

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

M.Muchibbulloh

R2(B) 210511078

Tugas Praktikum 4 :

1. Buatlah Composition yang terdiri dari Peneliti dan Jurnal
2. Buatlah Composition yang terdiri dari Mahasiswa dan Kelompok KKM
3. Buatlah Composition yang terdiri dari Penulis dan Buku

Jawaban

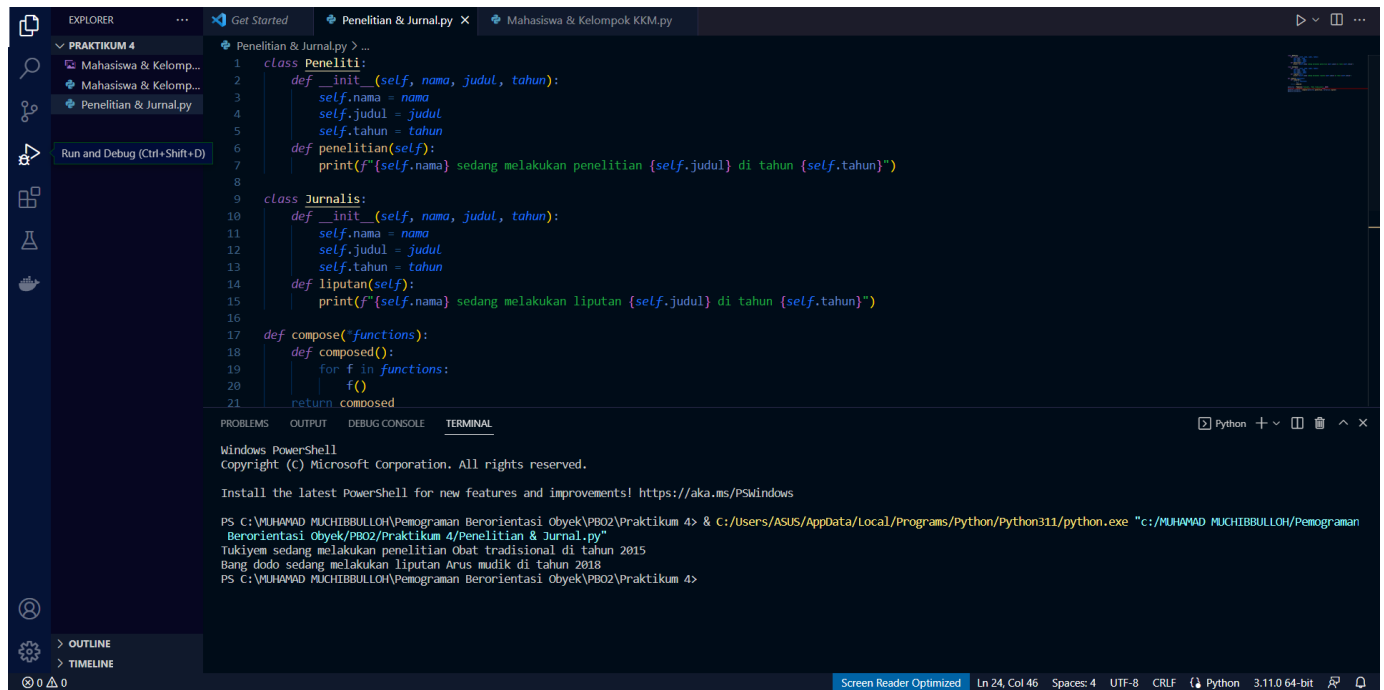
1. Penelitian dan jurnal

```
class Peneliti:
    def __init__(self, nama, judul, tahun):
        self.nama = nama
        self.judul = judul
        self.tahun = tahun
    def penelitian(self):
        print(f"{self.nama} sedang melakukan penelitian
{self.judul} di tahun {self.tahun}")
class Jurnalis:
    def __init__(self, nama, judul, tahun):
        self.nama = nama
        self.judul = judul
        self.tahun = tahun
    def liputan(self):
        print(f"{self.nama} sedang melakukan liputan
{self.judul} di tahun {self.tahun}")

def compose(*functions):
    def composed():
        for f in functions:
            f()
    return composed

peneliti1 = Peneliti("Tukime", "Obat tradisional", 2015)
jurnalis1 = Jurnalis("Bang dodo", "Arus mudik", 2018)
peneliti_jurnalis = compose(peneliti1.penelitian,
jurnalis1.liputan)
peneliti_jurnalis()
```

Output



The screenshot shows a VS Code editor with a Python file named 'Penelitian & Jurnal.py'. The code defines two classes: 'Peneliti' and 'Jurnal'. The 'Peneliti' class has methods for initialization, attribute assignment, and a 'penelitian' method that prints a message. The 'Jurnal' class has methods for initialization, attribute assignment, and a 'liputan' method that prints a message. A 'compose' function is also defined. The terminal output shows the execution of the script, displaying the messages from the 'penelitian' and 'liputan' methods.

```
1 class Peneliti:
2     def __init__(self, nama, judul, tahun):
3         self.nama = nama
4         self.judul = judul
5         self.tahun = tahun
6     def penelitian(self):
7         print(f"{self.nama} sedang melakukan penelitian {self.judul} di tahun {self.tahun}")
8
9 class Jurnal:
10     def __init__(self, nama, judul, tahun):
11         self.nama = nama
12         self.judul = judul
13         self.tahun = tahun
14     def liputan(self):
15         print(f"{self.nama} sedang melakukan liputan {self.judul} di tahun {self.tahun}")
16
17 def compose(functions):
18     def composed():
19         for f in functions:
20             f()
21     return composed
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\MUHAMMAD MUCHIBULLOH\Pemograman Berorientasi Obyek\PB02\Praktikum 4> & C:/Users/ASUS/AppData/Local/Programs/Python/python311/python.exe "c:/MUHAMMAD MUCHIBULLOH/Pemograman Berorientasi Obyek/PB02/Praktikum 4/Penelitian & Jurnal.py"

Tukiyem sedang melakukan penelitian Obat tradisional di tahun 2015
Bang dodo sedang melakukan liputan Arus mudik di tahun 2018
PS C:\MUHAMMAD MUCHIBULLOH\Pemograman Berorientasi Obyek\PB02\Praktikum 4>

2. Mahasiswa dan kelompok kkm

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

    def info(self):
        print(f>Nama: {self.nama}")
        print(f"NIM: {self.nim}")

class Kelompok:
    def __init__(self, nomor, anggota=[]):
        self.nomor = nomor
        self.anggota = anggota

    def tambah_anggota(self, mhs):
        self.anggota.append(mhs)
```

```

def info(self):
    print(f"Kelompok {self.nomor}")
    print("Daftar Anggota:")
    for mhs in self.anggota:
        mhs.info()
    print()

mhs1 = Mahasiswa("Muchib", "210511078")
mhs2 = Mahasiswa("Aulia", "230511003")
mhs3 = Mahasiswa("Mawang", "210511069")
mhs4 = Mahasiswa("Galih", "210511056")
kel1 = Kelompok("KKM 1")
kel1.tambah_anggota(mhs1)
kel1.tambah_anggota(mhs2)
kel2 = Kelompok("KKM 2")
kel2.tambah_anggota(mhs3)
kel2.tambah_anggota(mhs4)
kel1.info()
kel2.info()

```

Output

The screenshot shows a VS Code editor with a file named 'Mahasiswa & Kelompok KKM.py'. The code defines two classes: `Mahasiswa` and `Kelompok`. The `Mahasiswa` class has an `__init__` method that takes `nama` and `nim`, and an `info` method that prints the name and NIM. The `Kelompok` class has an `__init__` method that takes `nomor` and a list of `anggota`, a `tambah_anggota` method that appends a `Mahasiswa` object to the `anggota` list, and an `info` method that prints the group number and the list of members.

The terminal output shows the execution of the code. It first prints the information for the 'KKM 1' group, which includes members Muchib (210511078) and Aulia (230511003). Then it prints the information for the 'KKM 2' group, which includes members Mawang (210511069) and Galih (210511056).

```

class Mahasiswa:
    def __init__(self, nama, nim):
        self.nama = nama
        self.nim = nim

    def info(self):
        print(f>Nama: {self.nama}")
        print(f"NIM: {self.nim}")

class Kelompok:
    def __init__(self, nomor, anggota=[]):
        self.nomor = nomor
        self.anggota = anggota

    def tambah_anggota(self, mhs):
        self.anggota.append(mhs)

    def info(self):
        print(f"Kelompok {self.nomor}")
        print("Daftar Anggota:")

```

```

NIM: 210511069
Nama: Galih
NIM: 210511056

Kelompok KKM 2
Daftar Anggota:
Nama: Muchib
NIM: 210511078
Nama: Aulia
NIM: 230511003
Nama: Mawang
NIM: 210511069
Nama: Galih
NIM: 210511056

```

PS C:\MUHAMMAD MUCHIBBULLOH\Pemrograman Berorientasi Obyek\PB02\Praktikum 4>

3. Penulis dan buku

```
class Penulis:
    def __init__(self, nama, alamat):
        self.nama = nama
        self.alamat = alamat
        self.buku = []

    def tambah_buku(self, judul):
        self.buku.append(Buku(judul))

    def info(self):
        print(f>Nama Penulis: {self.nama}")
        print(f>Alamat: {self.alamat}")
        print("Buku yang ditulis:")
        for buku in self.buku:
            print("- " + buku.judul)
        print()

class Buku:
    def __init__(self, judul):
        self.judul = judul

    def info(self):
        print(f>Judul Buku: {self.judul}")

penulis1 = Penulis("Raditya dika", "Indonesia")
penulis2 = Penulis("Panji pragiwaksono", "Indonesia")
penulis3 = Penulis("Ernest", "Indonesia")

penulis1.tambah_buku("Kambing jantan")
penulis1.tambah_buku("Manusia setengah salmon")
penulis1.tambah_buku("Ubur ubur lembur")
```

```

penulis2.tambah_buku("Menemukan indonesia")
penulis2.tambah_buku("Nasional its me")
penulis3.tambah_buku("Cek took sebelah")

penulis1.info()
penulis2.info()
penulis3.info()

```

Output

The screenshot shows a VS Code editor with a file named 'Penulis dan buku.py'. The code defines two classes: 'Penulis' and 'Buku'. The 'Penulis' class has methods for adding books and displaying author information. The 'Buku' class is a simple book object. The terminal output shows the execution of the code, displaying the author's name, address, and a list of books.

```

class Penulis:
    def __init__(self, nama, alamat):
        self.nama = nama
        self.alamat = alamat
        self.buku = []

    def tambah_buku(self, judul):
        self.buku.append(Buku(judul))

    def info(self):
        print(f>Nama Penulis: {self.nama}")
        print(f>Alamat: {self.alamat}")
        print("Buku yang ditulis:")
        for buku in self.buku:
            print("- " + buku.judul)
        print()

class Buku:
    def __init__(self, judul):
        self.judul = judul

```

Terminal Output:

```

- Kambing jantan
- Manusia setengah salmon
- Ubur ubur lembur

Nama Penulis: Panji Pragiwaksono
Alamat: Indonesia
Buku yang ditulis:
- Menemukan indonesia
- Nasional its me

Nama Penulis: Ernest
Alamat: Indonesia
Buku yang ditulis:
- Cek toko sebelah

```

Link github< https://github.com/Muchibbulloh/PBO_2 >