

Pemrograman Berorientasi Objek

muhamad.soleh@iti.ac.id



Jenis-jenis Pewarisan



Home



Game



Tools



Facts



Team



Jenis-jenis Pewarisan

01

Package & Modul

02

Hubungan Package &
Modul

03

Jenis-Jenis Pewarisan

04

Implementasi Jenis-
Jenis Pewarisan



Home



Game



Tools



Facts



Team





01 Review Package

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





Package

- ❑ Semakin banyak kelas yang dibuat, semakin banyak pula baris kode yang dibuat.
- ❑ Semakin banyak baris kode, maka semakin rumit pula debugging error nya.
- ❑ Memisahkan file berdasarkan kelas nya, dan mengintegrasikan atau menghubungkan file-file tersebut merupakan solusi alternative.
- ❑ Package merupakan solusi dari kasus tersebut.
- ❑ Packaga terdiri dari beberapa file-file .py



Home



Game



Tools



Facts



Team





hero.py

item.py

main.py

hero.py > ...

```
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         self.hp = int(h)
10        self.mana = int(m)
11        self.movSpeed = int(ms)
12        self.phyAtt = int(pa)
13        self.maAtt = int(ma)
14        self.item5 = []
```





hero.py

item.py

main.py

hero.py > Hero > skill2

```
12         self.phyAtt = int(pa)
13         self.maAtt = int(ma)
14         self.item5 = []
15         self.gold = 0
16         Hero.jumlahHero += 1
17
18     def beliItem(self,item):
19         if (self.gold >= item.harga):
20             print(self.nama,"Membeli item",item.nama)
21             self.item5.append(item)
22             print(self.item5[0].nama,"sudah masuk slot")
23         else:
24             print(self.nama, "tidak memiliki cukup gold")
```



hero.py

item.py

main.py

hero.py > ...

```
25
26     #parsing objek pada argumen fungsi
27     def skill2 (self, musuh):
28         self.cd = 12 #menambahkan atribut cd pada hero,
29         print(self.nama , "skill 2 pada", musuh.nama)
30         efekMusuh = (60 * musuh.movSpeed // 100)
31         musuh.movSpeed -= efekMusuh
32         print("Movement Speed",musuh.nama, "menjadi:", \
33               | musuh.movSpeed)
34         efekPlayer = self.mana - 35
35         self.mana = efekPlayer
36         print("Mana ",self.nama, " tersisa:", self.mana)
37
```



 hero.pyitem.py 

main.py

item.py > ...

```
1 #membuat kelas item
2 #superclass
3 class Item:
4     jumlahItem = 0 #inisialisasi jumlah hero
5     def __init__(self,n,h,d):
6         self.nama = n
7         self.harga = int(h)
8         self.deskripsi = d
9         Item.jumlahItem += 1
10
11 #subclass
12 class Movement(Item):
13     def efek1(self,nama):
14         self.efek0 = 40 #movement speed
```



hero.py

item.py

x main.py

item.py > ...

```
15     self.efek1 = 0
16     self.efek2 = 0
17     self.pasif = False
18     if (nama == "Magic Shoes"):
19         self.efek1 = 0.1 #Cooldown Reduction
20     elif(nama == "Rapid Boots"):
21         self.efek0 += 40 #movement speed
22     else:
23         print("Item Movement yang lainnya")
24
25 #subClass Lain-nya
26 class Magic(Item):
27     pass
28
```



A white cloud with a brown outline and a dark brown shadow underneath.

hero.py

item.py 

main.py

item.py > ...

```
28
29 class Attack(Item):
30     pass
31
32 class Defense(Item):
33     pass
34
35 class Jungle(Item):
36     pass
37
38 class Roaming(Item):
39     pass
40
```



hero.py

item.py

main.py 

main.py > ...

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



hero.py

item.py

main.py 

main.py > ...

```
15     saveHero.append(hero.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"):
29             break
```



hero.py

item.py

main.py 

main.py > ...

```
30     else:  
31         listMasukan = masukan.split("#")  
32         saveItem.append(item.Item(listMasukan[0],\br/>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")  
42         saveHero[0].beliItem(saveItem[0])  
43  
44     landOfDown()  
45
```

Activate Windows
Get Started | Upgrade



+

+





01



Review Modul

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





Modul



- Modul merupakan file.py yang terdiri dari single class maupun multi class.
- Kumpulan dari setiap modul di simpan di dalam sebuah package.
- Best Practice nya, nama modul dibuat sesuai nama kelas, sedangkan nama package dibuat sesuai nama project.
- Pada kasus sebelumnya, kita memiliki 3 modul,yaitu: hero;item;main.



Home



Game



Tools



Facts

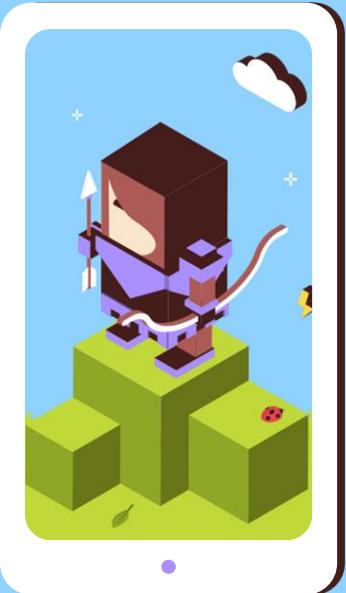


Team





Sintaks membuat objek dari modul (4 cara)



```
#nama modul (hero.py)
class Hero:
    pass

import hero
namaObjek = hero.Hero()

from hero import Hero
namaObjek = Hero()

from hero import Hero as h
namaObjek = h()

from hero import *
namaObjek = Hero()
```

Activate Windows



Implementasi syntaks membuat objek cara Modul 2

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





hero.py item.py main.py modul2.py X modul3.py modul4.py sintaks.py

modul2.py > landOfDown

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



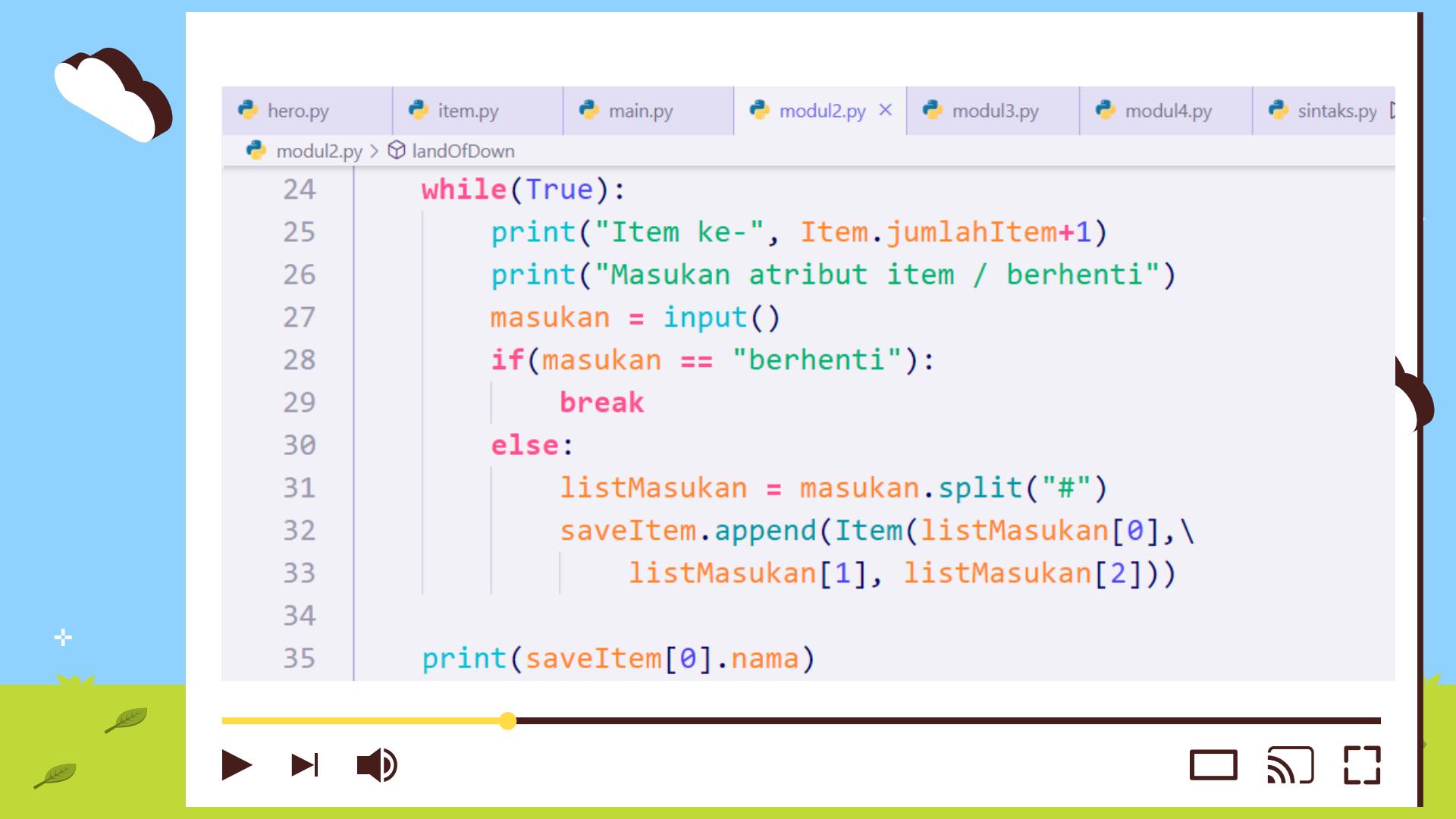


hero.py item.py main.py modul2.py X modul3.py modul4.py sintaks.py

modul2.py > landOfDown

```
13     else:  
14         listMasukan = masukan.split(";")  
15         saveHero.append(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 = [] #inisialisasi tempat menyimpan hero
```





A background illustration featuring a blue sky with a white and brown cloud on the left, green grass at the bottom, and small yellow flowers. A vertical brown tree trunk is on the right side.

```
hero.py item.py main.py modul2.py X modul3.py modul4.py sintaks.py D
modul2.py > landOfDown
```

```
24     while(True):
25         print("Item ke-", Item.jumlahItem+1)
26         print("Masukan atribut item / berhenti")
27         masukan = input()
28         if(masukan == "berhenti"):
29             break
30         else:
31             listMasukan = masukan.split("#")
32             saveItem.append(Item(listMasukan[0], \
33                                 listMasukan[1], listMasukan[2]))
34
35     print(saveItem[0].nama)
```



hero.py

item.py

main.py

modul2.py X

modul3.py

modul4.py

sintaks.py ▶

modul2.py > landOfDown

```
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")
42     saveHero[0].beliItem(saveItem[0])
43
44 landOfDown()
45
```



Implementasi syntaks membuat objek cara Modul 3

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





hero.py item.py main.py modul2.py modul3.py X modul4.py sintaks.py ▶

modul3.py > ⚙ landOfDown

```
1 from hero import Hero as h
2 from item import Item as i
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-", h.jumlahHero+1)
9         print("Masukan atribut hero / berhenti")
10        masukan = input()
11        if (masukan == "berhenti"):
12            break
```





hero.py item.py main.py modul2.py modul3.py × modul4.py sintaks.py

modul3.py > landOfDown

```
13     else:
14         listMasukan = masukan.split(";")
15         saveHero.append(h(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 = [] #inisialisasi tempat menyimpan hero
24         while(True):
```





hero.py

item.py

main.py

modul2.py

modul3.py X

modul4.py

sintaks.py

modul3.py > ⚙ landOfDown

```
25     print("Item ke-", i.jumlahItem+1)
26     print("Masukan atribut item / berhenti")
27     masukan = input()
28     if(masukan == "berhenti"):
29         break
30     else:
31         listMasukan = masukan.split("#")
32         saveItem.append(i(listMasukan[0], \
33                           listMasukan[1], listMasukan[2]))
34
35     print(saveItem[0].nama)
36     print("Cek beli item")
```



hero.py

item.py

main.py

modul2.py

modul3.py X

modul4.py

sintaks.py ▶ ▾

modul3.py > ...

```
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")
42     saveHero[0].beliItem(saveItem[0])
43
44 landOfDown()
45
```



Implementasi syntaks membuat objek cara Modul 4



Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



hero.py

item.py

main.py

modul2.py

modul3.py

modul4.py X

sintaks.py ▶

modul4.py > ⚙ landOfDown

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



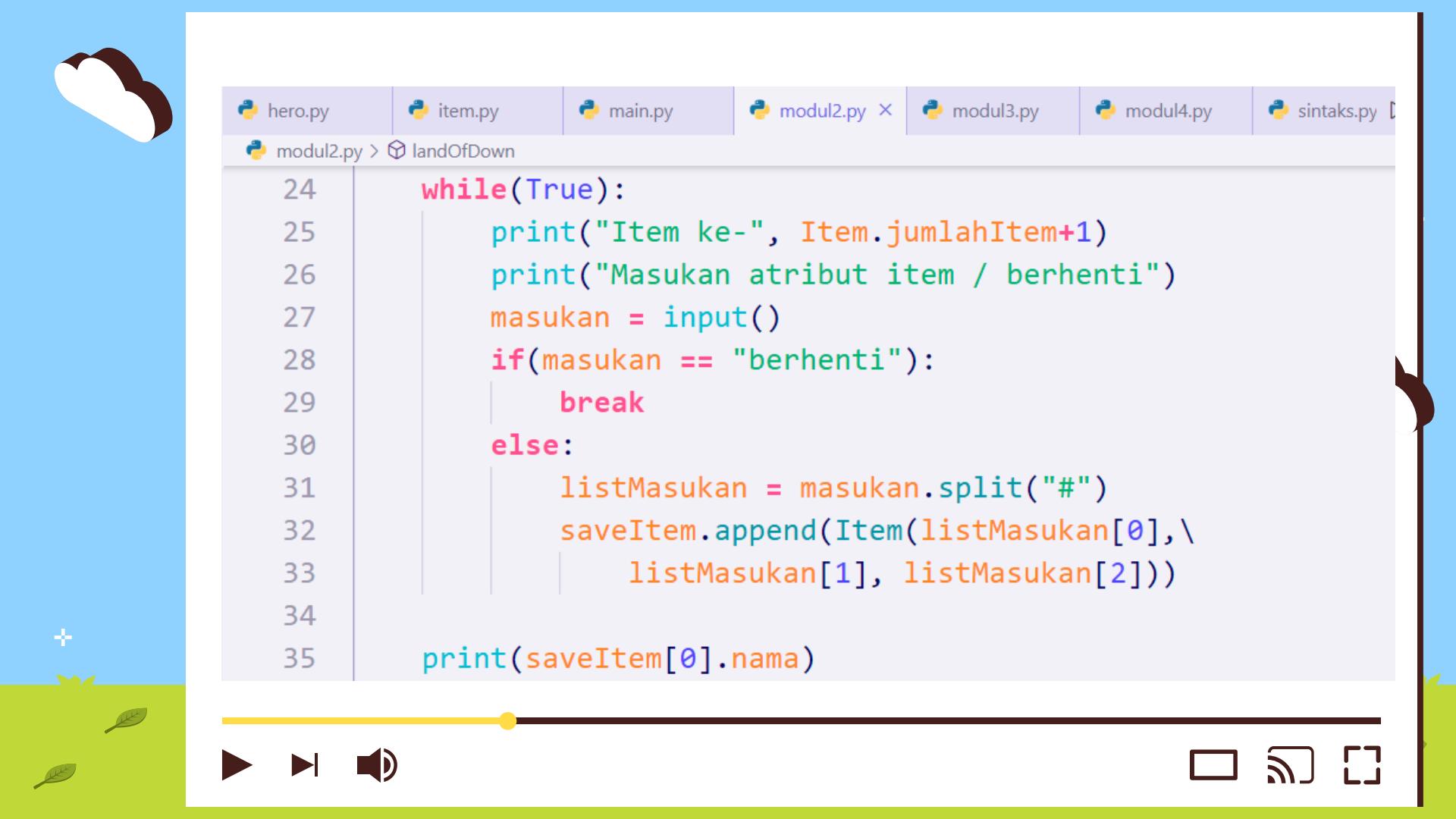


hero.py item.py main.py modul2.py X modul3.py modul4.py sintaks.py

modul2.py > landOfDown

```
13     else:  
14         listMasukan = masukan.split(";")  
15         saveHero.append(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 = [] #inisialisasi tempat menyimpan hero
```





A background illustration featuring a blue sky with a white and brown cloud on the left, green grass at the bottom, and small yellow flowers. A vertical brown tree trunk is on the right side.

```
hero.py item.py main.py modul2.py X modul3.py modul4.py sintaks.py D
modul2.py > landOfDown
```

```
24     while(True):
25         print("Item ke-", Item.jumlahItem+1)
26         print("Masukan atribut item / berhenti")
27         masukan = input()
28         if(masukan == "berhenti"):
29             break
30         else:
31             listMasukan = masukan.split("#")
32             saveItem.append(Item(listMasukan[0], \
33                                 listMasukan[1], listMasukan[2]))
34
35     print(saveItem[0].nama)
```



hero.py

item.py

main.py

modul2.py X

modul3.py

modul4.py

sintaks.py ▶

modul2.py > landOfDown

```
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")
42     saveHero[0].beliItem(saveItem[0])
43
44 landOfDown()
45
```





02

Hubungan Package & Modul

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



Modul +



Python
Modules

Python
Packages





Hubungan Package & Modul



- Package dan Modul saling berhubungan antara satu dengan yang lainnya.
- Setiap Package terdiri dari beberapa modul.
- Masing masing antar modul juga dapat berinteraksi.
- Sintaks import digunakan untuk berinterakhir antar modul dan package.
- Terdapat dua tipe import, yakni Absolute import dan Relative Import.



Home



Game



Tools



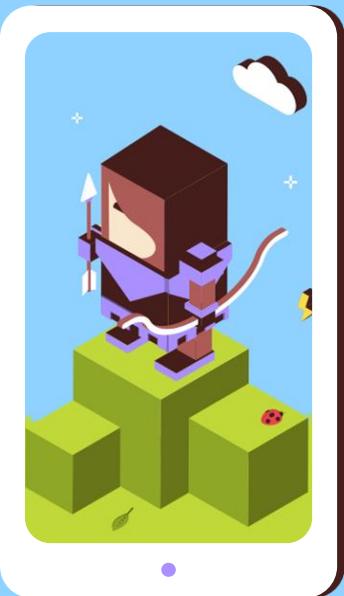
Facts



Team

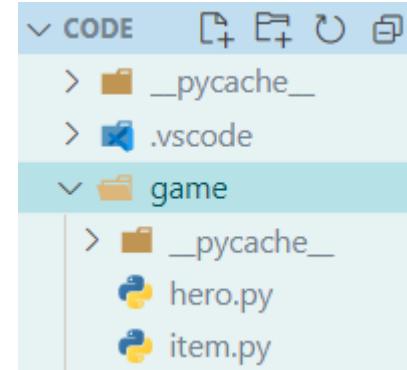


Sintaks absolute imports



```
#package = game
#modul = hero.py dan item.py
#main.py berada di Luar package

#membuat objek dari absolute import
#cara 1
import game.hero
namaObjek = game.hero.Hero()
#cara 2
from game.hero import Hero
namaObjek = Hero()
#cara 3
from game import hero
namaObjek = hero.Hero()
```





Implementasi syntaks membuat objek cara absolute package 1

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





package1.py > landOfDown

```
1 import game.hero
2 import game.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-", game.hero.Hero.jumlahHero+1)
9         print("Masukan atribut hero / berhenti")
10        masukan = input()
11        if (masukan == "berhenti"):
12            break
13        else:
14            listMasukan = masukan.split(";")
```





package1.py > landOfDown

```
15     saveHero.append(game.hero.Hero(listMasukan[0], \
16                                     listMasukan[1],listMasukan[2], \
17                                     listMasukan[3],listMasukan[4],\
18                                     listMasukan[5],listMasukan[6]))\n\n19\n20     saveHero[0].skill2(saveHero[1])\n21\n22     print("Membuat beberapa objek item sampai berhenti")\n23     saveItem = [] #inisialisasi tempat menyimpan hero\n24     while(True):\n25         print("Item ke-", game.item.Item.jumlahItem+1)\n26         print("Masukan atribut item / berhenti")\n27         masukan = input()
```

```
28     if(masukan == "berhenti"):
29         break
30     else:
31         listMasukan = masukan.split("#")
32         saveItem.append(game.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")
42         saveHero[0].beliItem(saveItem[0])
43
44 landOfDown()
```

Implementasi syntaks membuat objek cara absolute package 2

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



```
1 from game.hero import Hero
2 from game.item import Item
3
4 def landOfDown():
5     print("Membuat beberapa objek hero sampai berhenti")
6     saveHero = [] #inisialisasi tempat menyimpan hero
7     while(True):
8         print("Hero ke-", 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(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 = [] #inisialisasi tempat menyimpan hero  
24     while(True):  
25         print("Item ke-", Item.jumlahItem+1)  
26         print("Masukan atribut item / berhenti")  
27         masukan = input()  
28         if(masukan == "berhenti"):  
29             break
```

```
30     else:  
31         listMasukan = masukan.split("#")  
32         saveItem.append(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")  
42         saveHero[0].beliItem(saveItem[0])  
43  
44     landOfDown()  
45
```



Implementasi syntaks membuat objek cara absolute package 3

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



```
1 from game import hero
2 from game import item
3
4 def landOfDown():
5     print("Membuat beberapa objek hero sampai berhenti")
6     saveHero = [] #inisialisasi tempat menyimpan hero
7     while(True):
8         print("Hero ke-", hero.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.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 = [] #inisialisasi 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"):  
29             break
```

```
30     else:  
31         listMasukan = masukan.split("#")  
32         saveItem.append(item.Item(listMasukan[0],\br/>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")  
42         saveHero[0].beliItem(saveItem[0])  
43  
44     landOfDown()  
45
```



**Relative import
tidak di sarankan
untuk digunakan.
Relative import
menggunakan
operator . (dot)**

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team







03

Jenis-jenis pewarisan

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





Tipe Pewarisan OOP



- ❑ Dalam pemrograman berorientasi objek, terdapat 6 tipe pewarisan.
 - ❑ 1. single inheritance
 - ❑ 3. Multiple inheritance
 - ❑ 5. Hybrid inheritance
- ❑ Pada **class Item**, kita sudah mengimplementasikan Hierarchical inheritance.
- ❑ Perhatikan Visualisasi jenis-jenis pewarisan berikut, lalu cocokan dengan implementasi kodenya:



Home



Game



Tools

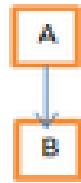


Facts



Team

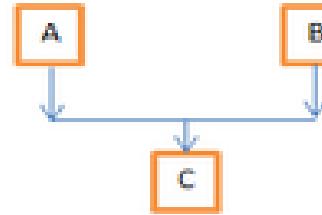




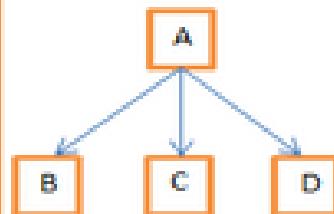
Single Inheritance



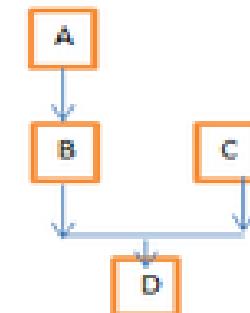
Multilevel Inheritance



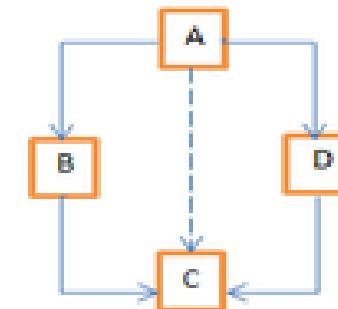
Multiple Inheritance



Hierarchical Inheritance



Hybrid Inheritance



Multipath Inheritance

Hierarchical Inheritance

```
game > item.py > Magic
1 #membut kelas item
2 #superclass
3 class Item:
4     jumlahItem = 0 #inisialisasi jumlah hero
5     def __init__(self,n,h,d):
6         self.nama = n
7         self.harga = int(h)
8         self.deskripsi = d
9         Item.jumlahItem += 1
10
11 #subclass
12 class Movement(Item):
13     def efek1(self,nama):
14         self.efek0 = 40 #movement speed
15         self.efek1 = 0
16         self.efek2 = 0
17         self.pasif = False
```

```
18     if (nama == "Magic Shoes"):
19         self.efek1 = 0.1 #Cooldown Reduction
20     elif(nama == "Rapid Boots"):
21         self.efek0 += 40 #movement speed
22     else:
23         print("Item Movement yang lainnya")
24
25 #subclass Lain-nya
26 class Magic(Item):
27     pass
28
29 class Attack(Item):
30     pass
31
32 class Defense(Item):
33     pass
34
```

Single Inheritance

game > 🐍 jungle.py > ...

```
1  class Jungle:  
2      def __init__(self,n,h,g):  
3          self.nama = n  
4          self.hp = h  
5          self.goldReward = g  
6  
7  class Lord(Jungle):  
8      def mati(self):  
9          print ("Lord has been slain")  
10
```

```
71  
72 class Jungle(Item):  
73     pass  
74  
75 class Roaming(Item):  
76     pass  
77
```



multilevel Inheritance

```
game > 🐍 emblem.py > ...
1  class AssassinEmblem:
2      physicalPen = 13.5
3      physicalAtt = 15
4      critLuck = 0.035
5      cdRed = 0.05
6      movSp = 0.02
7
8  class Subtalent(AssassinEmblem):
9      def __init__(self,n):
10         self.nama = n #Bravery, Mastery, etc
11
12 class Talents(Subtalent):
13     def __init__(self,n):
14         self.nama = n #Bounty Hunter, High and Dry, Killing Spree
15
```



Assassin Emblem

◇ Attributes

- PHYSICAL PEN:

+13.5

- PHYSICAL ATK:

+15

- CRIT. LUCK:

+3.5%

- CD REDUCTION:

+5%

- MOVEMENT SPD:

+2%

◇ Subtalents

- Bravery



- Invasion



- Mastery



- Fatal



- Agility



- Bloodthirst



◇ Talents

- Bounty Hunter



- High and Dry



- Killing Spree



Multiple Inheritance

```
game > 🐍 hybrid.py > ...
1  class Marksman:
2      lateDamage = True
3
4  class Assassin:
5      highMobility = True
6
7  class Lesley(Marksman,Assassin):
8      def __init__(self):
9          self.nama = "Lesley"
10
11 lesley = Lesley()
12 print(lesley.highMobility)
13 print(lesley.lateDamage)
14
```

```
C:\Users\PC DESAIN\OneDrive\0.
ESAIN\AppData\Local\Programs\Py
2 Genap\Pemrograman Berorientas
True
True
```



Lesley
General Rosa



403



8128



0



Skins



Role:

Marksman/Assassin

Specialty:

Reap/Burst

Durability

Offense

Ability Effects

Difficulty



General Rosa

BUY

IN USE

- Military Themed model!
- All new display animation!
- All new skill effect color changes!



Guides



Story



Live



Equip



04

Implementasi jenis pewarisan

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



Implementasi Class Jungle

pewarisan.py > ...

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



Implementasi Class Jungle

```
pewarisan.py > ...
16         saveHero.append(Hero(listMasukan[0], \
17             listMasukan[1],listMasukan[2], \
18             listMasukan[3],listMasukan[4],\
19             listMasukan[5],listMasukan[6]))
20
21     saveHero[0].skill2(saveHero[1])
22
23     print("Membuat beberapa objek item sampai berhenti")
24     saveItem = [] #inisialisasi tempat menyimpan hero
25     while(True):
26         print("Item ke-", Item.jumlahItem+1)
27         print("Masukan atribut item / berhenti")
28         masukan = input()
29         if(masukan == "berhenti"):
30             break
```



Implementasi Class Jungle



pewarisan.py > landOfDown

```
31     else:  
32         listMasukan = masukan.split("#")  
33         saveItem.append(Item(listMasukan[0], \  
34                             listMasukan[1], listMasukan[2]))  
35  
36         print(saveItem[0].nama)  
37         print("Cek beli item")  
38         saveHero[0].beliItem(saveItem[0])  
39         print("Parfming dulu dong")  
40         lord = Lord("Sanctuary Lord", 45, 800)  
41         saveHero[0].farming(lord)  
42         print("Cek beli item")  
43         saveHero[0].beliItem(saveItem[0])  
44  
45     landOfDown()  
46
```





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

