LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023

Prepared By:

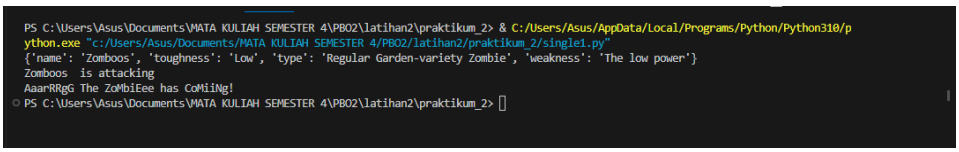
Nama : Nisa Maharani
Nim : 210511136
Kelas : Reguler 4

A. Single Inheritance

Script 1:

```
class Enemy:

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

    def move(self):
        print(self.name, " is attacking")

class Zombie(Enemy):

    def __init__(self, name, toughness, type, weakness):
        super().__init__(name, toughness)
        self.type = type
        self.weakness = weakness

    def sounds(self):
        print("AaarRRgG The ZoMbiEee has CoMiiNg!")

zombie = Zombie("Zomboos", 'Low', "Regular Garden-variety Zombie", 'The low
power')
print(zombie.__dict__)
zombie.move()
zombie.sounds()
```

Output:

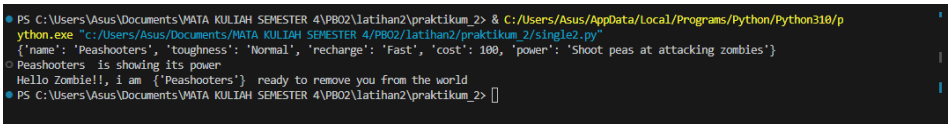


Script 2:

```
class Hero:

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

    def move(self):
        print(self.name,' is showing its power')


class Plant(Hero):
```

```
    def __init__(self, name, toughness, recharge, cost, power):
        super().__init__(name, toughness)
        self.recharge = recharge
        self.cost = cost
        self.power = power

    def desc(self):
        print(f'Hello Zombie!!, i am ', {self.name}, ' ready to remove you
from the world')


plant = Plant('Peashooters','Normal','Fast',100,'Shoot peas at attacking
zombies')
print(plant.__dict__)
plant.move()
plant.desc()
```

Output:



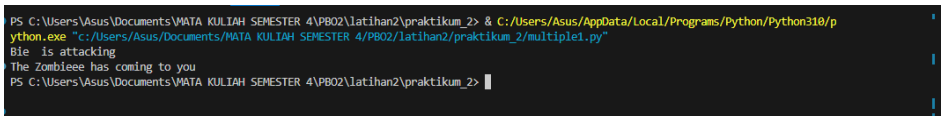
B. Multiple Inheritance

Script 1:

```
class Enemy:

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

    def move(self):
        print(self.name, " is attacking")

class Zombie:

    def __init__(self, type, weakness):
        self.type = type
        self.weakness = weakness

    def sounds(self):
        print("AaarRRgG The ZoMbiEee has CoMiiNg!")


class FlagZombie(Enemy, Zombie):
```

```
    def __init__(self, name, toughness, type, weakness):
        Enemy.__init__(self, name, toughness)
        Zombie.__init__(self, type, weakness)

    def do(self):
        print(self.name, ' is attacking')
        print("The Zombieee has coming to you")


zom = FlagZombie('Bie', 'Low', 'Flag Zombie', 'The low power')
zom.do()
```

Output:

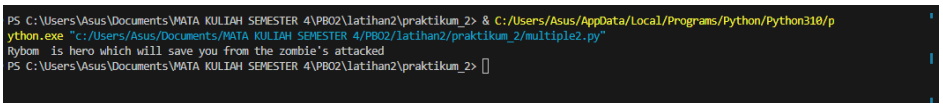


Script 2:

```
class Hero:

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

    def move(self):
        print(self.name, " is showing its power")

class Plant:

    def __init__(self, recharge, cost, power):
        self.recharge = recharge
        self.cost = cost
        self.power = power

    def plantpower(self):
        print(f'my power is  {self.power}  ready to remove ZOMBIE from the
world')


class CherryBombs(Hero, Plant):

    def __init__(self, name, toughness, recharge, cost, power):
        Hero.__init__(self, name, toughness)
        Plant.__init__(self, recharge, cost, power)

    def do(self):
```

```
        print(self.name,' is hero which will save you from the zombie\'s
attacked')


cheri = CherryBombs('Rybom', 'Massive', 'Very slow', '150', 'Blow up all
zombies in an area')
cheri.do()
```

Output:



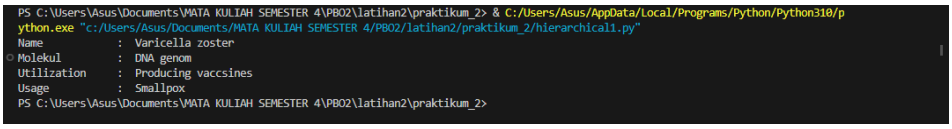
C. Hierarchical Inheritance

Script 1:

```
class Virus:

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

    def get_name(self):
        return self.name

    def get_molekul(self):
        return self.molekul


class VirusCategory(Virus):

    def __init__(self, name, molekul, category):
        super().__init__(name, molekul)
        self.category = category

    def get_category(self):
        return self.category


class VirusUsage(Virus):

    def __init__(self, name, molekul, utilization):
        super().__init__(name, molekul)
        self.utilization = utilization

    def get_utilization(self):
        return self.utilization
```

```
# Hierarchical Inheritance
class VirusVacsin(VirusUsage):

    def __init__(self, name, molekul, utilization, vaksin_for):
        super().__init__(name, molekul, utilization)
        self.vaksin_for = vaksin_for

    def get_vaksin_for(self):
        return self.vaksin_for

virusscandal = VirusVacsin('Varicella zoster', 'DNA genom',  'Producing
vaccsines', 'Smallpox')
print('Name\t\t: ',virusscandal.get_name())
print('Molekul\t\t: ',virusscandal.get_molekul())
print('Utilization\t: ',virusscandal.get_utilization())
print('Usage\t\t: ',virusscandal.get_vaksin_for())
```

Output:



Script 2:

```
class Bakteri:

    def __init__(self, nama, kapsul):
        self.nama = nama
        self.kapsul = kapsul

    def get_nama(self):
        return self.nama

    def get_kapsul(self):
        return self.kapsul


class BacterialGrouping(Bakteri):

    def __init__(self, nama, kapsul, pengelompokan):
        super().__init__(nama, kapsul)
        self.kelompok = pengelompokan

    def get_kelompok(self):
        return self.kelompok
```

```
class BacteriShape(Bakteri):

    def __init__(self, nama, kapsul, bentuk):
        super().__init__(nama, kapsul)
        self.bentuk = bentuk

    def get_bentuk(self):
        return self.bentuk

# Hierarchical Inheritance
class Stafilokokus(BacteriShape):

    def __init__(self, nama, kapsul, bentuk, jenis ):
        super().__init__(nama, kapsul, bentuk)
        self.jenis = jenis

    def get_jenis(self):
        return self.jenis

stafil = Stafilokokus('Staphylococcus aureus', 'Tidak berkapsul',
'Kokus(bulat)', 'Stafilokokus')
print('Nama bakteri\t: ',stafil.get_nama())
print('Kapsul\t\t: ',stafil.get_kapsul())
print('Bentuk\t\t: ',stafil.get_bentuk())
print('Jenis\t\t: ',stafil.get_jenis())
```

Output:

```
PS C:\Users\Asus\Documents\MATA KULIAH SEMESTER 4\PBO2\latihan2\praktikum_2> & C:/Users/Asus/AppData/Local/Programs/Python/Python310/p
ython.exe "c:/Users/Asus/Documents/MATA KULIAH SEMESTER 4/PBO2/latihan2/praktikum_2/hierarchical2.py"
Nama bakteri    :  Staphylococcus aureus
Kapsul          :  Tidak berkapsul
Bentuk          :  Kokus(bulat)
Jenis           :  Stafilokokus
PS C:\Users\Asus\Documents\MATA KULIAH SEMESTER 4\PBO2\latihan2\praktikum_2>
```

D. Multilevel Inheritance

Script 1:

```
class Protista:

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

    def get_details(self):
        print(f"Name\t\t: {self.name}")
        print(f"Group\t\t: {self.group}")
```

```
class ProtistaClasification(Protista):

    def __init__(self, name, group, sel, sifat):
        super().__init__(name, group)
        self.sel = sel
        self.sifat = sifat

    def get_details(self):
        super().get_details()
        print(f"Sel\t\t: {self.sel}")
        print(f"Sifat\t\t: {self.sifat}")


class Ciliata(ProtistaClasification):

    def __init__(self, name, group, sel, sifat, cara_gerak):
        super().__init__(name, group, sel, sifat)
        self.cara_gerak = cara_gerak

    def get_details(self):
        super().get_details()
        print(f"Cara gerak\t: {self.cara_gerak}")


manager = Ciliata('Ciliata', 'Protista mirip hewan', 'Sel
tunggal','Heterotrof', 'Bergerak dengan rambut getar (silia)')
manager.get_details()
```

Output:

```
PS C:\Users\Asus\Documents\MATA KULIAH SEMESTER 4\PBO2\latihan2\praktikum_2> & C:/Users/Asus/AppData/Local/Programs/Python/Python310/p
ython.exe "c:/Users/Asus/Documents/MATA KULIAH SEMESTER 4/PBO2/latihan2/praktikum_2/multilevel1.py"
Name            : Ciliata
Group           : Protista mirip hewan
Sel             : Sel tunggal
Sifat           : Heterotrof
Cara gerak      : Bergerak dengan rambut getar (silia)
PS C:\Users\Asus\Documents\MATA KULIAH SEMESTER 4\PBO2\latihan2\praktikum_2>
```

Script 2:

```
class Fungi:

    def __init__(self, nama, bentuk, warna):
        self.nama = nama
        self.bentuk = bentuk
        self.warna = warna

    def get_info(self):
        print(f'Nama\t: {self.nama}')
        print(f'Bentuk\t: {self.bentuk}')
        print(f'warna\t: {self.warna}')
```

```
class Zygomycota(Fungi):

    def __init__(self, nama, bentuk, warna, sifat):
        super().__init__(nama,bentuk,warna)
        self.sifat = sifat

    def get_info(self):
        super().get_info()
        print(f'Sifat\t: {self.sifat}')


class Rhizopus(Zygomycota):

    def __init__(self, nama, bentuk, warna, sifat, peran):
        super().__init__(nama, bentuk, warna, sifat)
        self.peran = peran

    def get_info(self):
        super().get_info()
        print(f'Peran\t: {self.peran}')


mycota = Rhizopus('Rhizopus Oryzae', 'Stolon berdinding halus', 'Koloni
berwarna putih berangsur-angsur menjadi abu-abu', 'Saprofit', 'Pembuatan
tempe')
print(mycota.get_info())
```

Output:

```
PS C:\Users\Asus\Documents\MATA KULIAH SEMESTER 4\PBO2\latihan2\praktikum_2> & C:/Users/Asus/AppData/Local/Programs/Python/Python310/p
ython.exe "c:/Users/Asus/Documents/MATA KULIAH SEMESTER 4/PBO2/latihan2/praktikum_2/multilevel2.py"
Nama    : Rhizopus Oryzae
Bentuk  : Stolon berdinding halus
warna   : Koloni berwarna putih berangsur-angsur menjadi abu-abu
Sifat   : Saprofit
Peran   : Pembuatan tempe
```

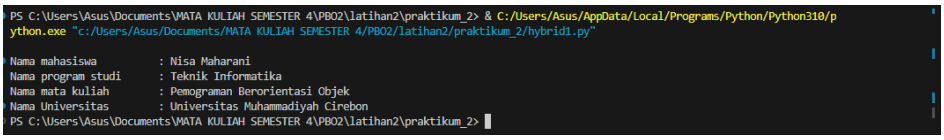
E. Hybrid inheritance

Script 1:

```
class Universitas:
  def __init__(self):
    self.univ = "Universitas Muhammadiyah Cirebon"
  def display(self):
    print(f"Nama Universitas\t: {self.univ}")


class Matkul(Universitas):
  def __init__(self):
    Universitas.__init__(self)
    self.course = "Pemograman Berorientasi Objek"
```

```
  def display(self):
    print(f"Nama mata kuliah\t: {self.course}")
    Universitas.display(self)


class Prodi(Universitas):
  def __init__(self):
    self.Prodi = "Teknik Informatika"
  def display(self):
    print(f"Nama program studi\t: {self.Prodi}")


class Mahasiswa(Matkul, Prodi):
  def __init__(self):
    self.name = "Nisa Maharani"
    Prodi.__init__(self)
    Matkul.__init__(self)
  def display(self):
    print(f"Nama mahasiswa\t\t: {self.name}")
    Prodi.display(self)
    Matkul.display(self)


ob = Mahasiswa()
print()
ob.display()
```

Output:



Script 2:

```
class Univ:

    def __init__(self, nama_univ):
        self.nama_univ = nama_univ

    def desc(self):
        print(f'Universitas\t: {self.nama_univ}')


class Fakultas(Univ):

    def __init__(self, nama_univ, nama_fakultas, nama_prodi):
        super().__init__(nama_univ)
```

```
        self.nama_fakultas = nama_fakultas
        self.nama_prodi = nama_prodi

    def desc(self):
        super().desc()
        print(f'Fakultas\t: {self.nama_fakultas}')
        print(f'Program studi\t: {self.nama_prodi}')

class Gedung(Univ):

    def __init__(self, nama_gedung):
        self.nama_gedung = nama_gedung

    def desc(self):
        super().desc()
        print(f'Gedung\t\t: {self.nama_gedung}')


class Mahasiswa(Gedung, Fakultas):

    def __init__(self, nama_mhsw, nim, nama_univ, nama_fakultas,
nama_prodi, nama_gedung):
        Fakultas.__init__(self, nama_univ, nama_fakultas, nama_prodi)
        Gedung.__init__(self, nama_gedung)
        self.nama_mhsw = nama_mhsw
        self.nim = nim

    def desc(self):
        super().desc()
        print(f"Nama\t\t: {self.nama_mhsw}")
        print(f"Nim\t\t: {self.nim}")


anto = Mahasiswa('Antonio Diegory', '210511111', 'Universitas Terbuka',
'Teknik', 'Teknik Informatika', 'Machdor')
anto.desc()
```

Output: