



UNIVERSITAS
GADJAH MADA

W12 – Pemrograman Lanjut

Oleh :

Alim Satria Fi'i Wijaya Kusuma (21/483503/SV/20304)

Video



```
class Hewan :
    def __init__(self,nama) :
        self.nama = nama;
    def gerak(self) :
        print("gerak-gerak!!")

class HewanDarat(Hewan) :
    def __init__(self,nama,kaki=0) :
        super().__init__(nama)
        self.kaki = kaki
    def gerak(self) :
        print(self.nama," gerak didarat dengan kaki ",self.kaki)

class HewanAir(Hewan) :
    def __init__(self,nama,sirip = "kecil"):
        super().__init__(nama)
        self.sirip = sirip
    def gerak(self) :
        print(self.nama, "gerak di air dengan sirip ",self.sirip)

hewan = Hewan("Cacing")
kambing = HewanDarat("kambing",kaki=4)
hiu = HewanAir('Hiu',sirip='lebar')

hewan.gerak()
kambing.gerak()
hiu.gerak()
```

```
gerak-gerak!!
kambing gerak didarat dengan kaki 4
Hiu gerak di air dengan sirip lebar
```

Video



- Mencari properties dan method
- bawah ke atas

```
class level_1 :  
    var = 100  
    def fun(self) :  
        return 101  
class level_2(level_1) :  
    def fun(self) :  
        return 201  
class level_3(level_2) :  
    pass  
  
obj = level_3()  
  
print(obj.var,obj.fun())
```

100 201

- kiri ke kanan

```
class level_1 :  
    var = 100  
    def fun(self) :  
        return 101  
class level_2 :  
    var = 200  
    def fun(self) :  
        return 201  
class level_3(level_1,level_2) :  
    pass  
  
obj = level_3()  
  
print(obj.var,obj.fun())
```

100 101



Video

- MRO

```
class top :  
    def m_top(self) :  
        print('top')  
  
class middle(top):  
    def m_middle(self) :  
        print('middle')  
  
class bottom(middle,top) :  
    def m_bottom(self) :  
        print('bottom')  
  
obj = bottom()  
obj.m_bottom()  
obj.m_middle()  
obj.m_top()
```

```
bottom  
middle  
top
```

- Multiple inheritance [modul]

```
class satu :  
    var      = "satu"  
    varSatu = "satusatu"  
    def coba(self) :  
        return "SATU"  
  
class dua :  
    var      = "dua"  
    varDua  = "duadua"  
    def coba(self) :  
        return "DUA"  
  
class sub(satu,dua) :  
    pass  
  
obj = sub()  
print(obj.var,obj.varSatu,obj.varDua,obj.coba())
```

satu satusatu duadua SATU

Multiple inheritance merupakan kondisi dimana dalam pemanggilan kelas terdapat kelas dengan level yang setara. pada umumnya, pemanggilan kelas dilakukan dari bawah ke atas, namun apabila tidak menyalahi MOR maka pemanggilan dapat di lakukan dari atas ke bawah dengan membuat class yang diisikan dengan kelas kelas lain sesuai urutan.

- isubclass dan isinstance [modul]

```
print(issubclass(sub,satu))  
print(issubclass(sub,dua))  
print(isinstance(obj,satu))  
print(isinstance(obj,dua))
```

```
True  
True  
True  
True
```

Perintah `issubclass()` digunakan untuk memeriksa apakah sebuah kelas merupakan sebuah subkelas dari kelas lain.

Perintah `isinstance()` digunakan untuk memeriksa apakah sebuah objek merupakan suatu instance dari kelas

- isoperator [modul]

```
String_1 = "Mary had a little "  
String_2 = "Mary had a little lamb"  
String_1 += "lamb"  
  
print(String_1 == String_2, String_1 is String_2)
```

True False

```
class sampleclass :  
    def __init__(self, val) :  
        self.val = val  
  
object_1 = sampleclass(0)  
object_2 = sampleclass(2)  
object_3 = object_1  
  
object_3.val += 1  
  
print(object_1 is object_2)  
print(object_2 is object_3)  
print(object_3 is object_1)  
  
print(object_1.val, object_2.val, object_3.val)
```

False
False
True
1 2 1

Fungsi is operator digunakan untuk memeriksa apakah sebuah variable atau object pada program memiliki isi yang sama (isi tidak dimodifikasi).

- diamond problem [kerjakan tugas 1][modul]

```
class top :  
    def m_top(self) :  
        print('top')  
class middle_left(top) :  
    def m_middle(self) :  
        print('middle_left')  
class middle_right(top) :  
    def m_middle(self) :  
        print('middle right')  
class bottom(middle_left,middle_right) :  
    def m_bottom(self) :  
        print('bottom')  
  
obj = bottom()  
obj.m_bottom()  
obj.m_middle()  
obj.m_top()
```

```
bottom  
middle_left  
top
```

Program disamping merupakan program yang menjelaskan diamond problem. diamond problem merupakan masalah yang timbul apabila penempatan superclass dan subclass dilakukan secara terbalik yang akan menghasilkan error. untuk menghindari error tersebut superclass dan subclass harus didefinisikan secara urut seperti contoh disamping.

Tugas



```
class top :  
    def m_top(self) :  
        print('top')  
class middle_left(top) :  
    def m_middle(self) :  
        print('middle_left')  
class middle_right(top) :  
    def m_middle(self) :  
        print('middle right')  
class bottom(middle_right,middle_left) :  
    def m_bottom(self) :  
        print('bottom')  
  
obj = bottom()  
obj.m_bottom()  
obj.m_middle()  
obj.m_top()
```

✓ 0.1s

bottom
middle right
top

Modul



- enkapsulasi [kerjakan tugas 1 dan 2][modul]

```
class rekening :  
    def __init__(self) :  
        self.__saldo = 0  
  
    def kredit(self, jumlah) :  
        if jumlah < 0 :  
            print('Gagal kredit, jumlah kurang dari nol')  
            return  
  
            self.__saldo += jumlah  
  
    def debit(self, jumlah) :  
        if jumlah > self.__saldo :  
            print('Gagal debit, jumlah melebihi saldo')  
            return  
        if jumlah < 0 :  
            print('Gagal debit, jumlah tidak bisa kurang dari nol')  
            return  
  
            self.__saldo -= jumlah  
  
    def cetakSaldo(self) :  
        print('Saldo saat ini : ', self.__saldo)
```

```
def cetakSaldo(self) :  
    print('Saldo saat ini : ', self.__saldo)
```

```
bni_alim = rekening()  
bni_alim.kredit(1000000)  
bni_alim.debit(100000)  
bni_alim.cetakSaldo()
```

Saldo saat ini : 900000

Program disamping merupakan program yang digunakan untuk menghitung transaksi bank. menu yang ada dalam program tersebut adalah menu kredit, debit dan cetak saldo. kredit diberikan apabila saldo user tidak kurang dari nol. debit diberikan apabila jumlah penarikan tidak melebihi saldo dan tidak kurang dari nol. menu cetaksaldo menampilkan saldo yang dimiliki oleh user.

Modul



UNIVERSITAS GADJAH MADA

```
class siswa :  
    __mapel = []  
    __nilai = []  
  
    def __init__(self,nama,alamat) :  
        self.nama = nama  
        self.alamat = alamat  
  
    def getName(self) :  
        return self.nama  
  
    def tambahNilaiMapel(self,mapel,nilai) :  
        self.__mapel.append(mapel)  
        self.__nilai.append(nilai)  
  
    def ambilNilai(self) :  
        print(self.__nilai)  
        print(self.__mapel)  
  
data = siswa('Alim satria','semarang')  
print(data.nama)  
print(data.alamat)  
data.tambahNilaiMapel('Matematika',90)  
data.tambahNilaiMapel('Biologi',100)  
print(data.getName())  
data.ambilNilai()
```

```
Alim satria  
semarang  
Alim satria  
[90, 100]  
['Matematika', 'Biologi']
```

Program disamping merupakan program yang digunakan untuk mendata siswa. informasi dalam data adalah nama, alamat, mata pelajaran (dalam bentuk list), dan nilai (dalam bentuk list). menu yang ada pada program antara lain getName untuk mendapatkan nama siswa, tambahNilaiMapel untuk menambahkan nilai mapel dan nama mapel kedalam list, dan ambilNilai untuk mencetak nilai dari siswa.

Tugas



1. Variabel yang dienkapsulasi antara lain adalah saldo, mata pelajaran, dan nilai siswa
- 2.

```
class login :  
    def __init__(self, masukan) :  
        self.__password = "yakiniku"  
        self.masukan = masukan  
  
    def unlock(self) :  
        if self.masukan != self.__password :  
            print('password salah')  
            return  
        else :  
            print('password benar')  
            return  
  
pwd = str(input("masukkan pwd"))  
print("password : ", pwd)  
user = login(pwd)  
user.unlock()  
✓ 3.2s  
  
password : yakiniku  
password benar
```