

Modul Pemrograman Berorientasi Objek (PBO) dengan Python

1. Pengenalan OOP

Pemrograman Berorientasi Objek (OOP) adalah paradigma pemrograman yang menggunakan 'objek' - data yang memiliki atribut dan fungsi. Python mendukung OOP dengan konsep seperti kelas (class), objek (object), pewarisan (inheritance), enkapsulasi (encapsulation), dan polimorfisme (polymorphism).

2. Kelas dan Objek

Kelas adalah cetakan atau blueprint untuk membuat objek. Objek adalah instance dari kelas.

Contoh:

```
class Mahasiswa:
```

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

```
        self.nama = nama
```

```
        self.nim = nim
```

```
    def tampilkan_info(self):
```

```
        print(f>Nama: {self.nama}, NIM: {self.nim}")
```

```
mhs1 = Mahasiswa("Budi", "123456")
```

```
mhs1.tampilkan_info()
```

3. Enkapsulasi

Enkapsulasi adalah menyembunyikan data agar tidak dapat diakses langsung dari luar kelas.

Modul Pemrograman Berorientasi Objek (PBO) dengan Python

Contoh:

```
class Mahasiswa:

    def __init__(self, nama):

        self.__nama = nama

    def get_nama(self):

        return self.__nama

mhs = Mahasiswa("Ani")

print(mhs.get_nama())
```

4. Pewarisan

Pewarisan memungkinkan sebuah kelas mewarisi atribut dan metode dari kelas lain.

Contoh:

```
class Kendaraan:

    def __init__(self, merk):

        self.merk = merk

class Mobil(Kendaraan):

    def klakson(self):
```

Modul Pemrograman Berorientasi Objek (PBO) dengan Python

```
print(f"{self.merk} berbunyi: Beep!")
```

```
m = Mobil("Toyota")
```

```
m.klakson()
```

5. Polimorfisme

Polimorfisme memungkinkan metode dengan nama sama berperilaku berbeda tergantung pada objeknya.

Contoh:

```
class Burung:
```

```
    def bersuara(self):
```

```
        print("Burung bersuara")
```

```
class Elang(Burung):
```

```
    def bersuara(self):
```

```
        print("Elang: screech!")
```

```
class Merpati(Burung):
```

```
    def bersuara(self):
```

```
        print("Merpati: kukuruyuk!")
```

```
hewan = [Elang(), Merpati()]
```

```
for b in hewan:
```

Modul Pemrograman Berorientasi Objek (PBO) dengan Python

```
b.bersuara()
```

6. Praktik OOP Sederhana

Buat program perpustakaan sederhana yang memungkinkan menambah dan menampilkan buku.

```
class Buku:
```

```
    def __init__(self, judul, pengarang):
```

```
        self.judul = judul
```

```
        self.pengarang = pengarang
```

```
    def info(self):
```

```
        print(f"{self.judul} oleh {self.pengarang}")
```

```
daftar_buku = [
```

```
    Buku("Python Dasar", "Dina"),
```

```
    Buku("OOP Python", "Rudi")
```

```
]
```

```
for buku in daftar_buku:
```

```
    buku.info()
```