

Contoh Kasus 2

Berikut adalah contoh kasus sederhana pembelian barang dengan menggunakan dua class, yaitu `Barang` dan `Pembelian` dalam Python.

```
# Class Barang
class Barang:
    def __init__(self, nama, harga, stok):
        self.nama = nama
        self.harga = harga
        self.stok = stok

    def kurangi_stok(self, jumlah):
        if jumlah <= self.stok:
            self.stok -= jumlah
        else:
            print(f"Stok {self.nama} tidak cukup! Tersisa {self.stok}.")

    def __str__(self):
        return f"{self.nama} - Harga: {self.harga}, Stok: {self.stok}"

# Class Pembelian
class Pembelian:
    def __init__(self):
        self.daftar_pembelian = []

    def tambah_barang(self, barang, jumlah):
        if barang.stok >= jumlah:
            self.daftar_pembelian.append((barang, jumlah))
            barang.kurangi_stok(jumlah)
        else:
            print(f"Pembelian gagal! Stok {barang.nama} tidak mencukupi.")

    def total_harga(self):
        total = 0
        for barang, jumlah in self.daftar_pembelian:
            total += barang.harga * jumlah
        return total

    def tampilkan_pembelian(self):
        print("Daftar Pembelian:")
        for barang, jumlah in self.daftar_pembelian:
            print(f"{barang.nama} - Jumlah: {jumlah}, Harga Total: {barang.harga * jumlah}")
        print(f"Total Harga: {self.total_harga()}")

# Contoh penggunaan
if __name__ == "__main__":
    # Membuat objek barang
```

```

barang1 = Barang("Laptop", 10000000, 5)
barang2 = Barang("Mouse", 100000, 10)

# Menampilkan stok awal
print(barang1)
print(barang2)

# Membuat objek pembelian
pembelian = Pembelian()

# Menambah barang ke pembelian
pembelian.tambah_barang(barang1, 2)
pembelian.tambah_barang(barang2, 3)

# Menampilkan hasil pembelian
pembelian.tampilkan_pembelian()

# Menampilkan stok barang setelah pembelian
print("\nStok setelah pembelian:")
print(barang1)
print(barang2)

```

Penjelasan:

1. **Class Barang**: Mewakili barang yang akan dibeli. Memiliki atribut `nama`, `harga`, dan `stok`. Metode `kurangi_stok` digunakan untuk mengurangi stok saat pembelian terjadi.
2. **Class Pembelian**: Mewakili proses pembelian barang. Atribut `daftar_pembelian` menyimpan daftar barang yang dibeli dan jumlahnya. Metode `tambah_barang` digunakan untuk menambahkan barang ke pembelian, dan metode `tampilkan_pembelian` untuk menampilkan detail pembelian.

Output contoh:

```

Laptop - Harga: 10000000, Stok: 5
Mouse - Harga: 100000, Stok: 10
Daftar Pembelian:
Laptop - Jumlah: 2, Harga Total: 20000000
Mouse - Jumlah: 3, Harga Total: 300000
Total Harga: 20300000

Stok setelah pembelian:
Laptop - Harga: 10000000, Stok: 3
Mouse - Harga: 100000, Stok: 7

```