

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 10



2411102441052

Angga Maulana Saputra

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

Latihan Praktikum 10:

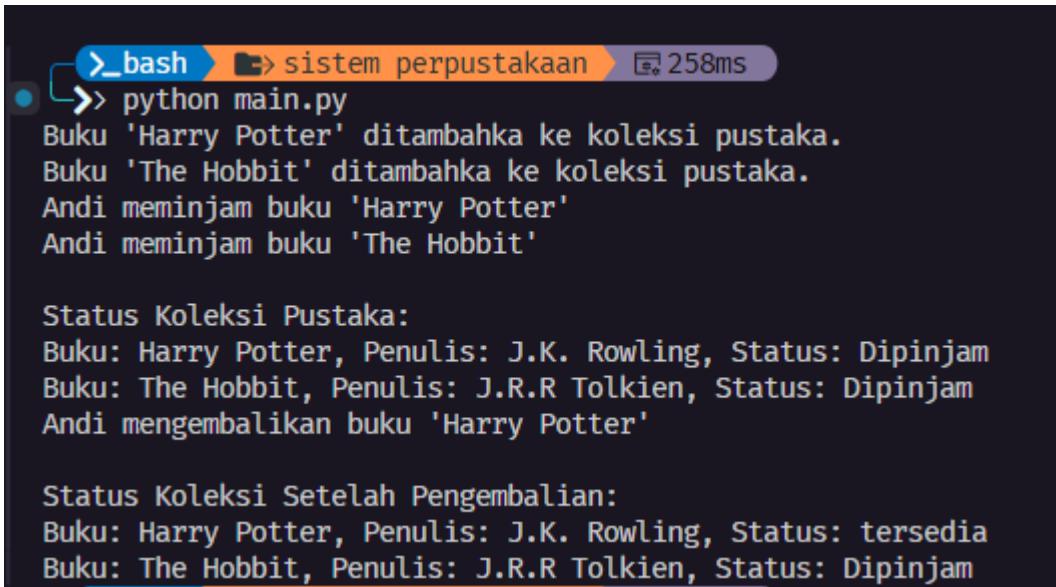
- a. Screenshot kode latihan, Sistem Perpustakaan/main.py



```
1 class Buku:
2     def __init__(self, judul, penulis):
3         self.judul = judul
4         self.penulis = penulis
5         self.dipinjam = False
6
7     def tampilkan_info(self):
8         status = "Dipinjam" if self.dipinjam else "tersedia"
9         return f"Buku: {self.judul}, Penulis: {self.penulis}, Status: {status}"
10
11 class Anggota:
12     def __init__(self, nama):
13         self.nama = nama
14         self.daftar_buku = []
15
16     def pinjam_buku(self, buku):
17         if not buku.dipinjam:
18             buku.dipinjam = True
19             self.daftar_buku.append(buku)
20             print(f"{self.nama} meminjam buku '{buku.judul}'")
21         else:
22             print(f"Buku '{buku.judul}' sedang dipinjam orang lain.")
23
24     def kembalikan_buku(self, buku):
25         if buku in self.daftar_buku:
26             buku.dipinjam = False
27             self.daftar_buku.remove(buku)
28             print(f"{self.nama} mengembalikan buku '{buku.judul}'")
29         else:
30             print(f"{self.nama} tidak meminjam buku '{buku.judul}'")
```

```
● ○ ●
1 class Pustaka:
2     def __init__(self):
3         self.koleksi_buku = []
4
5     def tambah_buku(self, buku):
6         self.koleksi_buku.append(buku)
7         print(f"Buku '{buku.judul}' ditambahka ke koleksi pustaka.")
8
9     def tampilkan_koleksi(self):
10        for buku in self.koleksi_buku:
11            print(buku.tampilkan_info())
12
13 # --- Membuat objek dan menjalankan simulasi ---
14 buku1 = Buku("Harry Potter", "J.K. Rowling")
15 buku2 = Buku("The Hobbit", "J.R.R Tolkien")
16
17 pustaka = Pustaka()
18 pustaka.tambah_buku(buku1)
19 pustaka.tambah_buku(buku2)
20
21 anggota1 = Anggota("Andi")
22 anggota1.pinjam_buku(buku1)
23 anggota1.pinjam_buku(buku2)
24
25 print("\nStatus Koleksi Pustaka:")
26 pustaka.tampilkan_koleksi()
27
28 anggota1.kembalikan_buku(buku1)
29 print("\nStatus Koleksi Setelah Pengembalian:")
30 pustaka.tampilkan_koleksi()
```

b. Screenshot hasil eksekusi program



```
>_bash ➤ sistem_perpustakaan ➤ 258ms
>> python main.py
Buku 'Harry Potter' ditambahka ke koleksi pustaka.
Buku 'The Hobbit' ditambahka ke koleksi pustaka.
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
```

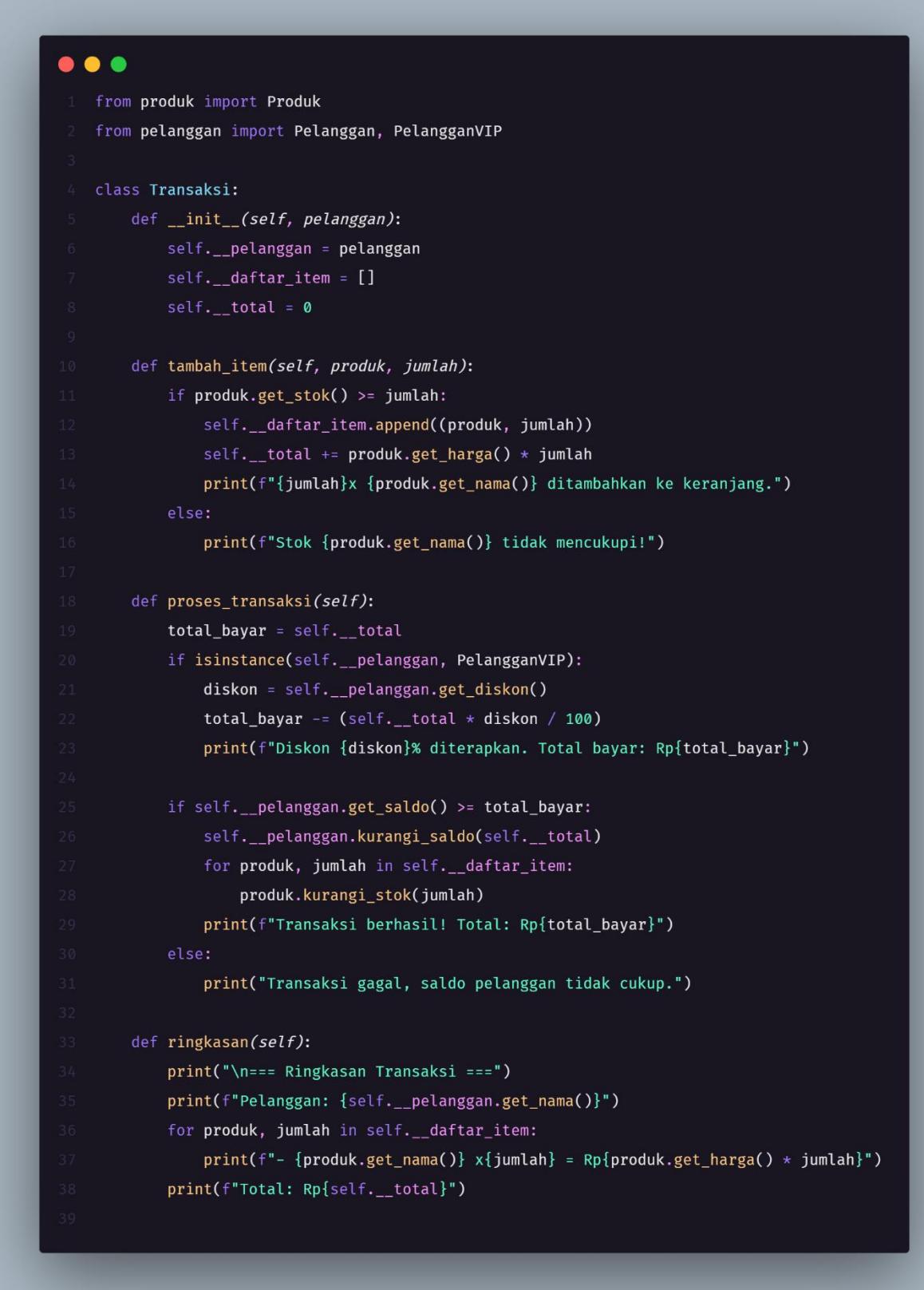
c. Screenshot kode Toko/pelanggan.py

```
● ● ●
1  class Pelanggan:
2      def __init__(self, nama, saldo):
3          self.__nama = nama
4          self.__saldo = saldo
5
6      def get_nama(self):
7          return self.__nama
8
9      def get_saldo(self):
10         return self.__saldo
11
12     def set_saldo(self, saldo_baru):
13         if saldo_baru >= 0:
14             self.__saldo = saldo_baru
15         else:
16             print("Saldo tidak boleh negatif!")
17
18     def kurangi_saldo(self, jumlah):
19         if jumlah <= self.__saldo:
20             self.__saldo -= jumlah
21             print(f"{self.__nama} membayar sebesar Rp{jumlah}")
22         else:
23             print("Saldo tidak cukup!")
24
25     def __str__(self):
26         return f"Pelanggan: {self.__nama}, Saldo: Rp{self.__saldo}"
27
28 class PelangganVIP(Pelanggan):
29     def __init__(self, nama, saldo, diskon):
30         super().__init__(nama, saldo)
31         self.__diskon = diskon
32
33     def get_diskon(self):
34         return self.__diskon
35
36     def kurangi_saldo(self, jumlah):
37         jumlah_diskon = jumlah - (jumlah * self.__diskon / 100)
38         if jumlah_diskon <= self.get_saldo():
39             self.set_saldo(self.get_saldo() - jumlah_diskon)
40             print(f"{self.get_nama()} (VIP) membayar Rp{jumlah_diskon} dengan diskon {self.__diskon}%")
41         else:
42             print("Saldo tidak cukup!")
43
44     def __str__(self):
45         return f"Pelanggan VIP: {self.get_nama()}, Saldo: Rp{self.get_saldo()}, Diskon: {self.__diskon}%"
46
```

d. Screenshot kode Toko/produk.py

```
1  class Produk:
2      def __init__(self, nama, harga, stok):
3          self.__nama = nama
4          self.__harga = harga
5          self.__stok = stok
6
7      def get_nama(self):
8          return self.__nama
9
10     def get_harga(self):
11         return self.__harga
12
13     def get_stok(self):
14         return self.__stok
15
16     def kurangi_stok(self, jumlah):
17         if jumlah <= self.__stok:
18             self.__stok -= jumlah
19             print(f"Stok {self.__nama} berkurang {jumlah} unit.")
20         else:
21             print(f"Stok {self.__nama} tidak cukup!")
22
23     def __str__(self):
24         return f"Produk: {self.__nama}, Harga: Rp{self.__harga}, Stok: {self.__stok}"
25
```

e. Screenshot kode Toko/transaksi.py



```
● ● ●

1  from produk import Produk
2  from pelanggan import Pelanggan, PelangganVIP
3
4  class Transaksi:
5      def __init__(self, pelanggan):
6          self.__pelanggan = pelanggan
7          self.__daftar_item = []
8          self.__total = 0
9
10     def tambah_item(self, produk, jumlah):
11         if produk.get_stok() >= jumlah:
12             self.__daftar_item.append((produk, jumlah))
13             self.__total += produk.get_harga() * jumlah
14             print(f"{jumlah}x {produk.get_nama()} ditambahkan ke keranjang.")
15         else:
16             print(f"Stok {produk.get_nama()} tidak mencukupi!")
17
18     def proses_transaksi(self):
19         total_bayar = self.__total
20         if isinstance(self.__pelanggan, PelangganVIP):
21             diskon = self.__pelanggan.get_diskon()
22             total_bayar -= (self.__total * diskon / 100)
23             print(f"Diskon {diskon}% diterapkan. Total bayar: Rp{total_bayar}")
24
25         if self.__pelanggan.get_saldo() >= total_bayar:
26             self.__pelanggan.kurangi_saldo(self.__total)
27             for produk, jumlah in self.__daftar_item:
28                 produk.kurangi_stok(jumlah)
29             print(f"Transaksi berhasil! Total: Rp{total_bayar}")
30         else:
31             print("Transaksi gagal, saldo pelanggan tidak cukup.")
32
33     def ringkasan(self):
34         print("\n== Ringkasan Transaksi ==")
35         print(f"Pelanggan: {self.__pelanggan.get_nama()}")
36         for produk, jumlah in self.__daftar_item:
37             print(f"- {produk.get_nama()} x{jumlah} = Rp{produk.get_harga() * jumlah}")
38         print(f"Total: Rp{self.__total}")
39
```

f. Screenshot kode Toko/main.py.

```
 1  from pelanggan import Pelanggan, PelangganVIP
 2  from produk import Produk
 3  from transaksi import Transaksi
 4
 5  pelanggan1 = Pelanggan("Bas", 200000)
 6  pelanggan2 = PelangganVIP("Kara", 300000, 10)
 7
 8  produk1 = Produk("Laptop", 150000, 5)
 9  produk2 = Produk("Mouse", 50000, 10)
10
11 print(pelanggan1)
12 print(pelanggan2)
13 print(produk1)
14 print(produk2)
15 print()
16
17 print("== Transaksi Pelanggan Biasa ==")
18 transaksi1 = Transaksi(pelanggan1)
19 transaksi1.tambah_item(produk1, 1)
20 transaksi1.ringkasan()
21 transaksi1.proses_transaksi()
22
23 print("\n== Transaksi Pelanggan VIP ==")
24 transaksi2 = Transaksi(pelanggan2)
25 transaksi2.tambah_item(produk2, 2)
26 transaksi2.ringkasan()
27 transaksi2.proses_transaksi()
28
29 print("\n== Setelah Transaksi ==")
30 print(pelanggan1)
31 print(pelanggan2)
32
```

g. Screenshot hasil eksekusi program.

```
>_bash ➜ toko ➜ 302ms
● ➤ python main.py
Pelanggan: Bas, Saldo: Rp200000
Pelanggan VIP: Kara, Saldo: Rp300000, Diskon: 10%
Produk: Laptop, Harga: Rp150000, Stok: 5
Produk: Mouse, Harga: Rp50000, Stok: 10

==== Transaksi Pelanggan Biasa ====
1x Laptop ditambahkan ke keranjang.

==== Ringkasan Transaksi ====
Pelanggan: Bas
- Laptop x1 = Rp150000
Total: Rp150000
Bas membayar sebesar Rp150000
Stok Laptop berkurang 1 unit.
Transaksi berhasil! Total: Rp150000

==== Transaksi Pelanggan VIP ====
2x Mouse ditambahkan ke keranjang.

==== Ringkasan Transaksi ====
Pelanggan: Kara
- Mouse x2 = Rp100000
Total: Rp100000
Diskon 10% diterapkan. Total bayar: Rp90000.0
Kara (VIP) membayar Rp90000.0 dengan diskon 10%
Stok Mouse berkurang 2 unit.
Transaksi berhasil! Total: Rp90000.0

==== Setelah Transaksi ====
Pelanggan: Bas, Saldo: Rp50000
Pelanggan VIP: Kara, Saldo: Rp210000.0, Diskon: 10%
```