

Pemrograman Berorientasi Objek

muhamad.soleh@iti.ac.id



Konsep Pewarisan



Home



Game



Tools



Facts



Team



Konsep Pewarisan / inheritance

01

Konsep Pewarisan

02

Superclass dan subclass

03

More Class

04

Package



Home



Game



Tools



Facts



Team





01

Konsep Pewarisan

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



Pewarisan: Inheritance⁺





Konsep Pewarisan



- ❖ Konsep pewarisan merupakan salah satu dari empat konsep dasar PBO.
- ❖ Konsep Pewarisan terinspirasi dari konsep pewarisan sifat genetika pada makhluk hidup.
- ❖ Seorang anak (*childern*) yang terlahir dari induknya akan memiliki sifat-sifat yang diturunkan dari induknya (*parent*), baik ibu maupun ayah.
- ❖ Begitu juga dalam konsep PBO. Parent disebut sebagai superclass, sedangkan children disebut subclass.



Home



Game



Tools



Facts



Team





Sintaks membuat Superclass dan Subclass

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



Sintaks membuat Superclass dan Subclass

```
1 class Superclass:  
2     pass  
3  
4 class SubClass(Superclass):  
5     pass
```



Home



Game



Tools




Facts






Team





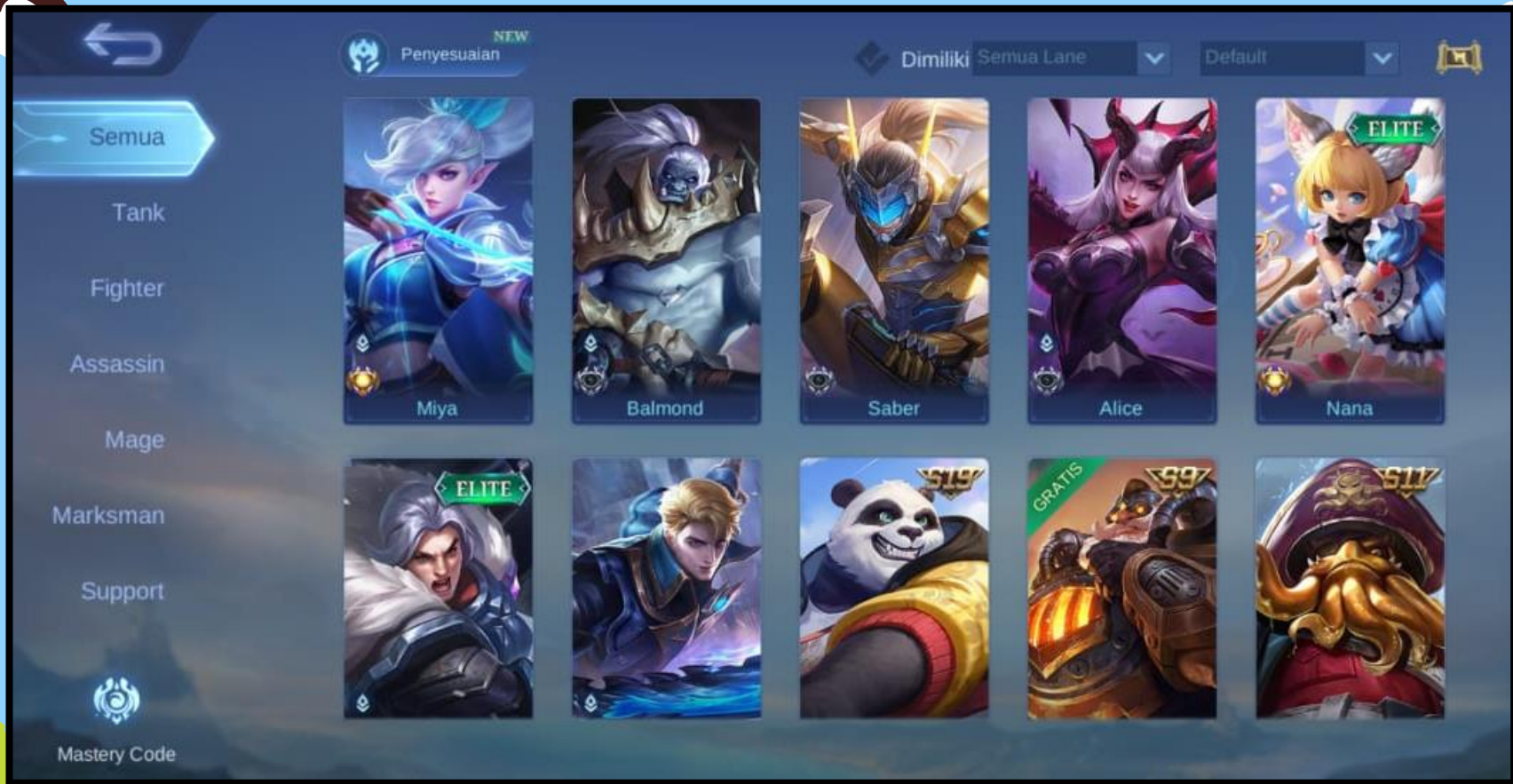
```
1 class Hero:
2     def __init__(self,n):
3         self.nama = n
4
5 class Marksman(Hero):
6     pass
7
8 class Assassin(Hero):
9     pass
10
11 class Tank(Hero):
12     pass
13
```





```
14 class Fighter(Hero):  
15     pass  
16  
17 class Mage(Hero):  
18     pass  
19  
20 class Support(Hero):  
21     pass  
22  
23 hanabi = Marksman("Hanabi")  
24 print(hanabi.nama)  
25
```





02

Superclass dan subclass

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team



Superclass dan Subclass

ALL SIBLINGS IN MOBILE LEGENDS





Superclass dan subclass



- Superclass merupakan parent class / kelas induk
- Subclass merupakan child class / kelas anak
- Superclass memuat informasi umum / general terkait objek
- Subclass memuat informasi spesifik dari objek yang menjadi kelas induknya
- Superclass mewarisi semua atribut dan fungsi kepada subclass nya.
- Subclass dapat mengakses semua atribut dan fungsi yang ada pada superclass nya, tetapi tidak bisa sebaliknya.



Home



Game



Tools



Facts



Team





Sintaks superclass dan subclass dengan atribut dan fungsi

Pemrograman Berorientasi Objek



Home



Game



Tools



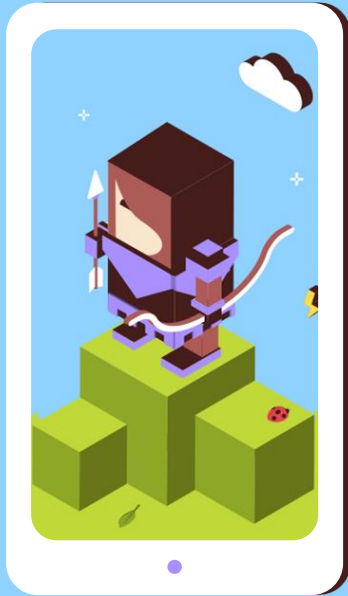
Facts



Team



Sintaks membuat atribut kelas



```
class Induk:
    def __init__(self, dataAtribut):
        self.namaAtribut = dataAtribut

    def namaFungsi(self):
        print("cetak nama atribut", self.namaAtribut)

class Anak(Induk):
    pass

objekAnak = Anak("dataAtribut")
print(objekAnak.namaAtribut)
print(objekAnak.namaFungsi())
```





Whoa!

Superclass dan Subclass studi
kasus Item



Home



Game



Tools




Facts




Team





```
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 = []
```





```
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")
25
```





```
26      #parsing objek pada argumen fungsi
27      def skill2 (self, musuh):
28          self.cd = 12 #atribut cd pada hero, bukan skill
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
```





```
38  #membuat kelas item
39  #superclass
40  class Item:
41      jumlahItem = 0 #inisialisasi jumlah hero
42      def __init__(self,n,h,d):
43          self.nama = n
44          self.harga = int(h)
45          self.deskripsi = d
46          Item.jumlahItem += 1
47
```





```
48 #subclass
49 class Movement(Item):
50     def efek1(self,nama):
51         self.efek0 = 40 #movement speed
52         self.efek1 = 0
53         self.efek2 = 0
54         self.pasif = False
55         if (nama == "Magic Shoes"):
56             self.efek1 = 0.1 #Cooldown Reduction
57         elif(nama == "Rapid Boots"):
58             self.efek0 += 40 #movement speed
59         else:
60             print("Item Movement yang lainnya")
```





```
62 def landOfDown():
63     print("Membuat beberapa objek hero sampai berhenti")
64     saveHero = [] #inisialisai tempat menyimpan hero
65     while(True):
66         print("Hero ke-", Hero.jumlahHero+1)
67         print("Masukan atribut hero / berhenti")
68         masukan = input()
69         if (masukan == "berhenti"):
70             break
71         else:
72             listMasukan = masukan.split(";")
73             saveHero.append(Hero(listMasukan[0],listMasukan[1],\
74                 listMasukan[2],listMasukan[3],listMasukan[4],\
75                 listMasukan[5],listMasukan[6]))
```





```
77     saveHero[0].skill2(saveHero[1])
78
79     print("Membuat beberapa objek item dari subclass")
80     saveItem = [] #inisialisai tempat menyimpan item
81     while(True):
82         print("Item ke-", Item.jumlahItem+1)
83         print("Masukan atribut item / berhenti")
84         masukan = input()
85         if(masukan == "berhenti"):
86             break
87         else:
88             listMasukan = masukan.split("#")
89             saveItem.append(Movement(listMasukan[0],listMasukan[1],\
90                                     listMasukan[2]))
```





```
91  
92     print(saveItem[0].nama)  
93     print("Cek beli item")  
94     saveHero[0].beliItem(saveItem[0])  
95     print("Parfming dulu dong")  
96     print("hero membunuh monster jungle mendapatkan 800 gold")  
97     saveHero[0].gold = 800  
98     print("Cek beli item")  
99     saveHero[0].beliItem(saveItem[0])  
100  
101     landOfDown()  
102
```





Penyesuaian



2289



10672



12



Attack

Magic

Defense

Movement

Jungle

Roaming

Emblem

Battle Spell

Build

Equipment

Efek ▾

Skin Paint

Quick Chat

Tutorial ▾

Reposisi Skill



Demon Shoes
Mana Regen
Cepat

720



Rapid Boots
Speed Up
ketika
Meninggalkan
Pertarungan

850



Swift Boots
Attack Speed

710



Arcane Boots
Magic
Penetration

690



Magic Shoes
Mengurangi
Cooldown

710



Tough Boots
Mengurangi
Durasi Efek
Crowd Control

700



Warrior Boots
Stack Physical
Defense

720



Boots

250

Build



850



250

Rapid Boots

+80 Movement SPD

Pasif Unik-Side Effect:
Memberikan atau menerima
Damage akan mengurangi
Movement Speed Hero sebanyak



Jalur Build



Introduksi

An isometric illustration of a small house with a brown roof and white walls, sitting on a green pedestal. A yellow butterfly is flying in the top left, and a dragonfly is in the middle right. The background is a light blue sky with white clouds and small white plus signs.

03

More Class

Pemrograman Berorientasi Objek



Home



Game



Tools



Facts



Team





Objek yang berbeda



- ❑ Pemrograman berorientasi objek memandang setiap komponen program dipandang menjadi sebuah objek.
- ❑ Objek yang dibuat berbagai macam tergantung kebutuhan program yang dibuat.
- ❑ Setiap objek yang berbeda atribut dan fungsi nya dibangkitkan dari kelas yang berbeda pula.
- ❑ Jadi, kita membutuhkan lebih banyak kelas dalam program yang kita buat.



Home



Game



Tools




Facts




Team







```
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 = []
```





```
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")
25
```





```
26      #parsing objek pada argumen fungsi
27      def skill2 (self, musuh):
28          self.cd = 12 #atribut cd pada hero, bukan skill
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
```





```
38  #membuat kelas item
39  #superclass
40  class Item:
41      jumlahItem = 0 #inisialisasi jumlah hero
42      def __init__(self,n,h,d):
43          self.nama = n
44          self.harga = int(h)
45          self.deskripsi = d
46          Item.jumlahItem += 1
47
```





```
48 #subclass
49 class Movement(Item):
50     def efek1(self,nama):
51         self.efek0 = 40 #movement speed
52         self.efek1 = 0
53         self.efek2 = 0
54         self.pasif = False
55         if (nama == "Magic Shoes"):
56             self.efek1 = 0.1 #Cooldown Reduction
57         elif(nama == "Rapid Boots"):
58             self.efek0 += 40 #movement speed
59         else:
60             print("Item Movement yang lainnya")
```





```
61
62  #subclass lain-nya
63  class Magic(Item):
64      |      pass
65
66  class Attack(Item):
67      |      pass
68
69  class Defense(Item):
70      |      pass
71
```






```
71
72 class Jungle(Item):
73     pass
74
75 class Roaming(Item):
76     pass
77
78 def landOfDown():
79     print("Membuat beberapa objek hero sampai berhenti")
80     saveHero = [] #inisialisai tempat menyimpan hero
```






```
81     while(True):
82         print("Hero ke-", Hero.jumlahHero+1)
83         print("Masukan atribut hero / berhenti")
84         masukan = input()
85         if (masukan == "berhenti"):
86             break
87         else:
88             listMasukan = masukan.split(";")
89             saveHero.append(Hero(listMasukan[0],listMasukan[1],\
90                                 listMasukan[2],listMasukan[3],listMasukan[4],\
91                                 listMasukan[5],listMasukan[6]))
92
```





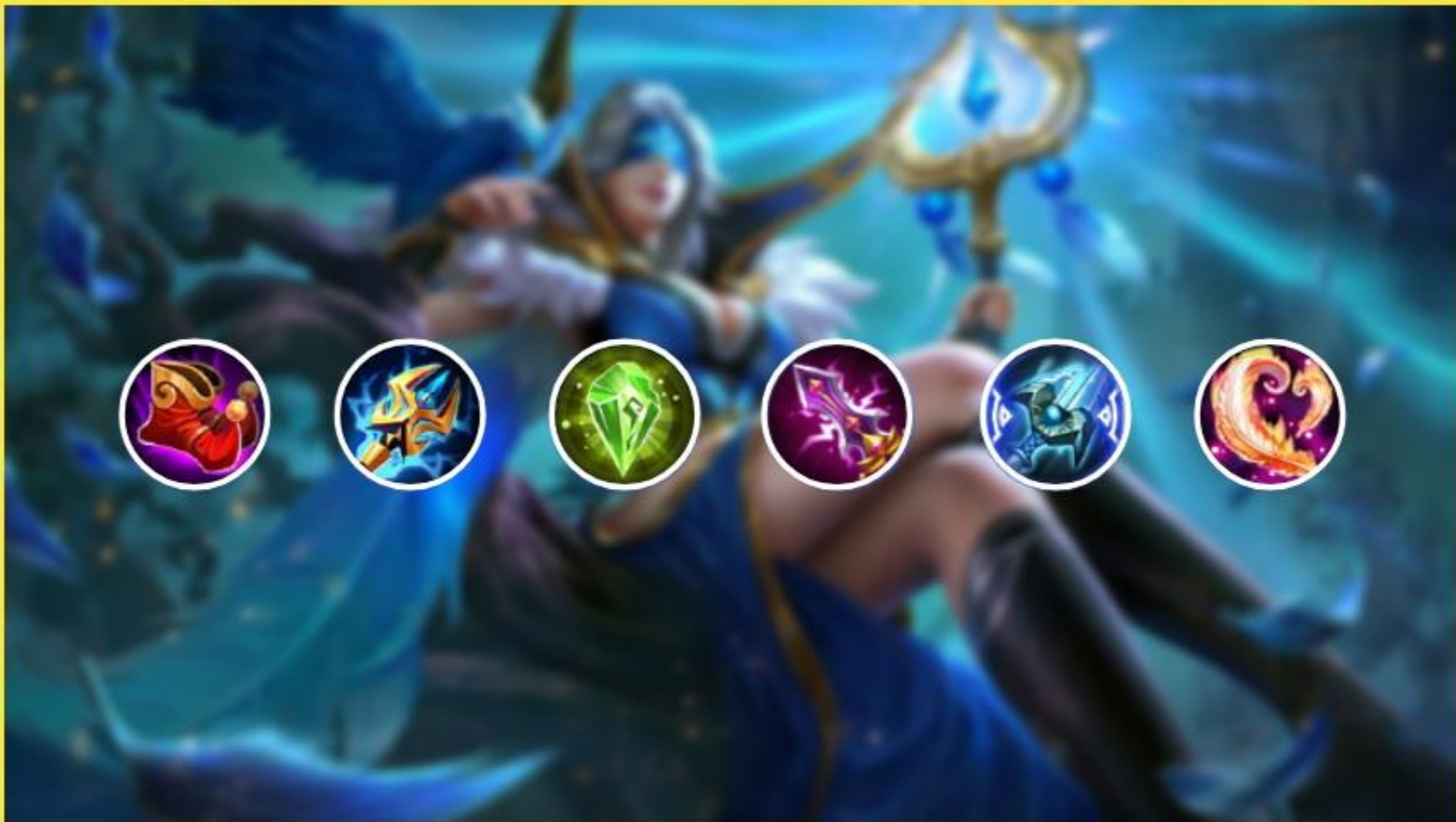
```
93     saveHero[0].skill2(saveHero[1])
94
95     print("Membuat beberapa objek item sampai berhenti")
96     saveItem = [] #inisialisai tempat menyimpan hero
97     while(True):
98         print("Item ke-", Item.jumlahItem+1)
99         print("Masukan atribut item / berhenti")
100        masukan = input()
101        if(masukan == "berhenti"):
102            break
103        else:
104            listMasukan = masukan.split("#")
105            saveItem.append(Item(listMasukan[0],listMasukan[1],\
106                                listMasukan[2]))
```





```
107
108     print(saveItem[0].nama)
109     print("Cek beli item")
110     saveHero[0].beliItem(saveItem[0])
111     print("Parfming dulu dong")
112     print("hero membunuh monster jungle mendapatkan 800 gold")
113     saveHero[0].gold = 800
114     print("Cek beli item")
115     saveHero[0].beliItem(saveItem[0])
116
117 landOfDown()
118
```







04 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





Whoa!

More Complicated !
but
More Interesting 😊



Home



Game



Tools



Facts



Team





Ayo Buat Game !

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

