LAPORAN PRAKTIKUM PEMROGRAMAN 2

"*Inheritance*"

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

Disusun Oleh Kelompok 5

Hanadia Navaz Kamalia	**(3322600007)**
Anita Damayanti	**(3322600013)**
Bayu Kurniawan	**(3322600019)**
Abdul Muffid	**(3322600021)**
Eky Fernanda Setyawan P.	**(3322600025)**

PROGRAM STUDI D4 SAINS DATA TERAPAN

DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

MEI 2023

Exercise 1

- Buatlah *class* ***Restaurant*** dengan dua atribut, yakni ***name*** dan ***cuisine_type***.
- Lengkapi *class* tersebut dengan *method* ***describe_restaurant()*** yang berfungsi untuk mencetak informasi terkait dua atribut di atas.
- Lengkapi pula *class* tersebut dengan *method* ***open_restaurant()*** yang mencetak pesan ke layar bahwa restoran telah dibuka.
- Buatlah 2 objek dari *class* ***Restaurant***.
- Lakukan pemanggilan *method* ***describe_restaurant()*** untuk semua objek yang sudah dibuat.

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

  def describe_restaurant(self):
    print(f"Restaurant {self.name} is serves {self.cuisine_type}")

  def open_restaurant(self):
    print(f"Restaurant {self.name} is open now!")
```

```
sushi_tei = Restaurant("Sushi Tei", "Japanese cuisine")
sushi_tei.describe_restaurant()

myoung_ga = Restaurant("Myoung Ga", "Korean cuisine")
myoung_ga.describe_restaurant()
```

```
Restaurant Sushi Tei is serves Japanese cuisine
Restaurant Myoung Ga is serves Korean cuisine
```

Analysis: Kode di atas mendefinisikan sebuah *class* ***Restaurant*** yang memiliki atribut ***name*** dan ***cuisine_type*** untuk menyimpan informasi tentang nama restoran dan jenis masakan yang disediakan. Terdapat dua *method* di dalam *class* tersebut. *Method* ***describe_restaurant*** digunakan untuk mencetak informasi tentang restoran, yaitu nama dan jenis masakan yang disajikan. Metode ***open_restaurant*** digunakan untuk mencetak pesan bahwa restoran sedang buka. Dalam kode tersebut, dua objek ***Restaurant*** dibuat yaitu ***sushi_tei*** dan ***myoung_ga*** dengan masing-masing atributnya. *Output* dari kode di atas adalah informasi mengenai masing-masing restoran yang merupakan fungsi dari *method* ***describe_restaurant***.

Exercise 2

- Buatlah *class* ***IceCreamStall*** yang merupakan anak dari *class* ***Restaurant***.
- Lengkapi *class* tersebut dengan atribut ***flavors*** yang berisi daftar rasa es krim yang tersedia.
- Lengkapi pula *class* tersebut dengan *method* ***show_flavors()*** yang bertugas untuk menampilkan ke layar daftar rasa yang tersedia.
- Buat objek dari *class* ***IceCreamStall*** dan ***Restaurant***. Lakukan uji coba untuk mengakses semua *method* dan atribut yang ada pada kedua *class* tersebut.

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

  def show_flavors(self):
    print(f"Ice cream stall {self.name} is serves {self.flavors} flavors")
```

```
mc_donald = IceCreamStall("McDonald", "American cuisine", "Chocolate, Strawberry, and vanilla")
mc_donald.describe_restaurant()
mc_donald.open_restaurant()
mc_donald.show_flavors()
```

```
Restaurant McDonald is serves American cuisine
Restaurant McDonald is open now!
Ice cream stall McDonald is serves Chocolate, Strawberry, and vanilla flavors
```

Analysis: Kode di atas mendefinisikan *subclass* ***IceCreamStall*** yang merupakan turunan dari *class* ***Restaurant***. *Keyword* ***super()*** digunakan oleh *subclass* ***IceCreamStall*** untuk mengakses dan menjalankan metode ***__init__*** dari *superclass* ***Restaurant***. Hal ini memastikan bahwa atribut ***name*** dan ***cuisine_type*** dari *superclass* diwarisi oleh *subclass*. Ciri khas dari *subclass* ***IceCreamStall*** adalah adanya atribut tambahan ***flavors*** yang digunakan untuk menyimpan daftar rasa es krim yang tersedia. Terdapat *method* ***show_flavors*** yang digunakan untuk mencetak informasi mengenai rasa-rasa es krim yang tersedia di stan es krim tersebut. Objek dari *subclass* ini adalah ***mc_donald*** dengan informasi restoran *McDonald* yang menyajikan masakan Amerika serta memiliki daftar rasa es krim *Chocolate*, *Strawberry*, dan *Vanilla*. *Output* dari kode di atas adalah informasi tentang restoran *McDonald*, informasi bahwa restoran tersebut sedang buka, dan daftar rasa es krim yang tersedia yang merupakan fungsi dari *method* ***show_flavors***.

Exercise 3

- Modifikasi *method* ***describe_restaurant()*** pada *class* ***IceCreamStall*** agar menampilkan informasi spesifik tentang kios es krim.
- Lakukan uji coba ulang untuk mengakses *method* ***describe_restaurant()*** menggunakan objek dari *class* ***Restaurant*** dan ***IceCreamStall***

```
class IceCreamStall(Restaurant):
  def __init__(self, name, cuisine_type, time_open, time_close, flavors):
    super().__init__(name, cuisine_type)
    self.flavors = flavors
    self.time_open = time_open
    self.time_close = time_close

  def describe_restaurant(self):
    super().describe_restaurant()
    print(f"Ice cream stall {self.name} opens at {self.time_open} and closes at {self.time_close}")

  def show_flavors(self):
     print(f"Ice cream stall {self.name} is serves {self.flavors} flavors")
```

```
mc_donald = IceCreamStall("McDonald", "American cuisine", "9 AM", "10 PM", "Chocolate, Strawberry, and vanilla")
mc_donald.describe_restaurant()
mc_donald.open_restaurant()
mc_donald.show_flavors()
```

```
Restaurant McDonald is serves American cuisine
Ice cream stall McDonald opens at 9 AM and closes at 10 PM
Restaurant McDonald is open now!
Ice cream stall McDonald is serves Chocolate, Strawberry, and vanilla flavors
```

Analysis: Kode di atas merupakan perluasan dari *subclass* ***IceCreamStall*** yang mewarisi *class* ***Restaurant*** dengan *keyword* ***super()*** dalam ***__init__***. *Subclass* ini menambahkan atribut ***time_open*** dan ***time_close*** yang mengindikasikan waktu buka dan tutupnya stan es krim. Pada *method* ***describe_restaurant*** terjadi *overriding* yang memberikan informasi tambahan tentang waktu buka dan tutup. Objek dari *subclass* ini adalah ***mc_donald*** dengan informasi restoran *McDonald* yang menyajikan masakan Amerika, waktu buka pada jam 9 AM, tutup pada jam 10 PM, serta memiliki daftar rasa es krim *Chocolate*, *Strawberry*, dan *Vanilla*. *Output* dari *method* ***describe_restaurant*** yaitu informasi tentang restoran yang diwarisi dari *superclass* ***Restaurant*** dan informasi tambahan (waktu buka dan tutup) dari *subclass* ***IceCreamStall***, informasi bahwa restoran sedang buka, dan daftar rasa es krim yang tersedia.

Exercise 4

- Buatlah *class* ***User*** dengan atribut ***first_name*** dan ***last_name***. Set hak akses keduanya menjadi *private*.
- Lengkapi *class* tersebut dengan *method* ***describe_user()*** yang berfungsi untuk mencetak informasi terkait *user*.
- Lengkapi pula *class* tersebut dengan *method* ***greet_user()*** yang mencetak sapaan dengan nama *user*.
- Buatlah 2 objek dari *class* ***User***.
- Lakukan pemanggilan *method* ***describe_user()*** dan ***greet_user()*** untuk semua objek yang sudah dibuat.

```
class User:
  def __init__(self, first_name, last_name):
    self.__first_name = first_name
    self.__last_name = last_name

  def describe_user(self):
    print(f"First Name : {self.__first_name}")
    print(f"Last Name : {self.__last_name}")

  def greet_user(self):
    print(f"Hello {self.__first_name} {self.__last_name}!")
```

```
anita = User("Anita", "Damayanti")
anita.describe_user()
anita.greet_user()

print()

sophia = User("Sophia", "Aqila")
sophia.describe_user()
sophia.greet_user()

print()

try :
  print(f"first name : {anita.__first_name}")
except AttributeError:
  print("There's an attribute error")
```

```
First Name : Anita
Last Name : Damayanti
Hello Anita Damayanti!

First Name : Sophia
Last Name : Aqila
Hello Sophia Aqila!

There's an attribute error
```

Analysis: Kode di atas mendefinisikan *class* ***User*** yang memiliki atribut pribadi ***__first_name*** dan ***__last_name***. Melalui *method* ***describe_user()*** dan ***greet_user()***, *class* ini dapat mendeskripsikan dan menyapa *user* dengan nilai atribut yang diberikan. Namun, ketika mencoba mengakses atribut *private* secara langsung, terjadi ***AttributeError*** karena atribut *private* sehingga tidak dapat diakses secara langsung. Untuk mengakses atribut *private*, solusinya dapat menggunakan *method* akses seperti ***getters*** atau ***property*** yang didefinisikan dalam kelas.

Exercise 5

- Buatlah *class* ***Admin*** yang menuruni sifat dari *class* ***User***.
- Tambahkan atribut ***privileges*** pada *class* tersebut yang berisi daftar *privilege*, antara lain: "**dapat menambahkan *post***", "**dapat menghapus *post***", "**dapat melarang *user***".
- Tambahkan *method* ***show_privileges()*** untuk menampilkan daftar ***privileges*** yang dimiliki oleh admin.
- Buat objek dari *class* ***User*** dan ***Admin***. Lakukan uji coba untuk mengakses semua *method* dan atribut yang ada pada kedua *class* tersebut.

```
class Admin(User):
  def __init__(self, first_name, last_name):
    super().__init__(first_name, last_name)
    self.privileges = ["Add a post", "Delete a post", "Block an user"]

  def show_privileges(self):
    print(f"This admin has privileges :")
    for privilege in self.privileges :
      print(privilege)
```

```
hendy = Admin("Hendy", "Norman")
hendy.describe_user()
hendy.greet_user()
hendy.show_privileges()
```

```
First Name : Hendy
Last Name : Norman
Hello Hendy Norman!
This admin has privileges :
Add a post
Delete a post
Block an user
```

Analysis: Kode di atas mendefinisikan *class* ***Admin*** yang merupakan turunan dari *class* ***User***. *Class* ***Admin*** memiliki atribut tambahan ***privileges*** yang berisi daftar hak akses admin. *Method* ***show_privileges()*** digunakan untuk menampilkan hak akses admin dari objek yang dibuat. Dalam contoh ini, objek **hendy** dibuat sebagai instance dari *class* ***Admin***. Ketika kode dijalankan, objek **hendy** akan mencetak deskripsi pengguna dan sapaan, diikuti dengan daftar hak akses admin yang dimiliki oleh **hendy** yang merupakan fungsi dari *method* ***show_privileges()***.