

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Nama: Muhammad Alif Taufiqurahman

Nim: 210511127

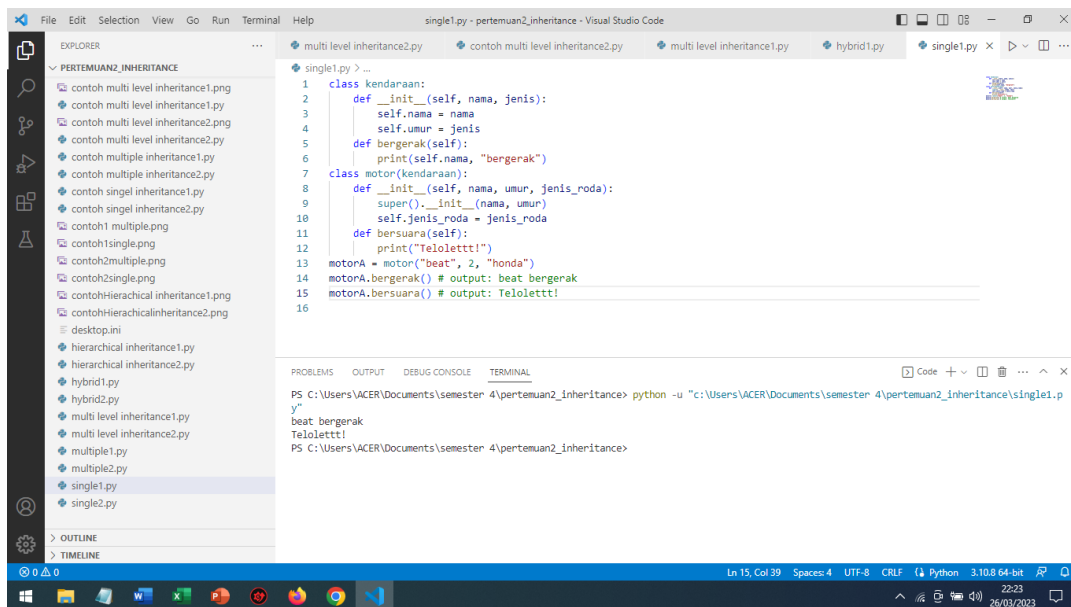
Kelas: R3/C Teknik Informatika



Edit dengan WPS Office

1. Singel1.py

2. `class kendaraan:`
3. `def __init__(self, nama, jenis):`
4. `self.nama = nama`
5. `self.umur = jenis`
6. `def bergerak(self):`
7. `print(self.nama, "bergerak")`
8. `class motor(kendaraan):`
9. `def __init__(self, nama, umur, jenis_roda):`
10. `super().__init__(nama, umur)`
11. `self.jenis_roda = jenis_roda`
12. `def bersuara(self):`
13. `print("Telolettt!")`
14. `motorA = motor("beat", 2, "honda")`
15. `motorA.bergerak() # output: beat bergerak`
16. `motorA.bersuara() # output: Telolettt!`
- 17.



```
1 class kendaraan:
2     def __init__(self, nama, jenis):
3         self.nama = nama
4         self.umur = jenis
5     def bergerak(self):
6         print(self.nama, "bergerak")
7 class motor(kendaraan):
8     def __init__(self, nama, umur, jenis_roda):
9         super().__init__(nama, umur)
10        self.jenis_roda = jenis_roda
11    def bersuara(self):
12        print("Telolettt!")
13 motorA = motor("beat", 2, "honda")
14 motorA.bergerak() # output: beat bergerak
15 motorA.bersuara() # output: Telolettt!
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\ACER\Documents\semester 4\pertemuan2_inheritance> python -u "c:\Users\ACER\Documents\semester 4\pertemuan2_inheritance\single1.py"

beat bergerak

Telolettt!

PS C:\Users\ACER\Documents\semester 4\pertemuan2_inheritance>



Edit dengan WPS Office

Singel2.py

class Buah:

```
def __init__(self, nama, warna):
    self.nama = nama
    self.warna = warna
def berbicara(self):
    print(f"{self.nama} ini sangat manis.")
```

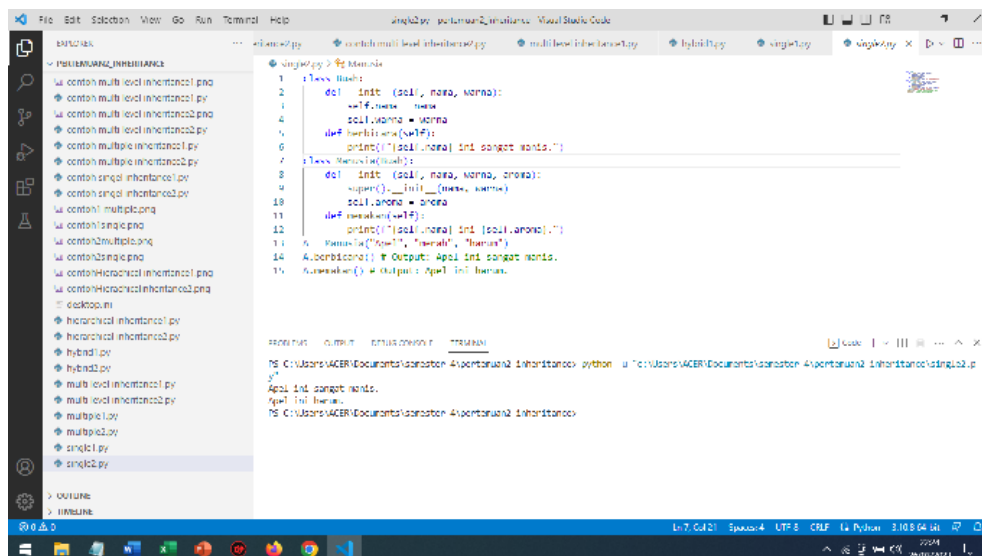
class Manusia(Buah):

```
def __init__(self, nama, warna, aroma):
    super().__init__(nama, warna)
    self.aroma = aroma
def memakan(self):
    print(f"{self.nama} ini {self.aroma}.")
```

A = Manusia("Apel", "merah", "harum")

A.berbicara() # Output: Apel ini sangat manis.

A.memakan() # Output: Apel ini harum.



The screenshot shows a code editor with a file explorer on the left and a terminal at the bottom. The file explorer lists various Python files, including 'singel2.py'. The code editor displays the following Python code:

```
1 class Buah:
2     def __init__(self, nama, warna):
3         self.nama = nama
4         self.warna = warna
5     def berbicara(self):
6         print(f"{self.nama} ini sangat manis.")
7
8 class Manusia(Buah):
9     def __init__(self, nama, warna, aroma):
10         super().__init__(nama, warna)
11         self.aroma = aroma
12     def memakan(self):
13         print(f"{self.nama} ini {self.aroma}.")
14
15 A = Manusia("Apel", "merah", "harum")
16 A.berbicara()
17 A.memakan()
```

The terminal output shows the execution of the code:

```
Python 3.10.5 Shell
> C:\Users\ACER\Documents\Aspirasi2\Inheritance\python> python u:\C:\Users\ACER\Documents\Aspirasi2\Inheritance\singel2.py
Apel ini sangat manis.
Apel ini harum.
```



Edit dengan WPS Office

2. multiple1.py

class Manusia:

```
def __init__(self, nama, jk):
    self.nama = nama
    self.jk = jk
def makan(self):
    print(self.nama, "sedang makan")
```

class Tidur:

```
def __init__(self, nama, kegiatan):
    self.nama = nama
    self.kegiatan = kegiatan
def bekerja(self):
    print(self.nama, "sedang tidur")
```

class MahasiswaPekerja(Manusia, Tidur):

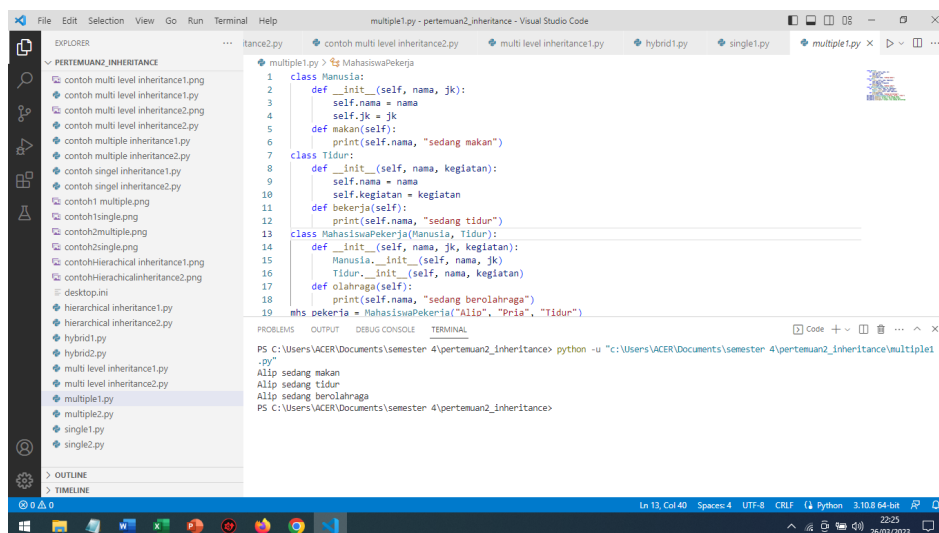
```
def __init__(self, nama, jk, kegiatan):
    Manusia.__init__(self, nama, jk)
    Tidur.__init__(self, nama, kegiatan)
def olahraga(self):
    print(self.nama, "sedang berolahraga")
```

mhs_pekerja = MahasiswaPekerja("Alip", "Pria", "Tidur")

mhs_pekerja.makan() # output: Alip sedang makan

mhs_pekerja.bekerja() # output: Alip sedang tidur

mhs_pekerja.olahraga() # output: Alip sedang berolahraga



```
1 class Manusia:
2     def __init__(self, nama, jk):
3         self.nama = nama
4         self.jk = jk
5     def makan(self):
6         print(self.nama, "sedang makan")
7
8 class Tidur:
9     def __init__(self, nama, kegiatan):
10         self.nama = nama
11         self.kegiatan = kegiatan
12     def bekerja(self):
13         print(self.nama, "sedang tidur")
14
15 class MahasiswaPekerja(Manusia, Tidur):
16     def __init__(self, nama, jk, kegiatan):
17         Manusia.__init__(self, nama, jk)
18         Tidur.__init__(self, nama, kegiatan)
19     def olahraga(self):
20         print(self.nama, "sedang berolahraga")
21
22 mhs_pekerja = MahasiswaPekerja("Alip", "Pria", "Tidur")
23
24 mhs_pekerja.makan()
25 mhs_pekerja.bekerja()
26 mhs_pekerja.olahraga()
```

PS C:\Users\ACER\Documents\semester 4\pertemuan2_inheritance> python -u "c:\Users\ACER\Documents\semester 4\pertemuan2_inheritance\multiple1.py"

Alip sedang makan
Alip sedang tidur
Alip sedang berolahraga

PS C:\Users\ACER\Documents\semester 4\pertemuan2_inheritance>



Edit dengan WPS Office

Multiple2.py

```
class Barang:
    def __init__(self, nama, kode, merek, ukuran):
        self.nama = nama
        self.kode_barang = kode
        self.merek = merek
        self.ukuran = ukuran
    def display_info(self):
        print(f>Nama barang: {self.nama}\nKode: {self.kode_barang} \nMerek: {self.merek} \nUkuran:
{self.ukuran}")
class Stok:
    def __init__(self, jumlah_barang):
        self.jumlah = jumlah_barang
    def display_info(self):
        print(f>Jumlah barang: {self.jumlah}")

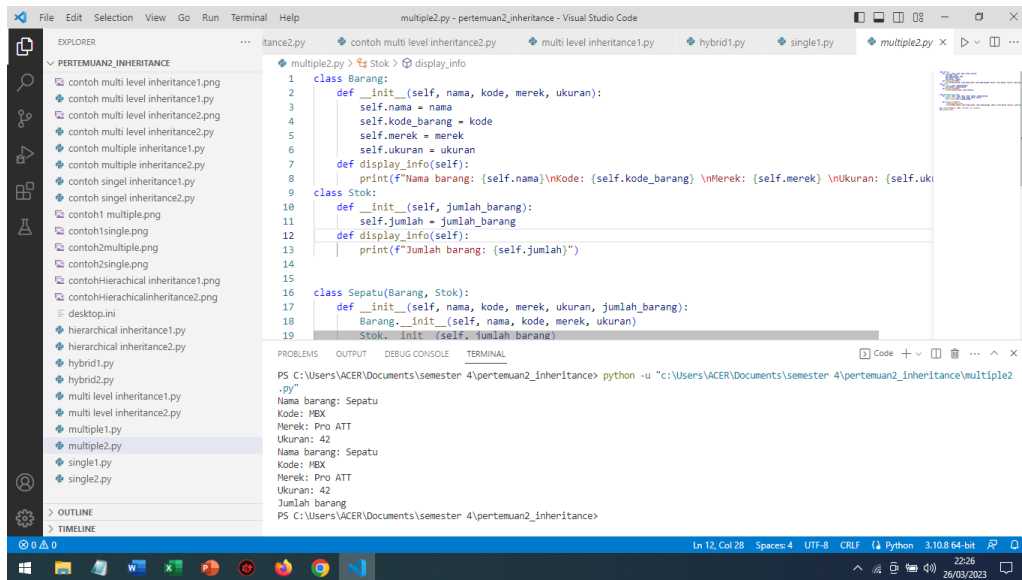
class Sepatu(Barang, Stok):
    def __init__(self, nama, kode, merek, ukuran, jumlah_barang):
        Barang.__init__(self, nama, kode, merek, ukuran)
        Stok.__init__(self, jumlah_barang)

    def display_info(self):
        super().display_info()
        print(f>Nama barang: {self.nama}\nKode: {self.kode_barang} \nMerek: {self.merek}
\nUkuran: {self.ukuran} \nJumlah barang")

spt = Sepatu("Sepatu", "MBX", "Pro ATT", 42, "12 pcs")
spt.display_info()
```



Edit dengan WPS Office



3. hierarichal1.py

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)
```



Edit dengan WPS Office

```
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("Muhammad Alif Taufiqurahman", 210511127, "Teknik", "Teknik Informatika")  
    mhs1.keterangan()  
    mhs1.detail()  
    mhs1.ket()
```

The screenshot displays a Python IDE with the following components:

- Explorer Panel (Left):** Shows a project named 'PERTUMAAN2_INHERITANCE' containing various files including 'contoh_multi_level_inheritance1.png', 'contoh_multi_level_inheritance2.png', 'contoh_multi_level_inheritance2.py', 'contoh_multiple_inheritance1.py', 'contoh_multiple_inheritance2.py', 'contoh_singel_inheritance1.py', 'contoh_singel_inheritance2.py', 'contoh1_multiple.png', 'contoh1_single.png', 'contoh2_multiple.png', 'contoh2_single.png', 'contohHierarchical_inheritance1.png', 'contohHierarchical_inheritance2.png', 'desktop.ini', 'hierarchical_inheritance1.py', 'hierarchical_inheritance2.py', 'hybrid1.py', 'hybrid2.py', 'multi_level_inheritance1.py', 'multi_level_inheritance2.py', 'multiple1.py', 'multiple2.py', 'single1.py', and 'single2.py'. It also shows an 'OUTLINE' and 'TIMELINE' section.
- Main Editor:** Displays the code for 'hierarchical_inheritance1.py'. It defines a base class 'Mahasiswa' with attributes 'name' and 'nim', and methods 'ket()', 'getName()', and 'getNim()'. A derived class 'Fakultas' inherits from 'Mahasiswa' and adds a 'fakultas' attribute. The code is as follows:


```

1 class Mahasiswa:
2     def __init__(self, name, nim):
3         self.name = name
4         self.nim = nim
5
6     def ket(self):
7         print(f'{self.name} adalah Mahasiswa UMC dengan NIM {self.nim}\n')
8
9     def getName(self):
10        return self.name
11
12    def getNim(self):
13        return self.nim
14
15 class Fakultas(Mahasiswa):
16     def __init__(self, name, nim, fakultas):
17         super().__init__(name, nim)
18         self.fakultas = fakultas
19
20 if __name__ == '__main__':
21     cal = Fakultas('Muhammad Alif Taufiqurrahman', '218511127', 'Teknik Informatika')
22     cal.ket()
23     cal.getName()
24     cal.getNim()
25 
```
- Output Panel (Bottom):** Shows the execution results of the script:


```

Muhammad Alif Taufiqurrahman adalah Mahasiswa UMC dengan NIM 21851127
Muhammad Alif Taufiqurrahman
21851127
Teknik Informatika
      
```

Hierarichal2.py

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

```
def getnama(self):  
    return self.nama
```

```
def getkode(self):  
    return self.kode
```

```
class Sandal(Barang):
```

```

def __init__(self,nama,kode,ukuran,merek):
    super().__init__(nama,kode)
    self.ukuran = ukuran
    self.merek = merek

def getukuran(self):
    return self.ukuran

def getmerek(self):
    return self.getmerek

class Model(Sandal):
    def __init__(self, nama, kode, ukuran, merek, model):
        super().__init__(nama, kode, ukuran, merek)
        self.model = model

    def getmodel(self):
        return self.model

    def detail(self):
        print(f'Nama: {self.nama}\nKode: {self.kode}\nUkuran: {self.ukuran}\nMerek: {self.merek}\nModel: {self.model}')

if __name__ == '__main__':
    po = Model("Sandal", "EMW",39,"Carvil","selop")
    po.detail()

```

The screenshot shows the Visual Studio Code editor with a file named 'hierarchical_inheritance2.py' open. The code defines a 'Barang' class with attributes 'nama', 'kode', and methods 'getnama', 'getkode', and 'getukuran'. It then defines a 'Sandal' class that inherits from 'Barang', adding a 'merek' attribute and a 'detail' method. The 'detail' method prints the object's attributes. A main block creates an instance of 'Sandal' and calls its 'detail' method. The terminal at the bottom shows the command to run the script and the resulting output.

```

class Barang:
    def __init__(self,nama,kode):
        self.nama = nama
        self.kode = kode

    def getnama(self):
        return self.nama

    def getkode(self):
        return self.kode

class Sandal(Barang):
    def __init__(self,nama,kode,ukuran,merek):
        super().__init__(nama,kode)
        self.ukuran = ukuran
        self.merek = merek

    def getukuran(self):
        return self.ukuran

    def detail(self):
        print(f'Nama: {self.nama}\nKode: {self.kode}\nUkuran: {self.ukuran}\nMerek: {self.merek}\nModel: {self.model}')

if __name__ == '__main__':
    po = Model("Sandal", "EMW",39,"Carvil","selop")
    po.detail()

```

```

PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance> python -u "C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance\hierarchical_inheritance2.py"
Nama: Sandal
Kode: EMW
Ukuran: 39
Merek: Carvil
Model: selop
PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance>

```

4. multi level1.py

class Person:



Edit dengan WPS Office


```

def __init__(self, name, age):
    self.name = name
    self.age = age
def get_details(self):
    print(f"Name: {self.name}, Age: {self.age}")
class Programmer(Person):
    def __init__(self, name, age, id, salary):
        super().__init__(name, age)
        self.id = id
        self.salary = salary
    def get_details(self):
        super().get_details()
        print(f"ID: {self.id}, Salary: {self.salary}")
class Seniorprogrammer(Programmer):
    def __init__(self, name, age, id, salary, level):
        super().__init__(name, age, id, salary)
        self.level = level
    def get_details(self):
        super().get_details()
        print(f"Level: {self.level}")

orang = Seniorprogrammer("loki", "26", "11", "50000", "Expert")
orang.get_details()

```

```

1 class Person:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5     def get_details(self):
6         print(f"Name: {self.name}, Age: {self.age}")
7 class Programmer(Person):
8     def __init__(self, name, age, id, salary):
9         super().__init__(name, age)
10        self.id = id
11        self.salary = salary
12    def get_details(self):
13        super().get_details()
14        print(f"ID: {self.id}, Salary: {self.salary}")
15 class Seniorprogrammer(Programmer):
16     def __init__(self, name, age, id, salary, level):
17         super().__init__(name, age, id, salary)
18         self.level = level
19     def get_details(self):

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance> python -u "C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance\multi_level_inheritance1.py"
Name: loki, Age: 26
ID: 11, Salary: 50000
Level: Expert
PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance>

```

Multi level2.py

```

class Manusia:
    def __init__(self, name):
        self.name = name

```



Edit dengan WPS Office

```

def speak(self):
    print("Halo saya manusia")
class Level(Manusia):
    def __init__(self, name, level):
        super().__init__(name)
        self.lvl = level
    def speak(self):
        print("saya Level 1")
class Level2(Level):
    def __init__(self, name, level, level_2):
        super().__init__(name, level)
        self.lvl2 = level_2
    def speak(self):
        print("saya telah menaik level")

player = Level2("lopi", "Level 1", "Level up")
print(player.name)
print(player.lvl)
print(player.lvl2)
player.speak()

```

```

class Manusia:
    def __init__(self, name):
        self.name = name
    def speak(self):
        print("Halo saya manusia")
class Level(Manusia):
    def __init__(self, name, level):
        super().__init__(name)
        self.lvl = level
    def speak(self):
        print("saya Level 1")
class Level2(Level):
    def __init__(self, name, level, level_2):
        super().__init__(name, level)
        self.lvl2 = level_2
    def speak(self):
        print("saya telah menaik level")

player = Level2("lopi", "Level 1", "Level up")

```

```

PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance> python -u "c:\Users\VACER\Documents\semester 4\pertemuan2_inheritance\multi lev
el inheritance2.py"
lopi
Level 1
Level up
saya telah menaik level
PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance>

```

5. hybrid1.py

```

class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def get_details(self):
        print(f"Name: {self.name}, Age: {self.age}")

```



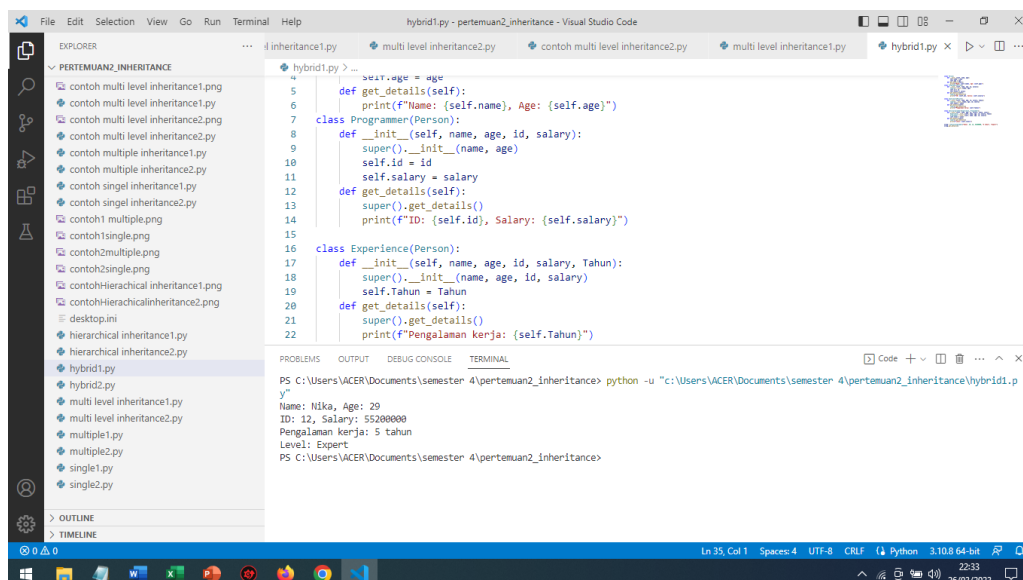
Edit dengan WPS Office

```
class Programmer(Person):
    def __init__(self, name, age, id, salary):
        super().__init__(name, age)
        self.id = id
        self.salary = salary
    def get_details(self):
        super().get_details()
        print(f"ID: {self.id}, Salary: {self.salary}")
```

```
class Experience(Person):
    def __init__(self, name, age, id, salary, Tahun):
        super().__init__(name, age, id, salary)
        self.Tahun = Tahun
    def get_details(self):
        super().get_details()
        print(f"Pengalaman kerja: {self.Tahun}")
```

```
class Seniorprogrammer(Experience, Programmer):
    def __init__(self, name, age, id, salary, Tahun, Level):
        Experience.__init__(self, name, age, id, salary, Tahun)
        Programmer.__init__(self, name, age, id, salary)
        self.level = Level
    def get_details(self):
        super().get_details()
        print(f"Level: {self.level}")
```

```
orang = Seniorprogrammer("Nika", 29, 12, 55200000, "5 tahun", "Expert")
orang.get_details()
```



```
hybrid1.py > ...
5     def get_details(self):
6         print(f"Name: {self.name}, Age: {self.age}")
7     class Programmer(Person):
8         def __init__(self, name, age, id, salary):
9             super().__init__(name, age)
10            self.id = id
11            self.salary = salary
12        def get_details(self):
13            super().get_details()
14            print(f"ID: {self.id}, Salary: {self.salary}")
15
16    class Experience(Person):
17        def __init__(self, name, age, id, salary, Tahun):
18            super().__init__(name, age, id, salary)
19            self.Tahun = Tahun
20        def get_details(self):
21            super().get_details()
22            print(f"Pengalaman kerja: {self.Tahun}")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance> python -u "C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance\hybrid1.py"
Name: Nika, Age: 29
ID: 12, Salary: 55200000
Pengalaman kerja: 5 tahun
Level: Expert
PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance>
```

Hybrid2.py



Edit dengan WPS Office

```

class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age
    def get_details(self):
        print(f"Name: {self.name}, Age: {self.age}")

class Player(Person):
    def __init__(self, name, age, No, salary):
        super().__init__(name, age)
        self.no = No
        self.salary = salary
    def get_details(self):
        super().get_details()
        print(f"No punggung: {self.no}, Salary: {self.salary}")

class Experience(Person):
    def __init__(self, name, age, no, salary, Trophy):
        super().__init__(name, age, no, salary)
        self.Tropi = Trophy
    def get_details(self):
        super().get_details()
        print(f"Trophy yang diraih: {self.Tropi}")

class GoatPlayer(Experience, Player):
    def __init__(self, name, age, no, salary, Trophy, Level):
        Experience.__init__(self, name, age, no, salary, Trophy)
        Player.__init__(self, name, age, no, salary)
        self.level = Level
    def get_details(self):
        super().get_details()
        print(f"Level: {self.level}")

orang = GoatPlayer("Cristiano Ronaldo", 38, 7, 55200000, "34 Trophy", "GOAT")
orang.get_details()

```



```
1 class Person:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5     def get_details(self):
6         print(f"Name: {self.name}, Age: {self.age}")
7 class Player(Person):
8     def __init__(self, name, age, no, salary):
9         super().__init__(name, age)
10        self.no = no
11        self.salary = salary
12    def get_details(self):
13        super().get_details()
14        print(f"No punggung: {self.no}, Salary: {self.salary}")
15
16 class Experience(Person):
17     def __init__(self, name, age, no, salary, Trophy):
18         super().__init__(name, age, no, salary)
19         self.Troopi = Trophy
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance> python -u "C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance\hybrid2.py"

Name: Cristiano Ronaldo, Age: 38
No punggung: 7, Salary: 55200000
Trophy yang diraih: 34 Trophy
Level: GOAT

PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance> python -u "C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance\hybrid2.py"

Name: Cristiano Ronaldo, Age: 38
No punggung: 7, Salary: 55200000
Trophy yang diraih: 34 Trophy
Level: GOAT

PS C:\Users\VACER\Documents\semester 4\pertemuan2_inheritance>



Edit dengan WPS Office



Edit dengan WPS Office