beberapa objek item sampai berhenti")
23     saveItem = [] #inisialisai tempat menyimpan hero
24     while(True):
25         print("Item ke-", item.Item.jumlahItem+1)
26         print("Masukan atribut item / berhenti")
27         masukan = input()
28         if(masukan == "berhenti"):
```





main_private.py > landOfDown

```
28         if(masukan == "berhenti"):
29             break
30         else:
31             listMasukan = masukan.split("#")
32             saveItem.append(item.Item(listMasukan[0],\
33                                     listMasukan[1], listMasukan[2]))
34
35     print(saveItem[0].nama)
36     print("Cek beli item")
37     saveHero[0].beliItem(saveItem[0])
38     print("Parfming dulu dong")
39     print("hero membunuh monster jungle mendapatkan 800 gold")
40     saveHero[0].gold = 800
41     print("Cek beli item")
```





04

Setter & Getter

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





Setter dan Getter



- ❑ Setter dan getter merupakan sebuah method dengan akses modifier public yang bertujuan untuk melakukan interaksi pada variable dan fungsi yang memiliki hak akses private.
- ❑ Hal ini juga membuktikan bahwa variable atau fungsi private bisa di akses pada kelas yang sama.



Home



Game



Tools



Facts



Team





hero_private.py > Hero > beliItem

```
16         self.gold = 0
17         Hero.jumlahHero += 1
18
19     def setHP (self, hp):
20         self.__hp = hp
21
22     def getHP (self):
23         return self.__hp
24
25     def beliItem(self, item):
26         if (self.gold >= item.harga):
27             print(self.nama, "Membeli item", item.nama)
28             self.item5.append(item)
```





04

Private name mangling

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



Private name mangling



- ❑ Selain menggunakan setter dan getter, bias juga menggunakan private name mangling. Hal ini hanya berlaku di python
- ❑ Syntaks private name mangling adalah sebagai berikut:
- ❑ *namaObjek._namaKelas__namaAtribut/Fungsi*



Home



Game



Tools



Facts



Team



Private name mangling

main_private.py > landOfDown

```
14 listMasukan = masukan.split(";")
15 saveHero.append(hero_private.Hero(listMasukan[0],
16 listMasukan[1],listMasukan[2], \
17 listMasukan[3],listMasukan[4],\
18 listMasukan[5],listMasukan[6]))
19
20 saveHero[0].__skill12(saveHero[1])
21 #Private name mangling
22 print(saveHero[0]._Hero__hp)
23
24 print("Membuat beberapa objek item sampai berhenti")
25 saveItem = [] #inisialisai tempat menyimpan hero
26 while(True):
27     print("Item ke-", item.Item.jumlahItem+1)
```





Thanks!

Do you have any questions?

Muhamad.soleh@iti.ac.id

+62813 21301 463

Soleh.staff.iti.ac.id



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



Home



Game



Tools



Facts



Team

