

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

NAMA	: TITO FEBRI ANGGARA
NIM	: 210511126
KELAS	: R3(C)

Single1

class Makanan:

```
def __init__(self, nama, rasa):
    self.nama = nama
    self.rasa = rasa
def memiliki_rasa(self):
    print(f"{self.nama}, memiliki rasa {self.rasa} rasa pedas")
```

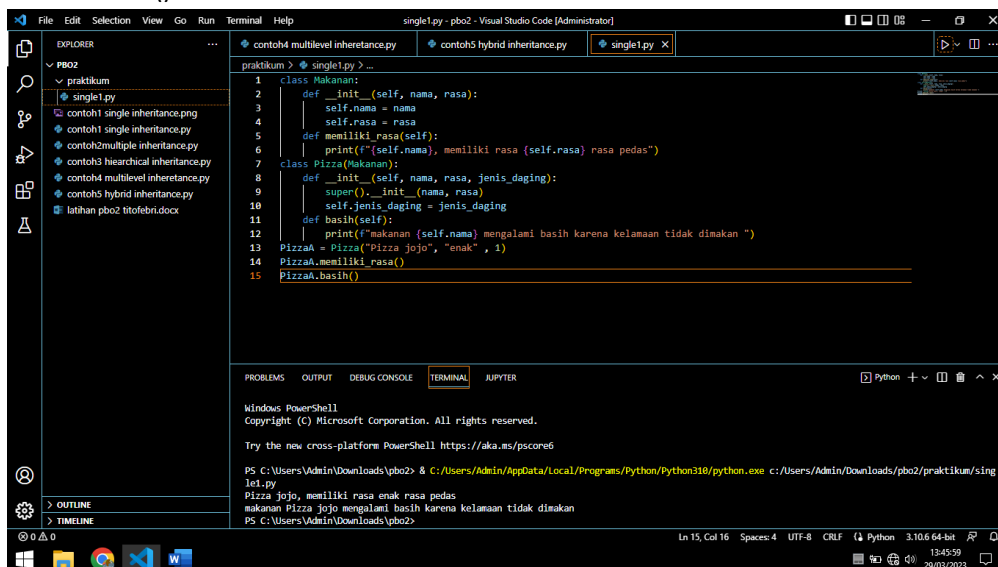
class Pizza(Makanan):

```
def __init__(self, nama, rasa, jenis_daging):
    super().__init__(nama, rasa)
    self.jenis_daging = jenis_daging
def basih(self):
    print(f"makanan {self.nama} mengalami basih karena kelamaan tidak dimakan ")
```

PizzaA = Pizza("Pizza jojo", "enak" , 1)

PizzaA.memiliki_rasa()

PizzaA.basih()



The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'pbo2' with subfolders 'praktikum' and 'latihan'. The 'praktikum' folder is expanded, showing 'single1.py' selected. The main editor displays the Python code for 'Single1'. The terminal at the bottom shows the execution output: 'Pizza jojo, memiliki rasa enak pedas' and 'makanan Pizza jojo mengalami basih karena kelamaan tidak dimakan'.

```
1 class Makanan:
2     def __init__(self, nama, rasa):
3         self.nama = nama
4         self.rasa = rasa
5     def memiliki_rasa(self):
6         print(f"{self.nama}, memiliki rasa {self.rasa} rasa pedas")
7 class Pizza(Makanan):
8     def __init__(self, nama, rasa, jenis_daging):
9         super().__init__(nama, rasa)
10        self.jenis_daging = jenis_daging
11    def basih(self):
12        print(f"makanan {self.nama} mengalami basih karena kelamaan tidak dimakan ")
13 PizzaA = Pizza("Pizza jojo", "enak" , 1)
14 PizzaA.memiliki_rasa()
15 PizzaA.basih()
```

Terminal Output:

```
PS C:\Users\Admin\Downloads\pbo2> python single1.py
Pizza jojo, memiliki rasa enak pedas
makanan Pizza jojo mengalami basih karena kelamaan tidak dimakan
PS C:\Users\Admin\Downloads\pbo2>
```

Single2

class BUNGA:

```
def __init__(self, nama, warna):
    self.nama = nama
    self.warna = warna
def mekar(self):
    print(f"{self.nama}, berwarna {self.warna} bunga mekar ")
```

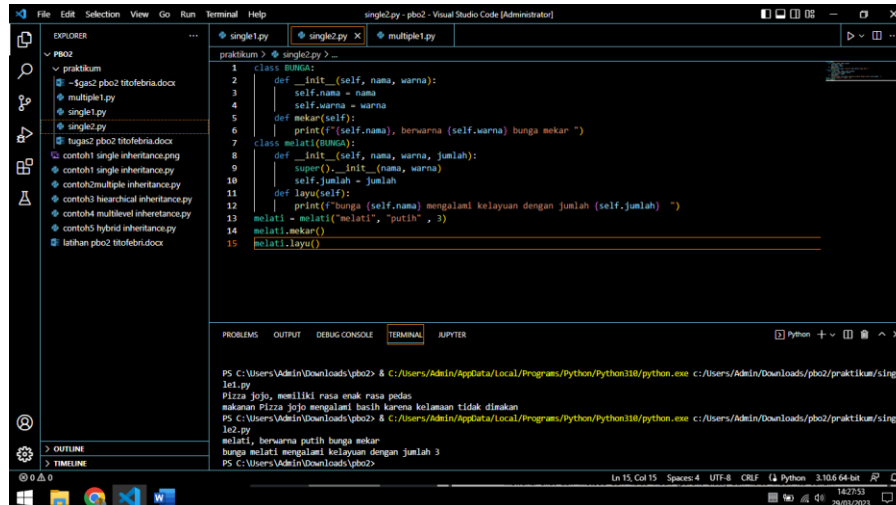
class melati(BUNGA):

```
def __init__(self, nama, warna, jumlah):
    super().__init__(nama, warna)
    self.jumlah = jumlah
def layu(self):
```

```

    print(f"bunga {self.nama} mengalami kelayuan dengan jumlah {self.jumlah} ")
melati = melati("melati", "putih" , 3)
melati.mekar()
melati.layu()

```



```

1 class BUNGA:
2     def __init__(self, nama, warna):
3         self.nama = nama
4         self.warna = warna
5     def mekar(self):
6         print(f"{self.nama}, berwarna {self.warna} bunga mekar ")
7     class melati(melati):
8         def __init__(self, nama, warna, jumlah):
9             super().__init__(nama, warna)
10            self.jumlah = jumlah
11            def layu(self):
12                print(f"bunga {self.nama} mengalami kelayuan dengan jumlah {self.jumlah} ")
13 melati = melati("melati", "putih" , 3)
14 melati.mekar()
15 melati.layu()

```

MULTIPLE1

class siswa:

```

def __init__(self, nama, nis):
    self.nama = nama
    self.nis = nis
def belajar(self):
    print(self.nama, "sedang selesaikan tugas")

```

class Pekerja:

```

def __init__(self, nama, pekerjaan):
    self.nama = nama
    self.pekerjaan = pekerjaan
def bekerja(self):
    print(self.nama, "sedang mengetik ")

```

class siswaPekerja(siswa, Pekerja):

```

def __init__(self, nama, nis, pekerjaan):
    siswa.__init__(self, nama, nis)
    Pekerja.__init__(self, nama, pekerjaan)
def mengobrol(self):
    print(self.nama, "sedang mengobrol bersama teman")

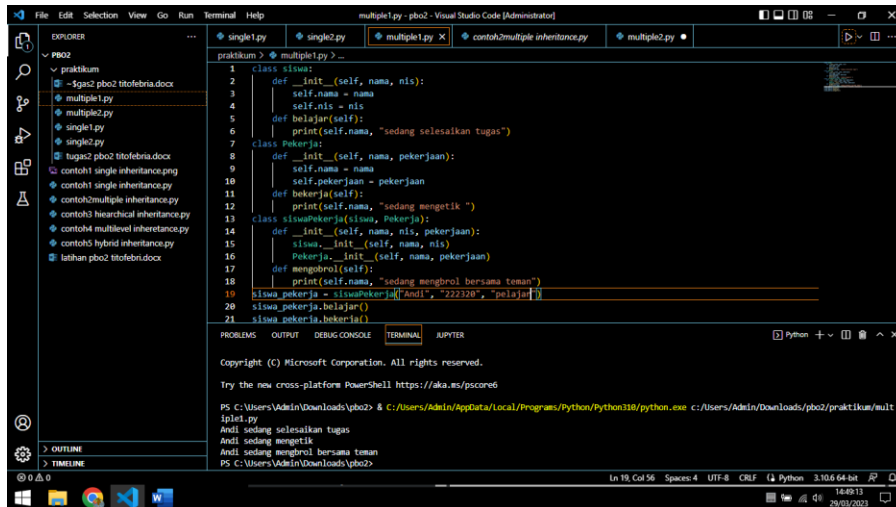
```

siswa_pekerja = siswaPekerja("Andi", "222320", "pelajar")

siswa_pekerja.belajar()

siswa_pekerja.bekerja()

siswa_pekerja.mengobrol()



MULTIPLE2

class Hewan:

```
def __init__(self, nama, umur):
    self.nama = nama
    self.umur = umur
def display_info(self):
    print(f>Nama: {self.nama}")
    print(f"Umur: {self.umur}")
```

class Reptil:

```
def __init__(self, jenis, habitat):
    self.jenis = jenis
    self.habitat = habitat
def display_info(self):
    print(f"Jenis: {self.jenis}")
    print(f"Habitat: {self.habitat}")
```

class Amphibi:

```
def __init__(self, metamorfosis, habitat):
    self.metamorfosis = metamorfosis
    self.habitat = habitat
def display_info(self):
    print(f"Metamorfosis: {self.metamorfosis}")
    print(f"Habitat: {self.habitat}")
```

class Katak(Reptil, Amphibi):

```
def __init__(self, nama, umur, jenis, habitat, metamorfosis):
    Hewan.__init__(self, nama, umur)
    Reptil.__init__(self, jenis, habitat)
    Amphibi.__init__(self, metamorfosis, habitat)
def display_info(self):
    super().display_info()
```

```
katakA = Katak("Katak pohon", "1", "Amphibi", "Darat", "Telur")
katakA.display_info()
```



```
def __init__(self, nama):
    self.nama = nama
def suara(self):
    print(f"{self.nama} suara")
```

```
def __init__(self, nama, panjang):
    super().__init__(nama)
    self.panjang = panjang
def merayap(self):
    print(f"seekor ular{self.nama} sepanjang{self.panjang} merayap di pohon")
```

```
def __init__(self, nama, panjang, warna):
    super().__init__(nama, panjang)
    self.warna = warna
def ganti_kulit(self):
    print(f"seekor ular {self.nama} yang berwarna {self.warna} sedang ganti kulit")
```

pucuk.merayap()



Multilevel2

class Hewan:

```
def __init__(self, nama):
```

```
    self.nama = nama
```

```
def suara(self):
```

```
    print(f"{self.nama} suara")
```

class harimau(Hewan):

```
def __init__(self, nama, berat):
```

```
    super().__init__(nama)
```

```
    self.berat = berat
```

```
def berjalan(self):
```

```
    print(f"seekor harimau {self.nama} seberat {self.berat} berjalan mencari mangsa di hutan")
```

class sumatra(harimau):

```
def __init__(self, nama, berat, warna):
```

```
    super().__init__(nama, berat)
```

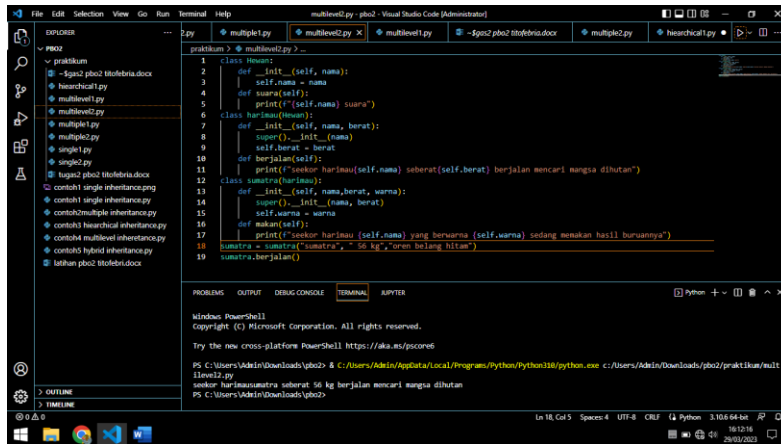
```
    self.warna = warna
```

```
def makan(self):
```

```
    print(f"seekor harimau {self.nama} yang berwarna {self.warna} sedang memakan hasil buruannya")
```

```
sumatra = sumatra("sumatra", " 56 kg", "oren belang hitam")
```

```
sumatra.berjalan()
```



HYBRID1

```
print('\nHybrid Inheritance_Kampus\n\n')
```

```
class Univ:
```

```
    def __init__(self):
        self.univ = 'UMC'
```

```
    def display(self):
        print('Universitas\t: ',self.univ)
```

```
class Fakultas(Univ):
```

```
    def __init__(self):
        Univ.__init__(self)
        self.fakultas = 'Teknik'
```

```
    def display(self):
        Univ.display(self)
        print('Fakultas\t: ', self.fakultas)
```

```
class Prodi(Univ):
```

```
    def __init__(self):
        Univ.__init__(self)
        self.prodi = 'Teknik Informatika\n'
```

```
    def display(self):
        print('Program Studi\t: ', self.prodi)
```

```
class Mhs(Fakultas, Prodi):
```

```
    def __init__(self):
        self.name = 'tito febri a'
```

```
Fakultas.__init__(self)
```

```
Prodi.__init__(self)
```

```
def display(self):  
    print('Nama\t\t: ', self.name)  
    Fakultas.display(self)  
    Prodi.display(self)
```

```
mhs1 = Mhs()
```

```
mhs1.display()
```



HYBRID2

```
print('\nHybrid Inheritance_IDOL\n\n')
```

Hybrid Inheritance

class vehicle:

```
def __init__(self,name,grup):  
    self.name = name  
    self.grup = grup
```

```
def show_details(self):  
    print(f'\nName : {self.name}')  
    print(f'Grup : {self.grup}')
```

class bike(vehicle):

Inherit Properties and Override

```
def __init__(self,name,grup,album):  
    super().__init__(name,grup)  
    self.album = album
```



```
# Inherit Behavior and Override
def show_details(self):
    super().show_details()
    print(f'Album : {self.album}\n')
```

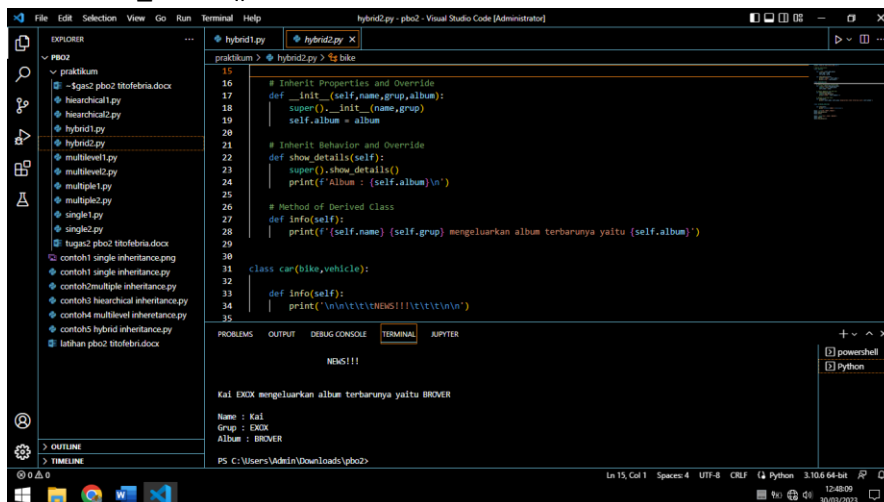
```
# Method of Derived Class
def info(self):
    print(f'{self.name} {self.grup} mengeluarkan album terbarunya yaitu {self.album}')
```

```
class car(bike,vehicle):
```

```
def info(self):
    print("\n\n\t\t\tNEWS!!!\t\t\t\n\n')
```

```
bajaj = car("SO", "KPOP", "BROSE")
bajaj.show_details()
bajaj.info()
```

```
idol1 = bike("Kai", "EXOX", "BROVER")
idol1.info()
idol1.show_details()
```



HIEARCHICAL1

```
print('\nHierarchical Inheritance_KPOP\n\n')
```

```
class Grup:
    def __init__(self, grup, anggota):
```

```

        self.grup = grup
        self.anggota = anggota

    def ket(self):
        print(f'{self.grup} beranggotakan {self.anggota} orang\n\n')

    def getGrup(self):
        return self.grup

    def getAnggota(self):
        return self.anggota

class Gen(Grup):
    def __init__(self, grup, anggota, gen):
        super().__init__(grup, anggota)
        self.gen = gen

    def detail(self):
        print(f'Grup {self.grup} merupakan Generasi Ke-{self.gen} Kpop\n')

    def getGen(self):
        return self.gen

class Agensi(Gen):
    def __init__(self, grup, anggota, gen, fandom, agensi):
        super().__init__(grup, anggota, gen)
        self.fandom = fandom
        self.ageni = agensi

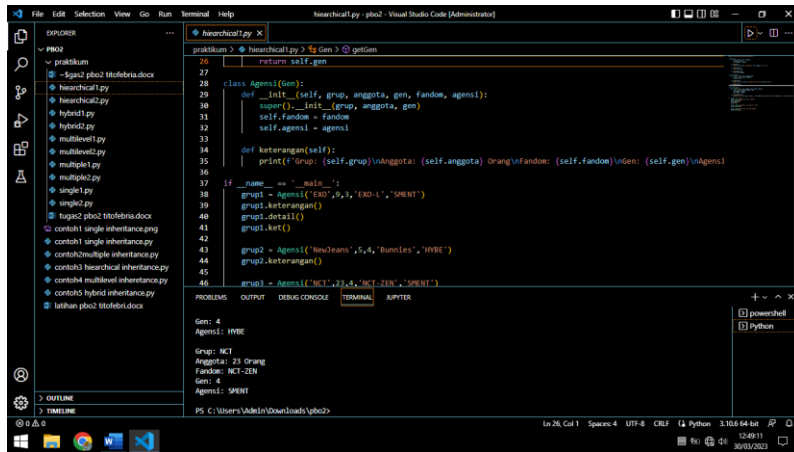
    def keterangan(self):
        print(f'Grup: {self.grup}\nAnggota: {self.anggota} Orang\nFandom: {self.fandom}\nGen: {self.gen}\nAgeni: {self.ageni}\n')

if __name__ == '__main__':
    grup1 = Agensi('EXO',9,3,'EXO-L','SMENT')
    grup1.keterangan()
    grup1.detail()
    grup1.ket()

    grup2 = Agensi('NewJeans',5,4,'Bunnies','HYBE')
    grup2.keterangan()

```

```
grup3 = Agensi('NCT',23,4,'NCT-ZEN','SMENT')
grup3.keterangan()
```



HIEARCHICAL2

```
print('\nHierarchical Inheritance_Mahasiswa\n\n')
```

class Mahasiswa:

```

    def __init__(self, name, nim):
        self.name = name
        self.nim = nim

```

```

    def ket(self):
        print(f'{self.name} adalah Mahasiswa UMC dengan NIM {self.nim}\n')

```

```

    def getName(self):
        return self.name

```

```

    def getNim(self):
        return self.nim

```

class Fakultas(Mahasiswa):

```

    def __init__(self, name, nim, fakultas):
        super().__init__(name, nim)
        self.fakultas = fakultas

```

```

    def detail(self):
        print(f>Nama: {self.name}\nNim: {self.nim}\nFakultas: {self.fakultas}\n')

```

```

    def getFakultas(self):
        return self.fakultas

```

class Prodi(Fakultas):

def __init__(self, name, nim, fakultas, prodi):

super().__init__(name, nim, fakultas)

self.prodi = prodi

def keterangan(self):

print(f'Nama: {self.name}\nNim: {self.nim}\nFakultas: {self.fakultas}\nProdi: {self.prodi}\n')

if __name__ == '__main__':

mhs1 = Prodi('TITO FEBRI', 210511126, 'Teknik', 'Teknik Informatika')

mhs1.keterangan()

mhs1.detail()

mhs1.ket()



The screenshot shows a Visual Studio Code window with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Python script with the following content:

```
praktikum > hierarchical2.py > ...
20 | self.fakultas = fakultas
21 |
22 | def detail(self):
23 | | print(f'Nama: {self.name}\nNim: {self.nim}\nFakultas: {self.fakultas}\n')
24 |
25 | def getFakultas(self):
26 | | return self.fakultas
27 |
28 | class Prodi(Fakultas):
29 | | def __init__(self, name, nim, fakultas, prodi):
30 | | | super().__init__(name, nim, fakultas)
31 | | | self.prodi = prodi
32 | |
33 | | def keterangan(self):
34 | | | print(f'Nama: {self.name}\nNim: {self.nim}\nFakultas: {self.fakultas}\nProdi: {self.prodi}\n')
35 | |
36 | if __name__ == '__main__':
37 | | mhs1 = Prodi('TITO FEBRI', 210511126, 'Teknik', 'Teknik Informatika')
38 | | mhs1.keterangan()
39 | | mhs1.detail()
40 | | mhs1.ket()
```

The terminal at the bottom shows the output of the script:

```
Fakultas: Teknik
Prodi: Teknik Informatika

Nama: TITO FEBRI
Nim: 210511126
Fakultas: Teknik

TITO FEBRI adalah Mahasiswa UPC dengan NIM 210511126
```