Praktikum Pemrograman II: Encapsulation

Dosen Pengampu Tri Hadiah Muliawati S.ST., M.Kom

In [ ]:

Wahyu Ikbal Maulana

3323600056

D3 SDT B

Politeknik Elektronika Negeri Surabaya

Latihan 1

In [ ]:

```
class Restaurant:
    def __init__(self, name, cuisine_type):
        self.name = name
        self.cuisine_type = cuisine_type

    def describe_restaurant(self):
        print(f"Restaurant name: {self.name}")
        print(f"Cuisine type: {self.cuisine_type}")

    def open_restaurant(self):
        print(f"The restaurant {self.name} is now open.")

# Membuat dua objek dari class Restaurant
restaurant1 = Restaurant("Pizza Hut", "Italian")
restaurant2 = Restaurant("Sushi King", "Japanese")

restaurant1.describe_restaurant()
restaurant2.describe_restaurant()
```

```
Restaurant name: Pizza Hut
Cuisine type: Italian
Restaurant name: Sushi King
Cuisine type: Japanese
```

Latihan 2

In [ ]:

```
class Restaurant:
    def __init__(self, name, address, cuisine_type):
        self.name = name
        self.address = address
        self.cuisine_type = cuisine_type
        self.menu = {}

    def describe_restaurant(self):
        print(f"Restaurant name: {self.name}")
        print(f"Address: {self.address}")
        print(f"Cuisine type: {self.cuisine_type}")

    def open_restaurant(self):
        print(f"The restaurant {self.name} is now open.")

    def display_menu(self):
        print(f"Menu at {self.name}:")
        if not self.menu:
            print("The menu is currently empty.")
        else:
            for menu_name, price in self.menu.items():
                print(f"- {menu_name}: ${price:.2f}")

# Penambahan menu
restaurant = Restaurant("Gourmet Place", "123 Flavor St.", "Italian")
restaurant.describe_restaurant()
restaurant.open_restaurant()
restaurant.display_menu()
restaurant.menu = {"Spaghetti Carbonara": 12.99, "Margherita Pizza": 9.99}
restaurant.display_menu()
```

```
Restaurant name: Gourmet Place
Address: 123 Flavor St.
Cuisine type: Italian
The restaurant Gourmet Place is now open.
Menu at Gourmet Place:
The menu is currently empty.
Menu at Gourmet Place:
- Spaghetti Carbonara: $12.99
- Margherita Pizza: $9.99
```

Latihan 3

In [ ]:

```
class Restaurant:
    def __init__(self, name, address, cuisine_type):
        self.name = name
        self.address = address
        self.cuisine_type = cuisine_type
        self.__menu = {}

    def describe_restaurant(self):
        print(f"Restaurant name: {self.name}")
        print(f"Address: {self.address}")
        print(f"Cuisine type: {self.cuisine_type}")

    def open_restaurant(self):
        print(f"The restaurant {self.name} is now open.")

    def add_menu_item(self, menu_name, price):
        self.__menu[menu_name] = price

    def remove_menu_item(self, menu_name):
        if menu_name in self.__menu:
            del self.__menu[menu_name]
        else:
            print(f"Menu item {menu_name} not found.")

    def display_menu(self):
        print(f"Menu at {self.name}:")
        if not self.__menu:
            print("The menu is currently empty.")
        else:
            for menu_name, price in self.__menu.items():
                print(f"- {menu_name}: ${price:.2f}")
```

Latihan 4

In [ ]:

```
class User:
    def __init__(self, first_name, last_name, age, umur):
        self.first_name = first_name
        self.last_name = last_name
        self.age = age
        self.umur = umur
```

```python
    def describe_user(self):
        print(f"User Information:")
        print(f"First Name: {self.first_name}")
        print(f"Last Name: {self.last_name}")
        print(f"Age: {self.age}")
        print(f"umur: {self.umur}")

    def greet_user(self):
        print(f"Hello, {self.first_name} {self.last_name}!")

# Membuat 2 objek dari class User
user1 = User("John", "Doe", 25, 18)
user2 = User("Jane", "Smith", 30, 25)

# Memanggil method describe_user() dan greet_user() untuk semua objek
user1.describe_user()
user2.describe_user()
```

```
User Information:
First Name: John
Last Name: Doe
Age: 25
umur: 18
User Information:
First Name: Jane
Last Name: Smith
Age: 30
umur: 25
```

In [ ]:

```python
user1.greet_user()
user2.greet_user()
```

```
Hello, John Doe!
Hello, Jane Smith!
```

Latihan 5

In [ ]:

```python
class User:
    def __init__(self):
        self.login_attempt = 0

    def increment_login_attempt(self):
        self.login_attempt += 1

    def reset_login_attempt(self):
        self.login_attempt = 0


# Uji coba
user = User()
print("Jumlah percobaan login awal:", user.login_attempt)

user.increment_login_attempt()
user.increment_login_attempt()
print("Jumlah percobaan login setelah 2 kali increment:", user.login_attempt)

user.reset_login_attempt()
print("Jumlah percobaan login setelah direset:", user.login_attempt)
```

Latihan 6

In [ ]:

```
class User:
    def __init__(self):
        self.__login_attempt = 0

    def increment_login_attempt(self):
        self.__login_attempt += 1

    def reset_login_attempt(self):
        self.__login_attempt = 0

    def get_login_attempt(self):
        return self.__login_attempt


# Contoh penggunaan
user = User()
user.increment_login_attempt()
user.increment_login_attempt()
user.increment_login_attempt()
print(user.get_login_attempt())  # Output: 3

user.reset_login_attempt()
print(user.get_login_attempt())  # Output: 0
```

Latihan 7

In [ ]:

```
class User:
    __max_login_attempt = 3

    def __init__(self, username, password):
        self.username = username
        self.password = password
        self.login_attempt = 0

    def increment_login_attempt(self):
        self.login_attempt += 1
        if self.login_attempt > self.__max_login_attempt:
            print("Anda telah melebihi batas maksimum percobaan login.")

    @classmethod
    def set_max_login_attempt(cls, max_attempt):
        cls.__max_login_attempt = max_attempt

    @classmethod
    def get_max_login_attempt(cls):
        return cls.__max_login_attempt
```