Nama : Purwoto

NIM : 210511014

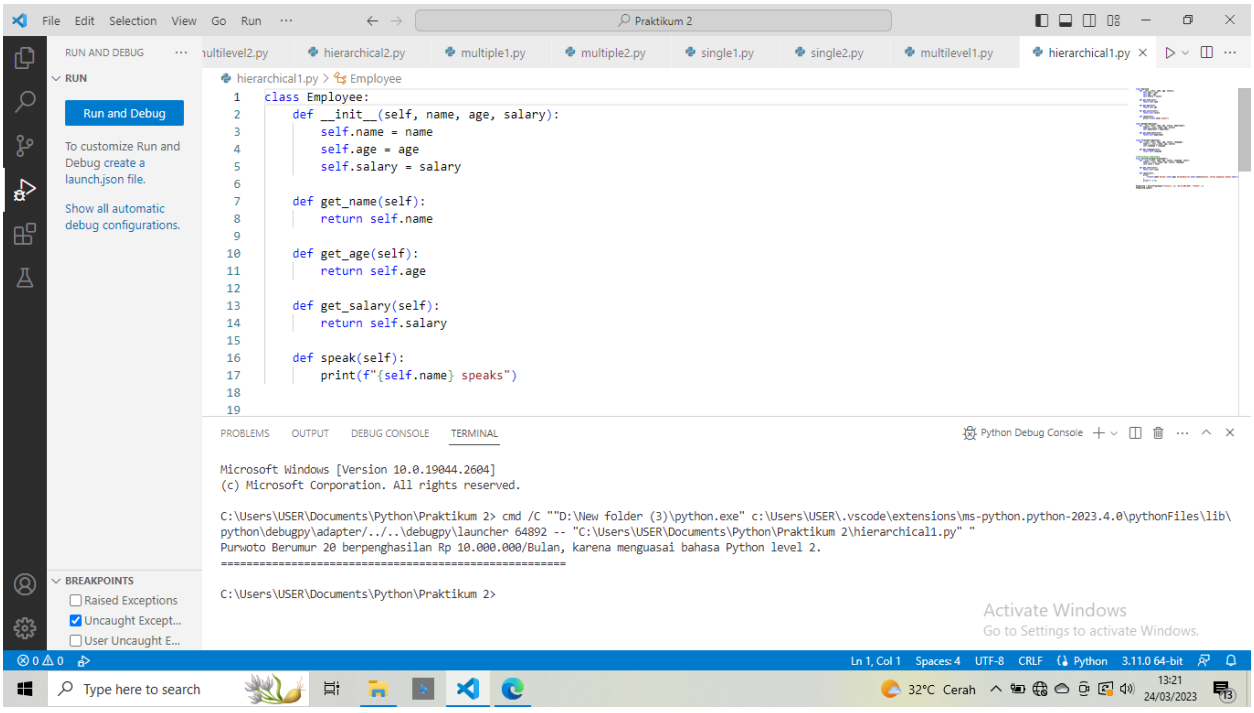
Kelas : R1

Tugas Parktikum PBO2

Hierarchical1.py

```
class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

    def get_name(self):
        return self.name

    def get_age(self):
        return self.age

    def get_salary(self):
        return self.salary

    def speak(self):
        print(f"{self.name} speaks")


class Manager(Employee):
    def __init__(self, name, age, salary, department):
        super().__init__(name, age, salary)
        self.department = department

    def get_department(self):
        return self.department


class Programmer(Employee):
    def __init__(self, name, age, salary, language):
        super().__init__(name, age, salary)
        self.language = language

    def get_language(self):
        return self.language
```

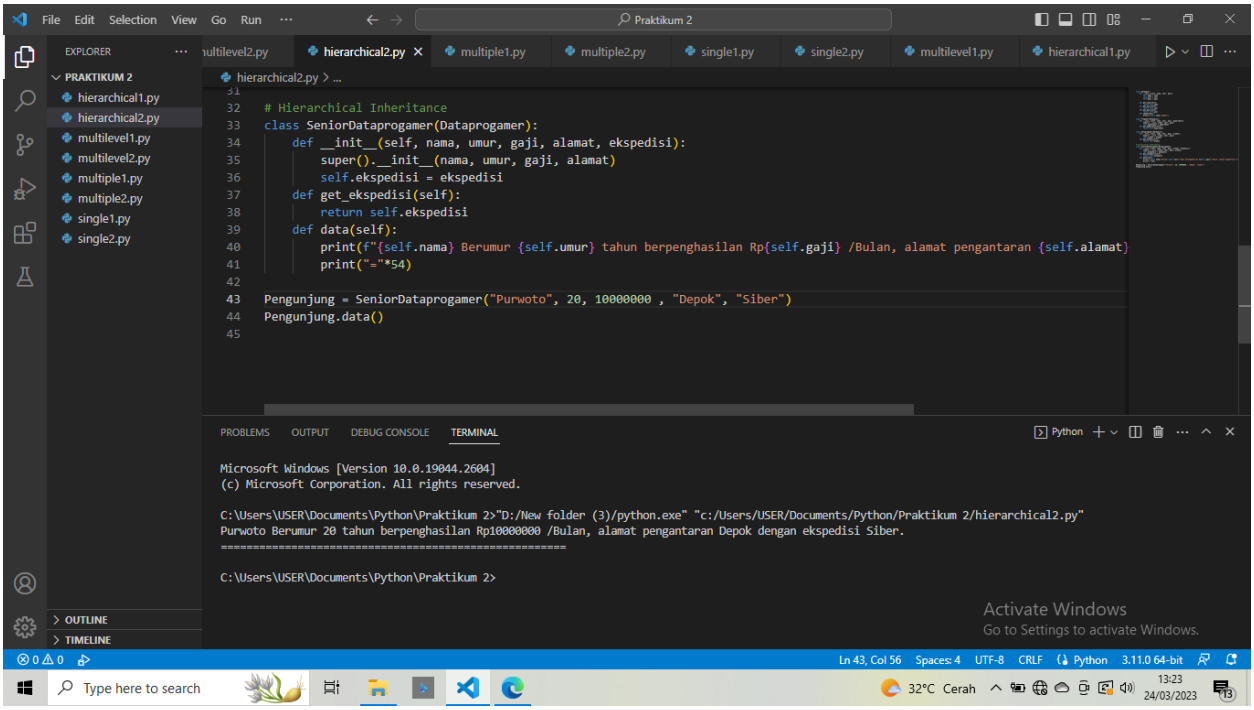
```python
# Hierarchical Inheritance
class SeniorProgrammer(Programmer):
    def __init__(self, name, age, salary, language, level):
        super().__init__(name, age, salary, language)
        self.level = level

    def get_level(self):
        return self.level

    def speak(self):
        print(
            f"{self.name} Berumur {self.age} berpenghasilan {self.salary}/Bulan,
karena menguasai bahasa {self.language} level {self.level}."
        )
        print("=" * 54)


Pengunjung = SeniorProgrammer("Purwoto", 20, "Rp 10.000.000", "Python", 2)
Pengunjung.speak()
```



Hierarchical 2

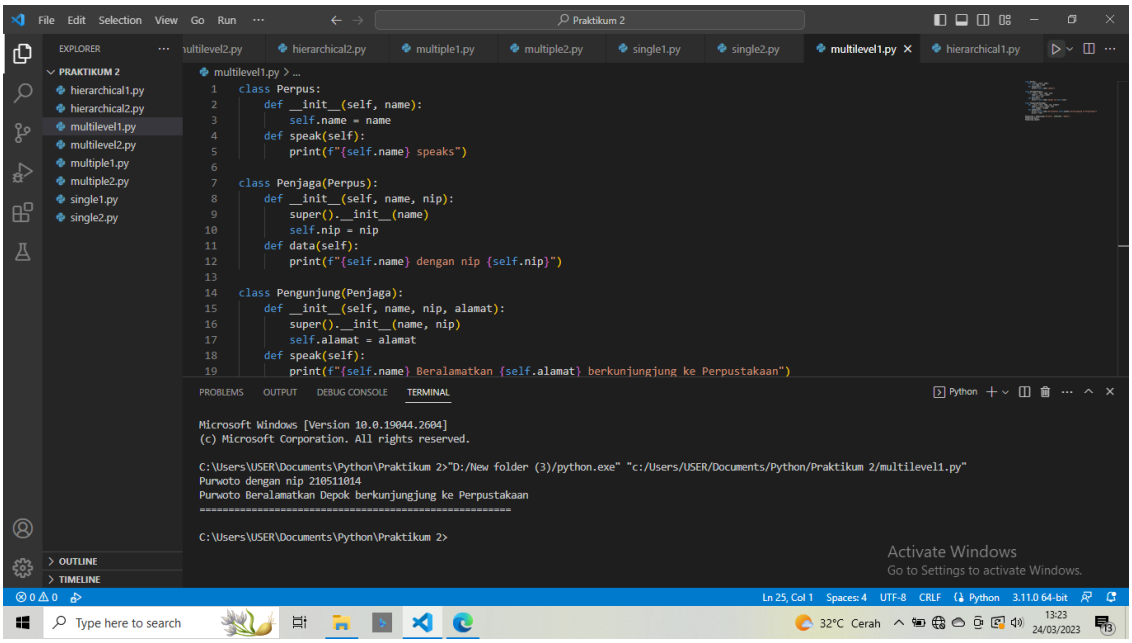
```
class Ekspedisi:
    def __init__(self, nama, umur, gaji):
        self.nama = nama
        self.umur = umur
        self.gaji = gaji

    def get_nama(self):
        return self.nama
    def get_umur(self):
        return self.umur
    def get_gaji(self):
        return self.gaji
    def speak(self):
        print(f"{self.nama} speaks")

class Progamer(Ekspedisi):
    def __init__(self, nama, umur, gaji, department):
        super().__init__(nama, umur, gaji)
        self.department = department
    def get_department(self):
        return self.department

class Dataprogamer(Ekspedisi):
    def __init__(self, nama, umur, gaji, alamat):
        super().__init__(nama, umur, gaji)
        self.alamat = alamat
    def get_alamat(self):
        return self.alamat

# Hierarchical Inheritance
class SeniorDataprogamer(Dataprogamer):
    def __init__(self, nama, umur, gaji, alamat, ekspedisi):
        super().__init__(nama, umur, gaji, alamat)
        self.ekspedisi = ekspedisi
    def get_ekspedisi(self):
        return self.ekspedisi
    def data(self):
        print(f"{self.nama} Berumur {self.umur} tahun berpenghasilan
Rp{self.gaji} /Bulan, alamat pengantaran {self.alamat} dengan ekspedisi
{self.ekspedisi}.")
        print("="*54)

Pengunjung = SeniorDataprogamer("Purwoto", 20, 10000000 , "Depok", "Siber")
```

Pengunjung.data()



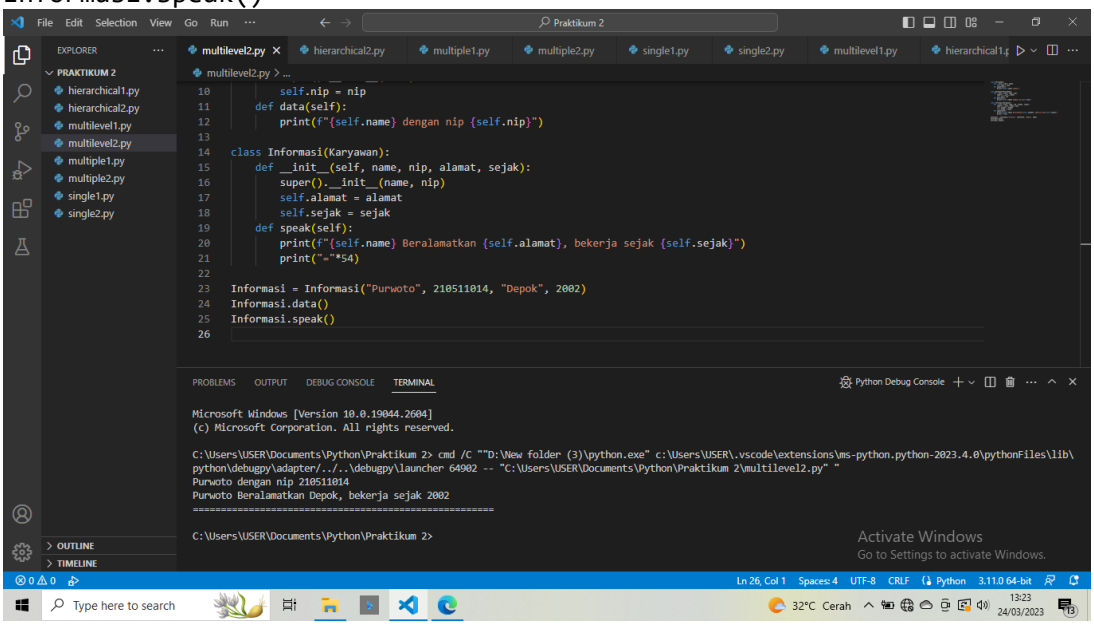
Multivel 1

```
class Perpus:

    def __init__(self, name):
        self.name = name
    def speak(self):
        print(f"{self.name} speaks")

class Penjaga(Perpus):
    def __init__(self, name, nip):
        super().__init__(name)
        self.nip = nip
    def data(self):
        print(f"{self.name} dengan nip {self.nip}")

class Pengunjung(Penjaga):
    def __init__(self, name, nip, alamat):
        super().__init__(name, nip)
        self.alamat = alamat
    def speak(self):
        print(f"{self.name} Beralamatkan {self.alamat} berkunjungjung ke
Perpustakaan")
        print("="*54)

Pengunjung = Pengunjung("Purwoto", 210511014, "Depok")
Pengunjung.data()
Pengunjung.speak()
```



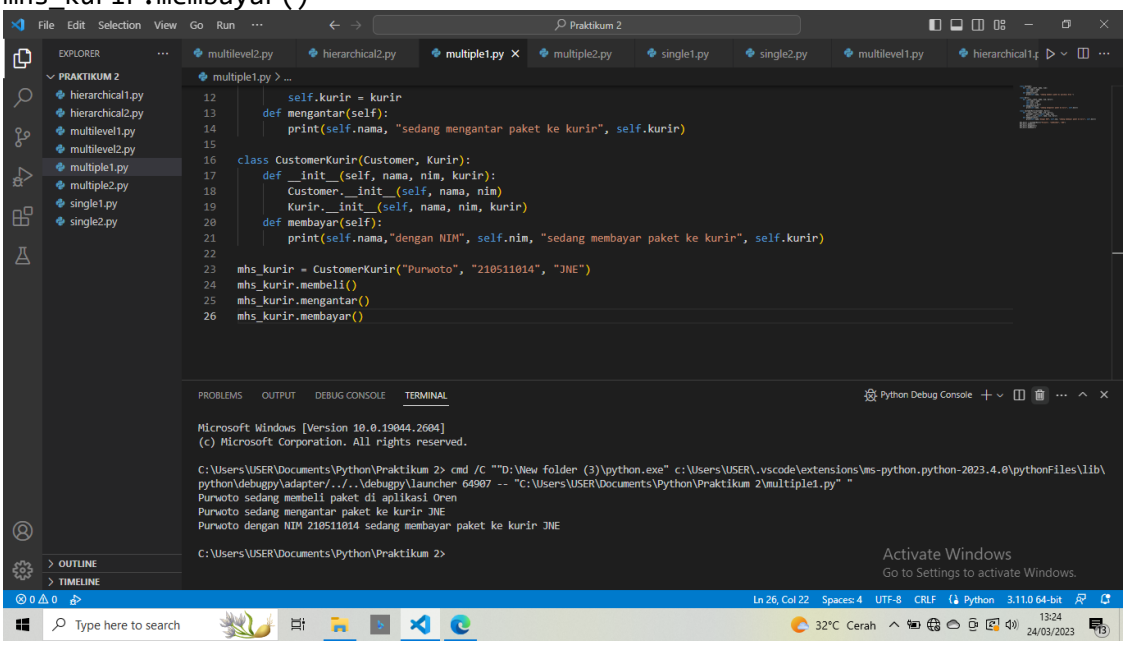
Muiltivel 2

```
class Perusahaan:
    def __init__(self, name):
        self.name = name
    def speak(self):
        print(f"{self.name} speaks")

class Karyawan(Perusahaan):
    def __init__(self, name, nip):
        super().__init__(name)
        self.nip = nip
    def data(self):
        print(f"{self.name} dengan nip {self.nip}")

class Informasi(Karyawan):
    def __init__(self, name, nip, alamat, sejak):
        super().__init__(name, nip)
        self.alamat = alamat
        self.sejak = sejak
    def speak(self):
        print(f"{self.name} Beralamatkan {self.alamat}, bekerja sejak
{self.sejak}")
        print("="*54)

Informasi = Informasi("Purwoto", 210511014, "Depok", 2002)
Informasi.data()
Informasi.speak()
```



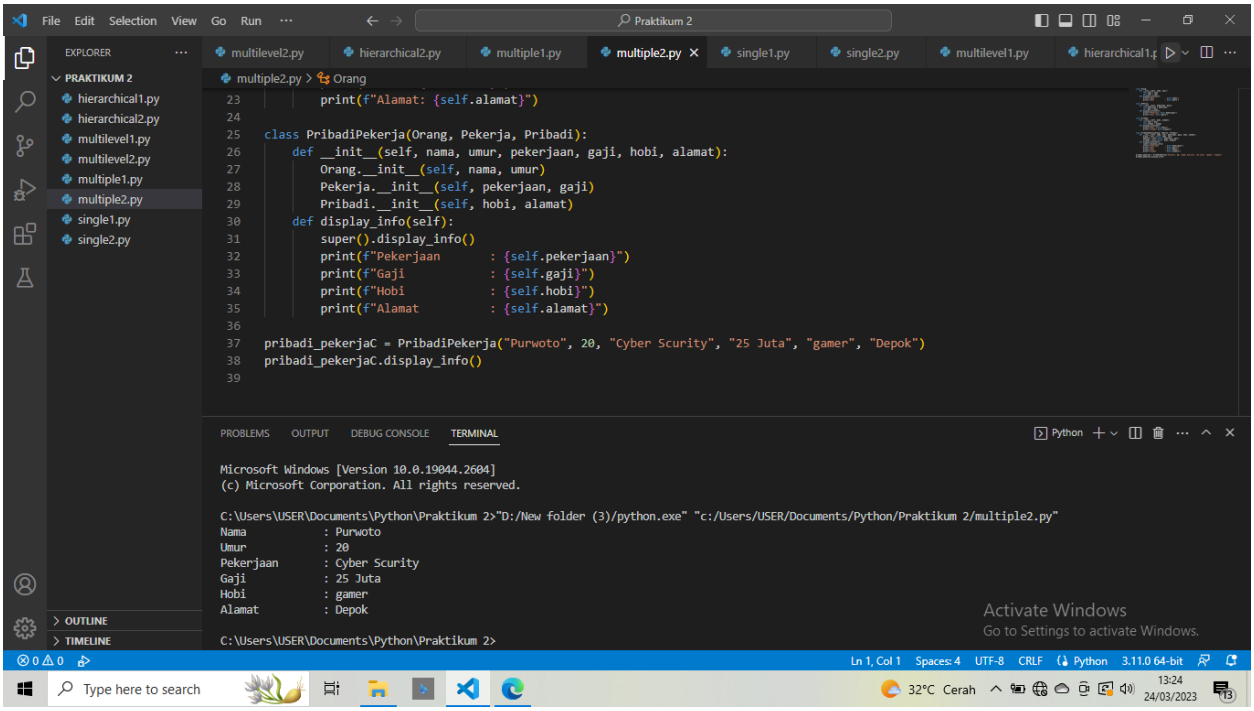
multiple1.py

```
class Customer:
    def __init__(self, nama, nim):
        self.nama = nama
        self.nim = nim
    def membeli(self):
        print(self.nama, "sedang membeli paket di aplikasi Oren ")

class Kurir:
    def __init__(self, nama, nim, kurir):
        self.nama = nama
        self.nim = nim
        self.kurir = kurir
    def mengantar(self):
        print(self.nama, "sedang mengantar paket ke kurir", self.kurir)

class CustomerKurir(Customer, Kurir):
    def __init__(self, nama, nim, kurir):
        Customer.__init__(self, nama, nim)
        Kurir.__init__(self, nama, nim, kurir)
    def membayar(self):
        print(self.nama,"dengan NIM", self.nim, "sedang membayar paket ke kurir",
self.kurir)

mhs_kurir = CustomerKurir("Purwoto", "210511014", "JNE")
mhs_kurir.membeli()
mhs_kurir.mengantar()
mhs_kurir.membayar()
```



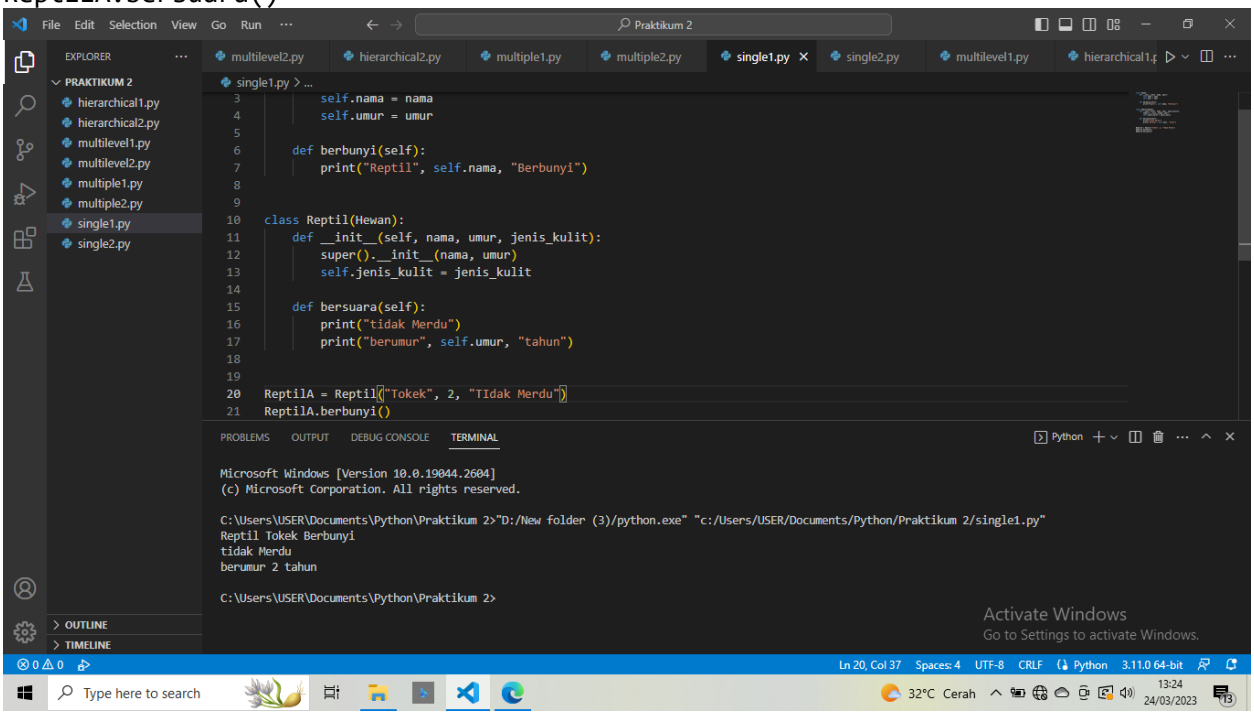
Multiple2.py

```
class Orang:
    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 Pekerja:
    def __init__(self, pekerjaan, gaji):
        self.pekerjaan = pekerjaan
        self.gaji = gaji
    def display_info(self):
        print(f"Pekerjaan: {self.pekerjaan}")
        print(f"Gaji: {self.gaji}")

class Pribadi:
    def __init__(self, hobi, alamat):
        self.hobi = hobi
        self.alamat = alamat
    def display_info(self):
        print(f"Hobi: {self.hobi}")
        print(f"Alamat: {self.alamat}")

class PribadiPekerja(Orang, Pekerja, Pribadi):
    def __init__(self, nama, umur, pekerjaan, gaji, hobi, alamat):
        Orang.__init__(self, nama, umur)
        Pekerja.__init__(self, pekerjaan, gaji)
        Pribadi.__init__(self, hobi, alamat)
    def display_info(self):
        super().display_info()
        print(f"Pekerjaan       : {self.pekerjaan}")
        print(f"Gaji            : {self.gaji}")
        print(f"Hobi            : {self.hobi}")
        print(f"Alamat          : {self.alamat}")

pribadi_pekerjaC = PribadiPekerja("Purwoto", 20, "Cyber Scurity", "25 Juta",
"gamer", "Depok")
pribadi_pekerjaC.display_info()
```


File Edit Selection View Go Run
Praktikum 2
EXPLORER
PRAKTIKUM 2
hierarchical1.py
hierarchical2.py
multilevel1.py
multilevel2.py
multiple1.py
multiple2.py
single1.py
single2.py
multiple2.py > Orang
print(f"Alamat: {self.alamat}")
class PribadiPekerja(Orang, Pekerja, Pribadi):
def __init__(self, nama, umur, pekerjaan, gaji, hobi, alamat):
Orang.__init__(self, nama, umur)
Pekerja.__init__(self, pekerjaan, gaji)
Pribadi.__init__(self, hobi, alamat)
def display_info(self):
super().display_info()
print(f"Pekerjaan : {self.pekerjaan}")
print(f"Gaji : {self.gaji}")
print(f"Hobi : {self.hobi}")
print(f"Alamat : {self.alamat}")
pribadi_pekerjaC = PribadiPekerja("Purwoto", 20, "Cyber Scurity", "25 Juta", "gamer", "Depok")
pribadi_pekerjaC.display_info()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.
C:\Users\USER\Documents\Python\Praktikum 2>"D:/New folder (3)/python.exe" "c:/Users/USER/Documents/Python/Praktikum 2/multiple2.py"
Nama : Purwoto
Umur : 20
Pekerjaan : Cyber Scurity
Gaji : 25 Juta
Hobi : gamer
Alamat : Depok
C:\Users\USER\Documents\Python\Praktikum 2>
Activate Windows
Go to Settings to activate Windows.
OUTLINE
TIMELINE
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.0 64-bit
Type here to search
32°C Cerah
13:24
24/03/2023

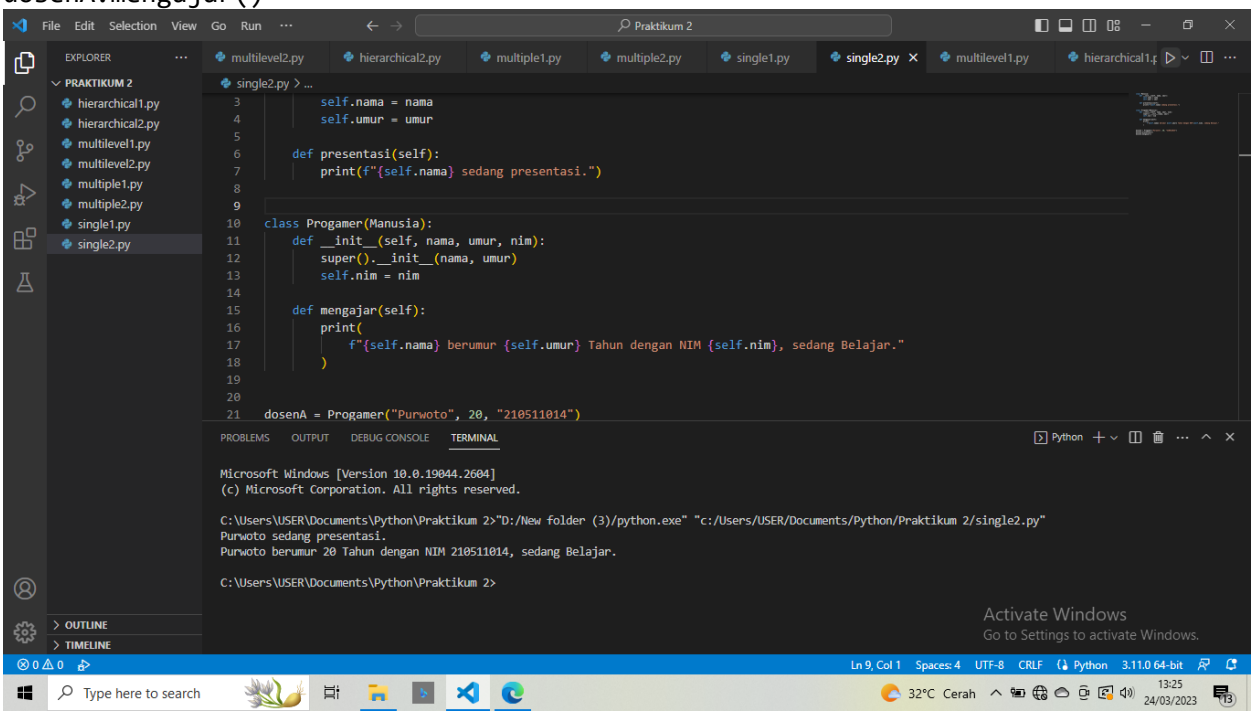
single1.py,

```python
class Hewan:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur

    def berbunyi(self):
        print("Reptil", self.nama, "Berbunyi")


class Reptil(Hewan):
    def __init__(self, nama, umur, jenis_kulit):
        super().__init__(nama, umur)
        self.jenis_kulit = jenis_kulit

    def bersuara(self):
        print("tidak Merdu")
        print("berumur", self.umur, "tahun")


ReptilA = Reptil("Tokek", 2, "TIdak Merdu")
ReptilA.berbunyi()
ReptilA.bersuara()
```



Single2.py

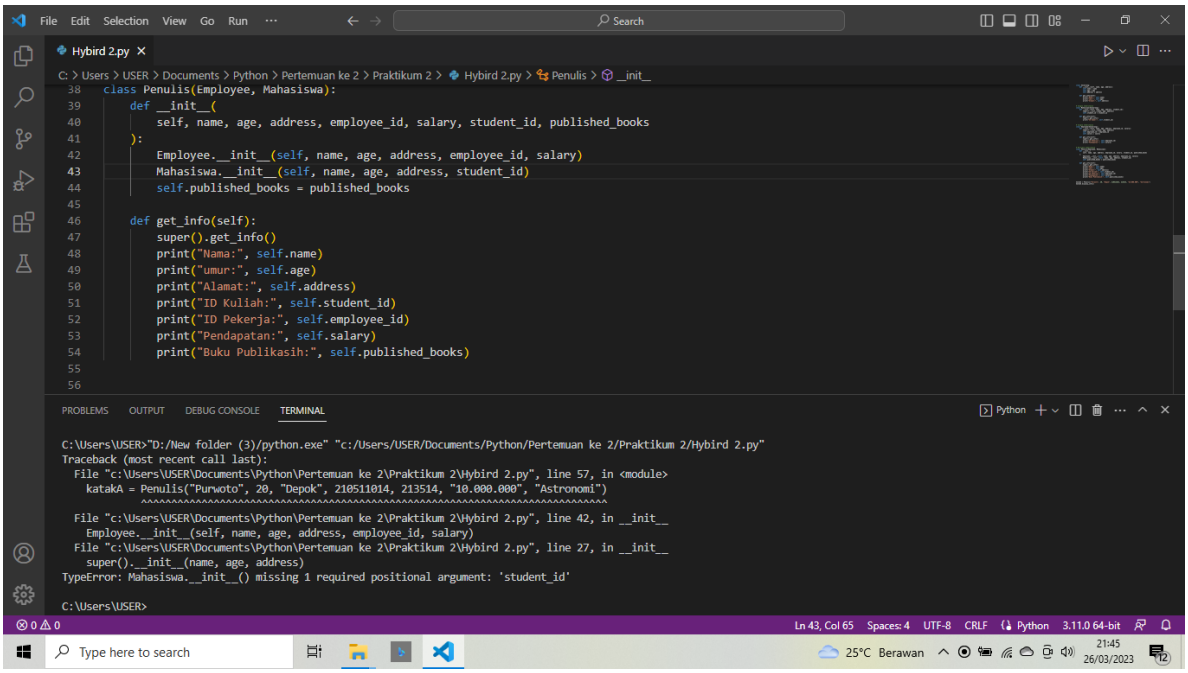
```
class Manusia:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur

    def presentasi(self):
        print(f"{self.nama} sedang presentasi.")


class Progamer(Manusia):
    def __init__(self, nama, umur, nim):
        super().__init__(nama, umur)
        self.nim = nim

    def mengajar(self):
        print(
            f"{self.nama} berumur {self.umur} Tahun dengan NIM {self.nim}, sedang
Belajar."
        )


dosenA = Progamer("Purwoto", 20, "210511014")
dosenA.presentasi()
dosenA.mengajar()
```



Hybrid1.py

```
class Seseorang:
    def __init__(self, name, age, address):
        self.name = name
        self.age = age
        self.address = address

    def get_info(self):
        print("Nama:", self.name)
        print("umur:", self.age)
        print("Alamat:", self.address)


# Single Inheritance
class Mahasiswa(Seseorang):
    def __init__(self, name, age, address, student_id):
        super().__init__(name, age, address)
        self.student_id = student_id

    def get_info(self):
        super().get_info()
        print("ID Gamer:", self.student_id)


# Single Inheritance
class Employee(Seseorang):
    def __init__(self, name, age, address, employee_id, salary):
        super().__init__(name, age, address)
        self.employee_id = employee_id
        self.salary = salary

    def get_info(self):
        super().get_info()
        print("ID Pekerja:", self.employee_id)
        print("Pendapatan:", self.salary)


# Multiple Inheritance
class Penulis(Employee, Mahasiswa):
    def __init__(
        self, name, age, address, employee_id, salary, student_id,
published_books
    ):
        Employee.__init__(self, name, age, address, employee_id, salary)
```

```
        Mahasiswa.__init__(self, name, age, address, student_id)
        self.published_books = published_books

    def get_info(self):
        super().get_info()
        print("Nama:", self.name)
        print("umur:", self.age)
        print("Alamat:", self.address)
        print("ID Kuliah:", self.student_id)
        print("ID Pekerja:", self.employee_id)
        print("Pendapatan:", self.salary)
        print("Buku Publikasih:", self.published_books)


katakA = Penulis("Purwoto", 20, "Depok", 210511014, 213514, "10.000.000",
"Astronomi")
katakA.display_info()
```



Hybrid2.py

```
class Seseorang:
    def __init__(self, name, age, address):
        self.name = name
        self.age = age
        self.address = address

    def get_info(self):
        print("Nama:", self.name)
        print("umur:", self.age)
        print("Alamat:", self.address)


# Single Inheritance
class Mahasiswa(Seseorang):
    def __init__(self, name, age, address, student_id):
        super().__init__(name, age, address)
        self.student_id = student_id

    def get_info(self):
        super().get_info()
        print("ID Kuliah:", self.student_id)


# Single Inheritance
class Employee(Seseorang):
    def __init__(self, name, age, address, employee_id, salary):
        super().__init__(name, age, address)
        self.employee_id = employee_id
        self.salary = salary

    def get_info(self):
        super().get_info()
        print("ID Pekerja:", self.employee_id)
        print("Pendapatan:", self.salary)


# Multiple Inheritance
class Penulis(Employee, Mahasiswa):
    def __init__(
        self, name, age, address, employee_id, salary, student_id,
published_books
    ):
        Employee.__init__(self, name, age, address, employee_id, salary)
        Mahasiswa.__init__(self, name, age, address, student_id)
```

```
        self.published_books = published_books

    def get_info(self):
        super().get_info()
        print("Nama:", self.name)
        print("umur:", self.age)
        print("Alamat:", self.address)
        print("ID Kuliah:", self.student_id)
        print("ID Pekerja:", self.employee_id)
        print("Pendapatan:", self.salary)
        print("Buku Publikasih:", self.published_books)


katakA = Penulis("Purwoto", 20, "Depok", 210511014, 213514, "5.000.000",
"Astronomi")
katakA.display_info()
```