

LAPORAN POYEK UAS

PEMROGAMAN 2

Dosen Pengampu: Tri Hadiyah Muliawati S.S., M.Kom.



Oleh:

Bayu Kurniawan
(3322600019)

PROGRAM STUDI D4 SAINS DATA TERAPAN
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2023

```
class Laptop:
    def __init__(self, brand, model, price):
        self.brand = brand
        self.model = model
        self.price = price

    # Method untuk menampilkan detail laptop
    def show(self):
        print(f"Brand: {self.brand}")
        print(f"Model: {self.model}")
        print(f"Price: {self.price}")
```

Class **Laptop** adalah class yang merepresentasikan objek laptop dengan atribut **brand**, **model**, dan **price**. Class ini memiliki method **__init__** untuk menginisialisasi atribut-atribut tersebut, dan method **show** untuk menampilkan detail laptop.

```
class LaptopStore:
    def __init__(self, name):
        self.name = name
        self.__laptops = {} # untuk menyimpan laptop berdasarkan model

    # Method untuk menambahkan laptop ke toko
    def add_laptop(self, laptop):
        # Periksa apakah model laptop sudah ada di toko
        if laptop.model in self.__laptops:
            self.__laptops[laptop.model]["stock"] += 1
        else:
            # Jika tidak ada, tambahkan data baru ke dictionary dengan stok sebagai satu
            self.__laptops[laptop.model] = {"laptop": laptop, "stock": 1}

    # Method untuk menampilkan apa yang dimiliki toko
    def display_laptops(self):
        print(f"{self.name} memiliki laptop berikut:")
        for model in self.__laptops:
            laptop = self.__laptops[model]["laptop"]
            stock = self.__laptops[model]["stock"]
            print(f"{model}: {stock} in stock")
            laptop.show()
            print()
```

Class **LaptopStore** adalah kelas yang merepresentasikan objek toko laptop dengan atribut **name** dan **__laptops**. Atribut **__laptops** adalah dictionary yang menyimpan laptop berdasarkan modelnya, dengan nilai berupa dictionary lain yang berisi objek laptop dan stoknya. Class ini memiliki method **__init__** untuk menginisialisasi atribut-atribut tersebut, dan beberapa method lain untuk melakukan operasi-operasi terkait toko laptop.

- Method **add_laptop** untuk menambahkan laptop ke toko. Method ini akan memeriksa apakah model laptop sudah ada di toko atau tidak, dan menambahkan data baru atau menambahkan stok sesuai dengan kondisinya.
- Method **display_laptops** untuk menampilkan apa yang dimiliki toko. Method ini akan mengulangi semua model laptop di dictionary **__laptops**, dan menampilkan model, stok, dan detail laptopnya menggunakan metode **show** dari class **Laptop**.

```

# Method untuk mencari laptop yang dijual oleh toko dengan memasukkan nama laptop
def search_laptop(self, name):
    if name in self.__laptops:
        laptop = self.__laptops[name]["laptop"]
        stock = self.__laptops[name]["stock"]
        print(f"Kita punya {name} di toko kami:")
        print(f"{name}: {stock} in stock")
        laptop.show()
    else:
        print(f"\nMaaf, kami tidak punya {name} di toko kami.")

# Method untuk menampilkan jumlah cek pembelian
def display_total(self, models):
    total = 0
    for model in models:
        # Periksa apakah model tersebut ada di toko dan memiliki stok yang cukup
        if model in self.__laptops and self.__laptops[model]["stock"] > 0:
            price = self.__laptops[model]["laptop"].price
            total += price
            self.__laptops[model]["stock"] -= 1
        else:
            print(f"\nMaaf, kami tidak punya {model} atau kehabisan stok.")
    print(f"\nJumlah yang harus dibayar adalah {total}.")

```

- Method **search_laptop** untuk mencari laptop yang dijual oleh toko dengan memasukkan nama laptop. Method ini akan memeriksa apakah nama laptop ada di dictionary **__laptops** atau tidak, dan menampilkan informasi terkait atau pesan kesalahan sesuai dengan kondisinya.
- Method **display_total** untuk menampilkan jumlah cek pembelian. Method ini akan menerima daftar model laptop yang ingin dibeli oleh pelanggan, dan menghitung total harga dari semua laptop yang tersedia dan memiliki stok yang cukup. Method ini juga akan mengurangi stok dari setiap laptop yang dibeli, dan mencetak pesan kesalahan jika ada model laptop yang tidak ada atau kehabisan stok.

```

# Buat beberapa objek
laptop1 = Laptop("Dell", "XPS 13", 1000)
laptop2 = Laptop("Apple", "MacBook Air", 1200)
laptop3 = Laptop("Lenovo", "ThinkPad X1", 900)

# Buat objek toko laptop dan tambahkan beberapa laptop ke dalamnya
store = LaptopStore("Laptop World")
store.add_laptop(laptop1)
store.add_laptop(laptop2)
store.add_laptop(laptop3)
store.add_laptop(laptop2)

# Tampilkan apa yang dijual toko
store.display_laptops()

# Cari laptop yang dijual oleh toko
store.search_laptop("MacBook Air")

# Tampilkan jumlah cek pembelian
store.display_total(["XPS 13", "MacBook Air"])

```

```
Laptop World memiliki laptop berikut:
```

```
XPS 13: 1 in stock
```

```
Brand: Dell
```

```
Model: XPS 13
```

```
Price: 1000
```

```
MacBook Air: 2 in stock
```

```
Brand: Apple
```

```
Model: MacBook Air
```

```
Price: 1200
```

```
ThinkPad X1: 1 in stock
```

```
Brand: Lenovo
```

```
Model: ThinkPad X1
```

```
Price: 900
```

```
Kita punya MacBook Air di toko kami:
```

```
MacBook Air: 2 in stock
```

```
Brand: Apple
```

```
Model: MacBook Air
```

```
Price: 1200
```

```
Jumlah yang harus dibayar adalah 2200.
```

Di akhir kode, membuat beberapa objek dari class Laptop dengan brand, model, dan harga yang berbeda-beda. Serta membuat satu objek dari class LaptopStore dengan nama "Laptop World", dan menambahkan beberapa laptop ke dalamnya menggunakan method **add_laptop**. Kemudian menampilkan apa yang dijual oleh toko menggunakan method **display_laptops**, mencari laptop tertentu menggunakan method **search_laptop**, dan menampilkan jumlah cek pembelian menggunakan method **display_total**.

```

import re
class LaptopAppleStore(LaptopStore): # Membuat subclass dari LaptopStore
    def __init__(self, name):
        # Memanggil metode __init__ dari superclass
        super().__init__(name)

    def add_laptop(self, laptop):
        # Mengecek apakah laptop memiliki brand "Apple"
        regex = re.compile("Apple", re.IGNORECASE) # buat pola regex
        if re.match(regex, laptop.brand):
            super().add_laptop(laptop)
        else:
            print(f"Maaf, kami hanya menjual laptop dengan brand Apple.\n")

```

Class **LaptopAppleStore** adalah subclass yang mewarisi dari class **LaptopStore**. Subclass ini menggunakan method yang sama dari method superclassnya. Terdapat method **init** untuk memanggil method **init** dari superclass dan meneruskan parameter **name**. Method **add_laptop** memeriksa apakah parameter **laptop** memiliki brand yang cocok dengan "Apple" menggunakan ekspresi reguler. Jika iya, method **add_laptop** akan berjalan sesuai dengan method **add_laptop** dari superclass dan menambahkan laptop ke dictionary **__laptops**. Jika tidak, ia mencetak pesan kesalahan yang mengatakan toko hanya menjual laptop dengan merek Apple.

```

# Membuat beberapa laptop
laptop1 = Laptop("Dell", "XPS 13", 1000)
laptop2 = Laptop("Apple", "MacBook Air", 1200)
laptop3 = Laptop("Lenovo", "ThinkPad X1", 900)

# Membuat sebuah laptop apple store dan menambahkan beberapa laptop ke dalamnya
StoreApple = LaptopAppleStore("Laptop Apple World")
StoreApple.add_laptop(laptop1) # Ini akan mencetak pesan kesalahan
StoreApple.add_laptop(laptop2) # Ini akan berhasil
StoreApple.add_laptop(laptop3) # Ini akan mencetak pesan kesalahan
StoreApple.add_laptop(laptop2) # Ini akan berhasil

# Tampilkan apa yang dijual oleh toko
StoreApple.display_laptops()

# Cari laptop yang dijual oleh toko
StoreApple.search_laptop("MacBook Air")

# Tampilkan jumlah cek pembelian
StoreApple.display_total(["XPS 13", "MacBook Air"])

```

```
... Maaf, kami hanya menjual laptop dengan brand Apple.

Maaf, kami hanya menjual laptop dengan brand Apple.

Laptop Apple World memiliki laptop berikut:
MacBook Air: 2 in stock
Brand: Apple
Model: MacBook Air
Price: 1200

Kita punya MacBook Air di toko kami:
MacBook Air: 2 in stock
Brand: Apple
Model: MacBook Air
Price: 1200

Maaf, kami tidak punya XPS 13 atau kehabisan stok.

Jumlah yang harus dibayar adalah 1200.
```

Di akhir kode, membuat beberapa objek laptop dengan brand **brand**, **model**, dan **price**. yang berbeda, dan membuat objek toko laptop apple dengan nama "Laptop Apple World". Kemudian mencoba menambahkan beberapa laptop ke toko menggunakan metode **add_laptop**. Hanya laptop dengan merek Apple yang berhasil ditambahkan, sementara yang lain mencetak pesan kesalahan. Kemudian menampilkan laptop apa saja yang dijual oleh toko menggunakan method **display_laptops**, yang diwarisi dari superclass. Kemudian mencari laptop berdasarkan nama menggunakan method **search_laptop**, yang juga diwarisi dari superclass. Terakhir, menampilkan jumlah cek pembelian menggunakan method **display_total**, yang juga diwarisi dari superclass.