

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Evodie Leanishar Harafi
221511011
TI21K

1. Buatlah masing-masing 2 contoh jenis pewarisan di luar dari contoh yang telah diberikan, beri nama: single1.py, single2.py, multiple1.py, multiple2.py, hierarchical1.py, hierarchical2.py, multilevel1.py, multilevel2, hybrid1.py, hybrid2.py

single1.py:

```
class Kuliner:
    def __init__(self, nama, daerah):
        self.nama = nama
        self.daerah = daerah

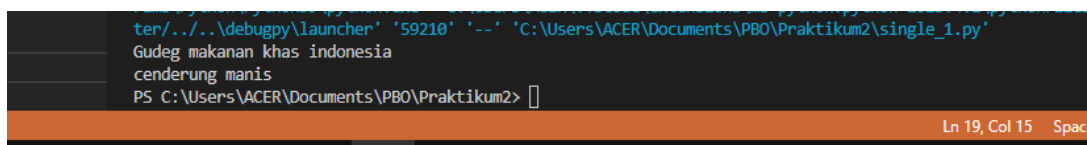
    def khas(self):
        print(self.nama, "makanan khas indonesia")

class Jogja(Kuliner):
    def __init__(self, nama, karakter, jenis_makanan):
        super().__init__(nama, karakter)
        self.jenis_makanan = jenis_makanan

    def rasa(self):
        print("cenderung manis")

jogjaA = Jogja("Gudeg", "Nasi" , "Berat")
jogjaA.khas()
jogjaA.rasa()
```

gambar 1.hasil single 1:



```
ter/../../debugpy\launcher' '59210' '--' 'C:\Users\ACER\Documents\PBO\Praktikum2\single_1.py'
Gudeg makanan khas indonesia
cenderung manis
PS C:\Users\ACER\Documents\PBO\Praktikum2> 
```

Ln 19, Col 15 Spac

single2.py:

```
class Motor:
    def __init__(self, nama, cc):
        self.nama = nama
        self.cc = cc

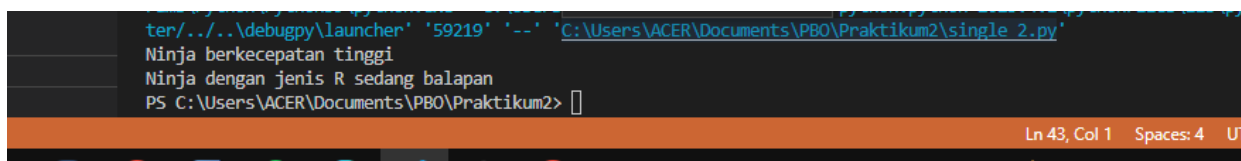
    def kecepatan(self):
        print(f"{self.nama} berkecepatan tinggi")

class kawasaki(Motor):
    def __init__(self, nama, cc, jenis):
        super().__init__(nama, cc)
        self.jenis = jenis

    def balapan(self):
        print(f"{self.nama} dengan jenis {self.jenis} sedang balapan")

kawasakiA = kawasaki("Ninja", 250, "R")
kawasakiA.kecepatan()
kawasakiA.balapan()
```

gambar 2.hasil single 2:



```
ter/../../debugpy\launcher' '59219' '--' 'C:\Users\ACER\Documents\PBO\Praktikum2\single 2.py'
Ninja berkecepatan tinggi
Ninja dengan jenis R sedang balapan
PS C:\Users\ACER\Documents\PBO\Praktikum2>
```

multiple1.py:

```
class motor:
    def __init__(self, nama, cc):
        self.nama = nama
        self.cc = cc

    def servis(self):
        print(self.nama, "sedang di servis")

class matic:
    def __init__(self, nama, transmisi):
        self.nama = nama
        self.transmisi = transmisi

    def modif(self):
        print(self.nama, "sedang di modif")

class MotorMatic(motor, matic):
    def __init__(self, nama, cc, transmisi):
        motor.__init__(self, nama, cc)
        matic.__init__(self, nama, transmisi)

    def uji(self):
        print(self.nama, "sedang di uji kelayakan")

motor_at = MotorMatic("mio", "150", "Automatic")
motor_at.servis()
motor_at.modif()
motor_at.uji()
```

gambar 3.hasil multiple 1:

```
PS C:\Users\ACER\Documents\PBO\Praktikum2> python multiple1.py
mio sedang di servis
mio sedang di modif
mio sedang di uji kelayakan
PS C:\Users\ACER\Documents\PBO\Praktikum2> █
```

multiple2.py:

```
class Mobil:
    def __init__(self, nama, jenis):
        self.nama = nama
        self.jenis = jenis
    def display_info(self):
        print(f>Nama: {self.nama}")
        print(f>Jenis: {self.jenis}")

class Manual:
    def __init__(self, jenis, transmisi):
        self.jenis = jenis
        self.transmisi = transmisi

    def display_info(self):
        print(f>Jenis: {self.jenis}")
        print(f>Transmisi: {self.transmisi}")

class Sport:
    def __init__(self, model, transmisi):
        self.model = model
        self.transmisi = transmisi

    def display_info(self):
        print(f>Model: {self.model}")
        print(f>Transmisi: {self.transmisi}")

class Lamborghini(Manual, Sport):
    def __init__(self, nama, jenis, transmisi, model, cc):
        self.cc= cc
        Mobil.__init__(self, nama, jenis)
        Manual.__init__(self, jenis, transmisi)
        Sport.__init__(self, model, transmisi)

    def display_info(self):
        super().display_info()
        print(f>Nama: {self.nama}")
        print(f>CC: {self.cc}")

LamborghiniA = Lamborghini("Aventador", "Sport", "Manual", "ZR", "5000")
LamborghiniA.display_info()
```

gambar 4.hasil multiple 2:

```
C:\Users\ACER\Documents\PR0\Praktikum2>
Jenis: Sport
Transmisi: Manual
Nama: Aventador
CC: 5000
PS: C:\Users\ACER\Documents\PR0\Praktikum2>
```

hierarchical1.py:

```
class pegawai:
    def __init__(self, nama, umur, gaji):
        self.nama = nama
        self.umur = umur
        self.gaji = gaji

    def display_info(self):
        print("Nama:", self.nama)
        print("Umur:", self.umur)
        print("Gaji:", self.gaji)

class Manager(pegawai):
    def __init__(self, name, umur, gaji, department):
        super().__init__(name, umur, gaji)
        self.department = department

    def display_info(self):
        super().display_info()
        print("Department:", self.department)

class SalesManager(Manager):
    def __init__(self, nama, umur, gaji, department, bonus):
        super().__init__(nama, umur, gaji, department)
        self.bonus = bonus

    def display_info(self):
        super().display_info()
        print("Bonus:", self.bonus)

class HRManager(Manager):
    def __init__(self, nama, umur, gaji, department, tunjangan):
        super().__init__(nama, umur, gaji, department)
        self.tunjangan = tunjangan

    def display_info(self):
        super().display_info()
        print("Tunjangan:", self.tunjangan)

pegawaiA = pegawai("wahyu", 20, 6000)
managerA = Manager("aji harka", 21, 50000, "Kyai")
sma = SalesManager("nopal", 25, 15000, "Pengerus", 5000)
hrA = HRManager("upi", 40, 20000, "santri", 10000)

pegawaiA.display_info()
print("-----")
managerA.display_info()
print("-----")
```

```
smA.display_info()  
print("-----")  
hrA.display_info()
```

gambar5 hasil **hierarchical1.py**:

```
Umur: 25  
Gaji: 15000  
Department: Pengerus  
Bonus: 5000  
-----  
Nama: upi  
Umur: 40  
Gaji: 20000  
Department: santri  
Tunjangan: 10000  
PS C:\Users\ACER\Documents\PBO\Praktikum2>
```


hierarchical2.py:

```
class Hewan:
    def __init__(self, nama, species, suara):
        self.nama = nama
        self.species = species
        self.suara = suara

    def bersuara(self):
        print(self.suara)

class Mammalia(Hewan):
    def __init__(self, nama, species, suara, kaki):
        super().__init__(nama, species, suara)
        self.kaki = kaki

    def berjalan(self):
        print(f"{self.nama} berjalan dengan {self.kaki} kaki.")

class paus(Mammalia):
    def __init__(self, nama, ras, ekor):
        super().__init__(nama, "paus", "orka", ekor)
        self.ras = ras

    def mengaung(self):
        print(f"{self.nama} bersuara oummmmmm.")

class sapi(Mammalia):
    def __init__(self, nama, ras, kaki):
        super().__init__(nama, "sapi", "brahman", kaki)
        self.ras = ras

    def menyeruduk(self):
        print(f"{self.nama} menyeruduk orang.")

pausA = paus("biru", "hiu paus", 4)
sapiA = sapi("mouu", "simental", 4)

pausA.bersuara()
pausA.mengaung()
pausA.berjalan()
print("-----")

sapiA.bersuara()
sapiA.berjalan()
sapiA.menyeruduk()
```

gambar6 hasil hierarchical2.py:

```
orka
biru bersuara oummmmmm.
-----
```

multilevel1.py:

```
class kendaraan:
    def __init__(self, brand, model, tahun):
        self.brand = brand
        self.model = model
        self.tahun = tahun

    def drive(self):
        pass

class mobil(kendaraan):
    def __init__(self, brand, model, tahun, jml_pintu):
        super().__init__(brand, model, tahun)
        self.jml_pintu = jml_pintu

    def drive(self):
        return "Mengendarai Mobil " + self.brand + " " + self.model

class motor(kendaraan):
    def __init__(self, brand, model, tahun, cc):
        super().__init__(brand, model, tahun)
        self.cc = cc

    def drive(self):
        return "Mengendarai Motor " + self.brand + " " + self.model

class mobil_listrik(mobil):
    def __init__(self, brand, model, tahun, jml_pintu, kapasitas_baterai):
        super().__init__(brand, model, tahun, jml_pintu)
        self.kapasitas_baterai = kapasitas_baterai

    def charge(self):
        return "Mengisi Daya Mobil " + self.brand + " " + self.model + " dengan  

daya" + str(self.kapasitas_baterai) + " kWh"

mbl_saya = mobil("Kijang", "Inova", 2022, 5)
mtr_saya = motor("Supra", "X", 2020, "125 cc")
ev_saya = mobil_listrik("Wuling", "Air EV", 2023, 4, 100)

print(mbl_saya.drive())
print(mtr_saya.drive())
print(ev_saya.drive())
print(ev_saya.charge())
```

gambar7 hasil multilevel1.py:

```
Mengendarai Mobil Kijang Inova  
Mengendarai Motor Supra X  
Mengendarai Mobil Wuling Air EV  
Mengisi Daya Mobil Wuling Air EV dengan daya100 kwh  
PS C:\Users\ACER\Documents\PBO\Praktikum2>
```

multilevel2.py:

```
class hewan:
    def __init__(self, nama):
        self.nama = nama

    def berbunyi(self):
        pass

class mamalia(hewan):
    def __init__(self, nama):
        super().__init__(nama)

    def melahirkan(self):
        pass

class anjing(mamalia):
    def __init__(self, nama):
        super().__init__(nama)

    def berbunyi(self):
        return "guk guk"

class kucing(mamalia):
    def __init__(self, nama):
        super().__init__(nama)

    def berbunyi(self):
        return "meong"

class burung(hewan):
    def __init__(self, nama):
        super().__init__(nama)

    def fly(self):
        pass

class beo(burung):
    def __init__(self, nama):
        super().__init__(nama)

    def berbunyi(self):
        return "assalamualaikum"

anjingA = anjing("bleki")
kucingA = kucing("oyen")
beoA = beo("bribin")

print(anjingA.nama + ": " + anjingA.berbunyi())
print(kucingA.nama + ": " + kucingA.berbunyi())
print(beoA.nama + ": " + beoA.berbunyi())
```

gambar8 hasil multilevel2.py:

bleki: guk guk

oyen: meong

bribin: assalamualaikum

PS C:\Users\ACER\Documents\PBO\Praktikum2> |

hybrid1.py:

```
class kendaraan:
    def __init__(self, nyala, model, tahun, berat):
        self.nyala = nyala
        self.model = model
        self.tahun = tahun
        self.berat = berat

    def nyalakan_mesin(self):
        print(f"Mesin {self.nyala} {self.model} di nyalakan.")

    def matikan_mesin(self):
        print(f"Mesin {self.nyala} {self.model} di matikan")

class mobil(kendaraan):
    def __init__(self, nyala, model, tahun, berat, jml_pintu):
        super().__init__(nyala, model, tahun, berat)
        self.jml_pintu = jml_pintu

    def memiliki_pintu(self):
        print(f"Mobil {self.nyala} {self.model} Memiliki {self.jml_pintu}
pintu")

class motor(kendaraan):
    def __init__(self, nyala, model, tahun, berat, jml_roda):
        super().__init__(nyala, model, tahun, berat)
        self.jml_roda = jml_roda

    def berjalan(self):
        print(f"Motor {self.nyala} {self.model} berjalan dengan {self.jml_roda}
roda.")

class mbl_listrik(mobil):
    def __init__(self, nyala, model, tahun, berat, jml_pintu,
kapasitas_baterai):
        super().__init__(nyala, model, tahun, berat, jml_pintu)
        self.kapasitas_baterai = kapasitas_baterai

    def charge_baterai(self):
        print(f"Mobil {self.nyala} {self.model} memiliki kapasitas baterai
{self.kapasitas_baterai} kWh.")

class Electricmotor(motor):
    def __init__(self, nyala, model, tahun, berat, jml_roda, kapasitas_baterai):
        super().__init__(nyala, model, tahun, berat, jml_roda)
        self.kapasitas_baterai = kapasitas_baterai

    def charge_baterai(self):
        print(f"Motor {self.nyala} {self.model} memiliki kapasitas baterai
{self.kapasitas_baterai} kWh.")
```

```

mobilA = mobil("Toyota", "Inova", 2022, 1200, 4)
mobilA.nyalakan_mesin()
mobilA.memiliki_pintu()
mobilA.matikan_mesin()

motorA = motor("Honda", "Supra X", 2021, 250, 2)
motorA.nyalakan_mesin()
motorA.berjalan()
motorA.matikan_mesin()

ev = mbl_listrik("Wuling", "Air Ev", 2023, 2000, 4, 100)
ev.nyalakan_mesin()
ev.charge_baterai()
ev.charge_baterai()
ev.matikan_mesin()

e_bike = Electricmotor("flywin", "Ø1", 2023, 300, 2, 15.5)
e_bike.nyalakan_mesin()
e_bike.charge_baterai()
e_bike.charge_baterai()
e_bike.matikan_mesin()

```

gambar9 hasil hybrid1.py:

```

Motor Honda Supra X berjalan dengan 2 roda.
Mesin Honda Supra X di matikan
Mesin Wuling Air Ev di nyalakan.
Mobil Wuling Air Ev memiliki kapasitas baterai 100 kwh.
Mobil Wuling Air Ev memiliki kapasitas baterai 100 kwh.
Mesin Wuling Air Ev di matikan
Mesin flywin Ø1 di nyalakan.
Motor flywin Ø1 memiliki kapasitas baterai 15.5 kwh.
Motor flywin Ø1 memiliki kapasitas baterai 15.5 kwh.
Mesin flywin Ø1 di matikan
PS C:\Users\ACER\Documents\PBO\Praktikum2>

```

hybrid2.py :

```
class hewan:
    def __init__(self, nama, species, umur):
        self.nama = nama
        self.species = species
        self.umur = umur

    def gerak(self):
        print(f" {self.species} bernama {self.nama} sedang bergerak.")

    def makan(self):
        print(f" {self.species} bernama {self.nama} sedang makan.")

class mamalia(hewan):
    def __init__(self, nama, species, umur, jml_kaki):
        super().__init__(nama, species, umur)
        self.jml_kaki = jml_kaki

    def melahirkan(self):
        print(f" {self.species} bernama {self.nama} telah melahirkan.")

class burung(hewan):
    def __init__(self, nama, species, umur, warna_bulu):
        super().__init__(nama, species, umur)
        self.warna_bulu = warna_bulu

    def warna(self):
        print(f" {self.species} bernama {self.nama} memiliki warna bulu {self.warna_bulu} ")

class reptil(hewan):
    def __init__(self, nama, species, umur, habitat):
        super().__init__(nama, species, umur)
        self.habitat = habitat

    def hidup(self):
        print(f" {self.species} bernama {self.nama} banyak hidup di {self.habitat} ")

class anjing(mamalia):
    def __init__(self, nama, umur, ras, jml_kaki):
        super().__init__(nama, "anjing", umur, jml_kaki)
        self.ras = ras

    def menggonggong(self):
        print(f" {self.nama} bernama {self.ras} suka menggonggong")

class beo(burung):
    def __init__(self, nama, umur, warna_bulu, bicara):
        super().__init__(nama, "beo", umur, warna_bulu)
        self.bicara = bicara
```



```

def ngomong(self):
    if self.bicara:
        print(f"beo bernama {self.nama} suka ngomong")
    else:
        print(f"beo bernama {self.nama} ga bisa ngomong")

class ular(reptil):
    def __init__(self, nama, umur, habitat, berbisa):
        super().__init__(nama, "ular", umur, habitat)
        self.berbisa = berbisa

    def menggigit(self):
        if self.berbisa:
            print(f"ular bernama {self.nama} memiliki bisa")
        else:
            print(f"ular bernama {self.nama} tidak berbisa")

anjing = anjing("guguk", 3, "kintamani", 4)
anjing.gerak()
anjing.makan()
anjing.melahirkan()
anjing.menggonggong()

beo = beo("ngomul", 3, "merah", True)
beo.gerak()
beo.makan()
beo.warna()
beo.ngomong()

ular = ular("mamba", 3, "semak semak", True)
ular.gerak()
ular.makan()
ular.hidup()
ular.menggigit()

```

gambar10 hasil hybrid2.py:

```

    anjing bernama guguk telah melahirkan.
    guguk bernama kintamani suka menggonggong
    beo bernama ngomul sedang bergerak.
    beo bernama ngomul sedang makan.
    beo bernama ngomul memiliki warna bulu merah
    beo bernama ngomul suka ngomong
    ular bernama mamba sedang bergerak.
    ular bernama mamba sedang makan.
    ular bernama mamba banyak hidup di semak semak
    ular bernama mamba memiliki bisa
PS C:\Users\ACER\Documents\PBO\Praktikum2>

```