

**LAPORAN PRAKTIKUM PEMOGRAMAN
BERORIENTASI OBJEK (PBO)
PRAKTIKUM 10**



2411102441285

HAFIDZAL MUFTY

**FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI S1 TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH KALIMANTAN
TIMUR**

1. Sistem perpustakaan

```

1 # Definisi class Buku
2 class Buku:
3     def __init__(self, judul, penulis):
4         self.judul = judul
5         self.penulis = penulis
6         self.dipinjam = False
7
8     def tampilkan_info(self):
9         status = "Dipinjam" if self.dipinjam else "Tersedia"
10        return f"Buku: {self.judul}, Penulis: {self.penulis}, Status: {status}"
11
12 # Definisi class Pustaka
13 class Pustaka:
14     def __init__(self):
15         self.koleksi_buku = []
16
17     def tambah_buku(self, buku):
18         self.koleksi_buku.append(buku)
19
20     def tampilkan_koleksi(self):
21         for buku in self.koleksi_buku:
22             print(buku.tampilkan_info())
23

```

```

1 # Definisi class Anggota
2 class Anggota:
3     def __init__(self, nama):
4         self.nama = nama
5         self.daftar_buku = []
6
7     def pinjam_buku(self, buku):
8         if not buku.dipinjam:
9             buku.dipinjam = True
10            self.daftar_buku.append(buku)
11            print(f"{self.nama} meminjam buku '{buku.judul}'")
12        else:
13            print(f"Buku '{buku.judul}' sedang dipinjam orang lain.")
14
15     def kembalikan_buku(self, buku):
16         if buku in self.daftar_buku:
17             buku.dipinjam = False
18             self.daftar_buku.remove(buku)
19             print(f"{self.nama} mengembalikan buku '{buku.judul}'")
20         else:
21             print(f"{self.nama} tidak meminjam buku '{buku.judul}'")
22

```

```
● ● ●

1 # --- Membuat objek dan menjalankan simulasi ---
2 buku1 = Buku("Harry Potter", "J.K. Rowling")
3 buku2 = Buku("The Hobbit", "J.R.R. Tolkien")
4
5 pustaka = Pustaka()
6 pustaka.tambah_buku(buku1)
7 pustaka.tambah_buku(buku2)
8
9 anggota1 = Anggota("Andi")
10 anggota1.pinjam_buku(buku1)
11 anggota1.pinjam_buku(buku2)
12
13 print("\nStatus Koleksi Pustaka:")
14 pustaka.tampilan_koleksi()
15
16 anggota1.kembalikan_buku(buku1)
17 print("\nStatus Koleksi Setelah Pengembalian:")
18 pustaka.tampilan_koleksi()
```

Output :

```
PS C:\Users\Infinix\PBO> & C:/Users/Infinix/AppData/Local/Programs/PBO/praktikum 10/sistem-perpustakaan.py"
Andi meminjam buku 'Harry Potter'
Andi meminjam buku 'The Hobbit'

Status Koleksi Pustaka:
Buku: Harry Potter, Penulis: J.K. Rowling, Status: Dipinjam
Buku: The Hobbit, Penulis: J.R.R. Tolkien, Status: Dipinjam
Andi mengembalikan buku 'Harry Potter'

Status Koleksi Setelah Pengembalian:
Buku: Harry Potter, Penulis: J.K. Rowling, Status: Tersedia
Buku: The Hobbit, Penulis: J.R.R. Tolkien, Status: Dipinjam
PS C:\Users\Infinix\PBO>
```

2. Sistem restoran

```

● ● ●

1 # class menu
2 class Menu:
3     def __init__(self, nama, harga):
4         self._nama = nama
5         self._harga = harga
6
7     def info(self):
8         return f"{self._nama} - Rp{self._harga}"
9
10 # child class dari menu
11 class MenuMinuman(Menu):
12     def __init__(self, nama, harga, ukuran):
13         super().__init__(nama, harga)
14         self._ukuran = ukuran
15
16     def info(self): # overriding
17         return f"{self._nama} ({self._ukuran}) - Rp{self._harga}"

```

```

● ● ●

1 # class pesanan
2 class Pesanan:
3     def __init__(self, pelanggan):
4         self.pelanggan = pelanggan
5         self.menu_dipesan = []
6         self.status = "Belum diproses"
7
8     def tambah_menu(self, menu):
9         self.menu_dipesan.append(menu)
10    print(f"{self.pelanggan.nama} memesan {menu._nama}")
11
12    def ubah_status(self, status):
13        self.status = status
14        print(f"Status pesanan {self.pelanggan.nama}: {self.status}")
15
16    def tampil(self):
17        print(f"\nPesanannya {self.pelanggan.nama}:")
18        for m in self.menu_dipesan:
19            print("-", m.info())
20        total = sum(m._harga for m in self.menu_dipesan)
21        print(f"Total: Rp{total}")
22        print(f"Status: {self.status}")

```

```
● ○ ●  
1 # class pelanggan  
2 class Pelanggan:  
3     def __init__(self, nama):  
4         self.nama = nama  
5         self.pesanan = Pesanan(self)  
6  
7     def pesan(self, menu):  
8         self.pesanan.tambah_menu(menu)  
9  
10    def konfirmasi(self):  
11        self.pesanan.ubah_status("Dikonfirmasi")  
12  
13    def lihat(self):  
14        self.pesanan.tampil()
```

```
● ○ ●  
1 # Simulasi  
2 menu1 = Menu("Nasi Goreng", 20000)  
3 minuman = MenuMinuman("Es Teh", 5000, "Besar")  
4  
5 andi = Pelanggan("Andi")  
6 andi.pesan(menu1)  
7 andi.pesan(minuman)  
8 andi.konfirmasi()  
9 andi.lihat()
```

Output :

```
PS C:\Users\Infinix\PBO> & C:/Users/Infinix/AppData/Local/  
/PBO/praktikum 10/sistem-restoran.py"  
Andi memesan Nasi Goreng  
Andi memesan Es Teh  
Status pesanan Andi: Dikonfirmasi  
  
Pesanan Andi:  
- Nasi Goreng - Rp20000  
- Es Teh (Besar) - Rp5000  
Total: Rp25000  
Status: Dikonfirmasi  
PS C:\Users\Infinix\PBO> █
```

Program ini merupakan simulasi sistem restoran sederhana menggunakan konsep OOP. Class **Menu** menyimpan data nama dan harga, sedangkan **MenuMinuman** mewarisi **Menu** dan menambahkan atribut ukuran dengan method **overriding**. Class **Pesanan** mengatur daftar **menu**, **status**, dan **total harga**, sementara **Pelanggan** merepresentasikan **pelanggan** yang dapat memesan, mengonfirmasi, dan melihat pesanan. Melalui interaksi antar objek, program ini menerapkan **enkapsulasi, inheritance, dan polymorphism**.