

# UAS Pemrograman Berbasis Objek

Aldi Maulana Iqbal – 202108012

<https://github.com/depuuttt/PBOsmt3>

## Jelaskan menurut anda apa itu inheritance/kelas turunan dan buatlah contoh codingannya serta capture hasilnya

Inheritance adalah mekanisme di dalam pemrograman berorientasi objek yang memungkinkan sebuah kelas untuk menurunkan sifat-sifat dan perilaku dari kelas yang lain. Ini memungkinkan kita untuk membuat hierarki kelas yang berguna, di mana kelas yang lebih spesifik dapat menurunkan sifat-sifat dari kelas yang lebih umum.

Contoh inheritance dalam Python adalah sebagai berikut:

```
class Mobil:
    def __init__(self, merk, tahun):
        self.merk = merk
        self.tahun = tahun

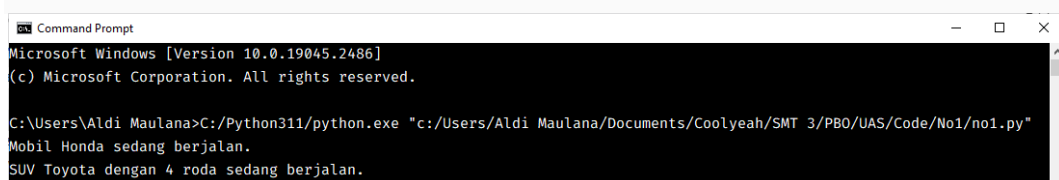
    def jalan(self):
        print(f"Mobil {self.merk} sedang berjalan.")

class SUV(Mobil):
    def __init__(self, merk, tahun, jumlah_roda):
        super().__init__(merk, tahun)
        self.jumlah_roda = jumlah_roda

    def jalan(self):
        print(f"SUV {self.merk} dengan {self.jumlah_roda} roda sedang berjalan.")

mobil = Mobil("Honda", 2020)
mobil.jalan()
# Output: Mobil Honda sedang berjalan.

suv = SUV("Toyota", 2021, 4)
suv.jalan()
# Output: SUV Toyota dengan 4 roda sedang berjalan.
```



```
Command Prompt
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aldi Maulana>C:/Python311/python.exe "c:/Users/Aldi Maulana/Documents/Coolyeah/SMT 3/PBO/UAS/Code/No1/no1.py"
Mobil Honda sedang berjalan.
SUV Toyota dengan 4 roda sedang berjalan.
```

Dalam contoh ini, kelas `Mobil` adalah kelas induk dan kelas `SUV` adalah kelas turunan. Kelas `SUV` mewarisi sifat-sifat dari kelas `Mobil` dan juga memiliki sifat tambahan `jumlah_roda`. Kelas `SUV` juga memiliki metode `jalan` yang di-override dari kelas `Mobil`.

## Jelaskan menurut anda apa itu encapsulation dan buatlah contoh codingannya serta capture hasilnya

Encapsulation adalah mekanisme yang digunakan untuk menyembunyikan implementasi dari suatu class dari pemakai class tersebut. Ini meningkatkan tingkat keamanan dan modularitas dari kode. Dalam python, encapsulation dapat dicapai dengan menggunakan private dan protected variable dan method.

Contoh sederhana dari encapsulation dalam python:

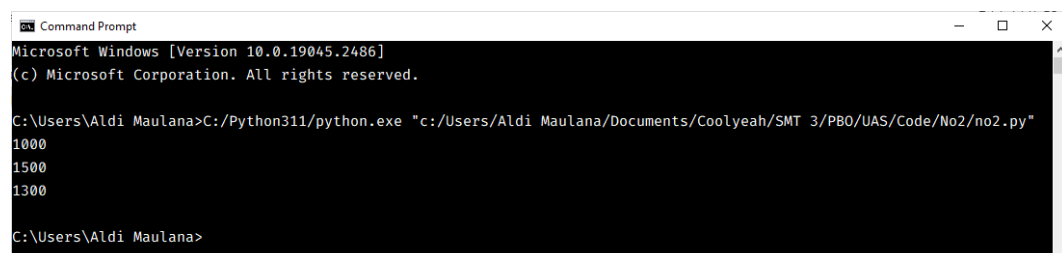
```
class BankAccount:
    def __init__(self, balance):
        self.__balance = balance

    def deposit(self, amount):
        self.__balance += amount

    def withdraw(self, amount):
        if self.__balance >= amount:
            self.__balance -= amount
            return True
        else:
            return False

    def get_balance(self):
        return self.__balance

account = BankAccount(1000)
print(account.get_balance()) # 1000
account.deposit(500)
print(account.get_balance()) # 1500
account.withdraw(200)
print(account.get_balance()) # 1300
```



```
Command Prompt
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aldi Maulana>C:/Python311/python.exe "c:/Users/Aldi Maulana/Documents/Coolyeah/SMT 3/PB0/UAS/Code/No2/no2.py"
1000
1500
1300

C:\Users\Aldi Maulana>
```

Dalam contoh ini, variable `__balance` adalah private variable. Ini hanya dapat diakses dan diubah melalui method yang didefinisikan dalam class. Pemakai class tidak dapat mengakses atau mengubah variable ini secara langsung.

Jelaskan menurut anda apa itu polymorphism dan buatlah contoh codingannya serta capture hasilnya

Polymorphism adalah konsep dalam pemrograman yang mengizinkan sebuah objek atau kelas untuk memiliki beberapa bentuk atau sifat yang berbeda. Ini memungkinkan kita untuk menggunakan objek atau kelas yang sama dengan cara yang berbeda, tanpa harus mengetahui jenis objek atau kelas yang sebenarnya digunakan.

Contoh sederhana dari polymorphism dalam python:

```
class Hewan:
    def suara(self):
        pass

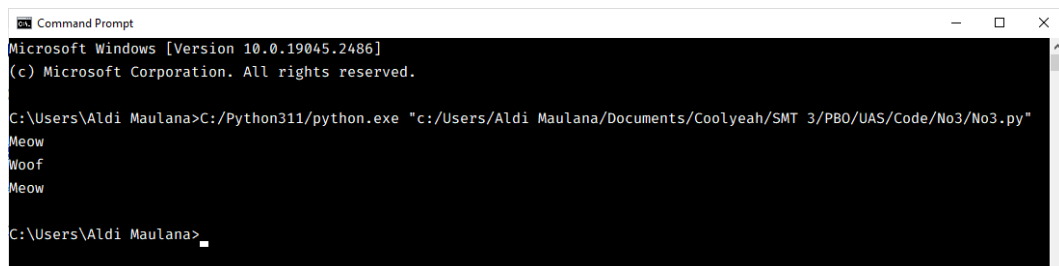
class Kucing(Hewan):
    def suara(self):
        print("Meow")

class Anjing(Hewan):
    def suara(self):
        print("Woof")

def main():
    hewan1 = Kucing()
    hewan2 = Anjing()
    hewan3 = Kucing()

    hewan1.suara() # akan mencetak "Meow"
    hewan2.suara() # akan mencetak "Woof"
    hewan3.suara() # akan mencetak "Meow"

if __name__ == "__main__":
    main()
```



```
Command Prompt
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aldi Maulana>C:\Python311\python.exe "c:/Users/Aldi Maulana/Documents/Coolyeah/SMT 3/PB0/UAS/Code/No3/No3.py"
Meow
Woof
Meow

C:\Users\Aldi Maulana>
```

Di dalam contoh di atas, kita membuat kelas Hewan yang memiliki method suara() yang tidak diimplementasikan. Kemudian kita membuat dua kelas

turunan yaitu Kucing dan Anjing yang masing-masing mengimplementasikan method `suara()` dengan output yang berbeda. Kemudian kita membuat objek dari kelas Kucing dan Anjing, dan pada saat method `suara()` dijalankan, objek akan mengeluarkan suara yang sesuai dengan kelas yang digunakan. Ini adalah contoh dari polymorphism dimana objek Hewan dapat digunakan dalam berbagai bentuk atau sifat yang berbeda.

## Jelaskan menurut anda apa itu function exception handler dan buatlah contoh codingannya serta capture hasilnya

Exception handler adalah mekanisme di dalam pemrograman yang digunakan untuk menangani kesalahan (error) yang mungkin terjadi selama eksekusi program. Ini memungkinkan kita untuk menangani kesalahan secara terpisah dari logika utama program, sehingga program dapat beroperasi dengan normal meskipun terdapat kesalahan.

Function exception handler dalam Python dapat dibuat menggunakan perintah `try` dan `except`. Contohnya sebagai berikut:

```
def bagi(a, b):  
    try:  
        hasil = a / b  
    except ZeroDivisionError:  
        print("Error: Pembagian dengan nol.")  
        return  
    return hasil
```

```
hasil = bagi(5, 2)
```

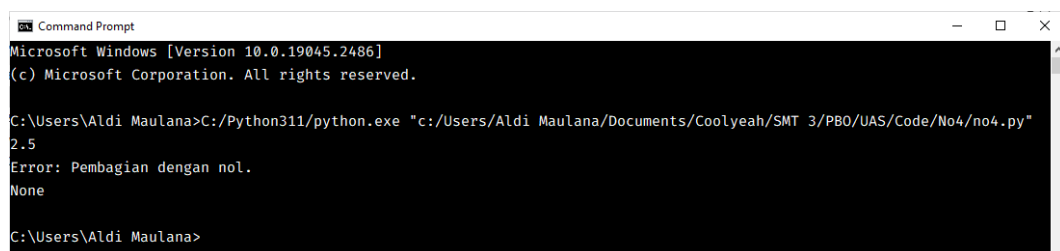
```
print(hasil)
```

```
# Output: 2.5
```

```
hasil = bagi(5, 0)
```

```
print(hasil)
```

```
# Output: Error: Pembagian dengan nol.
```



```
Command Prompt  
Microsoft Windows [Version 10.0.19045.2486]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\Aldi Maulana>C:/Python311/python.exe "c:/Users/Aldi Maulana/Documents/Coolyeah/SMT 3/PB0/UAS/Code/No4/no4.py"  
2.5  
Error: Pembagian dengan nol.  
None  
  
C:\Users\Aldi Maulana>
```

Dalam contoh di atas, function `bagi` mencoba untuk melakukan pembagian dua bilangan dengan operator `'/'`. Namun jika terjadi kesalahan `ZeroDivisionError` yang disebabkan karena pembagian dengan nol, maka akan dikeluarkan pesan error dan function akan segera keluar tanpa return apapun.

Namun jika tidak terjadi kesalahan, maka function akan mengembalikan hasil pembagian tersebut.

## Jelaskan menurut anda apa itu GUI dan berikan contoh codingannya serta capture hasilnya

GUI (Graphical User Interface) adalah sebuah antarmuka pengguna yang menyediakan tampilan grafis sebagai cara untuk menginteraksi dengan perangkat lunak atau sistem. GUI menyediakan tampilan visual yang memudahkan pengguna untuk mengakses fitur-fitur yang tersedia dalam sebuah aplikasi atau sistem, seperti tombol, menu, dan ikon.

Python memiliki beberapa library yang dapat digunakan untuk membuat GUI, salah satunya adalah Tkinter. Tkinter adalah library standar Python yang digunakan untuk membuat antarmuka grafis. Berikut ini adalah contoh sederhana dari GUI sederhana menggunakan Tkinter:

```
import tkinter as tk

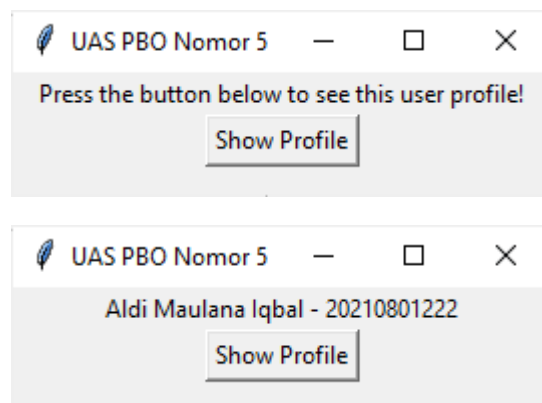
def tampil_pesan():
    label.config(text="Hello, World!")

root = tk.Tk()
root.title("Contoh GUI")

label = tk.Label(root, text="Tekan tombol dibawah")
label.pack()

tombol = tk.Button(root, text="Tekan Saya", command=tampil_pesan)
tombol.pack()

root.mainloop()
```



Dalam contoh di atas, kita menggunakan Tkinter untuk membuat sebuah jendela utama (root) yang berisi sebuah label dan sebuah tombol. Ketika tombol

ditekan, label akan berubah menjadi "Hello, World!". Tkinter juga menyediakan banyak komponen lain yang dapat digunakan untuk membuat antarmuka yang lebih kompleks.

## Jelaskan menurut anda bagaimana App bisa berkomunikasi dengan database dan buatlah contoh codingannya serta capture hasilnya

Aplikasi atau App dapat berkomunikasi dengan database adalah sebuah aplikasi yang dapat mengakses, menambah, mengubah, atau menghapus data yang disimpan dalam sebuah database. Proses ini dapat dilakukan dengan menggunakan sebuah bahasa pemrograman yang dapat berkomunikasi dengan database, seperti SQL (Structured Query Language).

Python menyediakan library yang dapat digunakan untuk berkomunikasi dengan berbagai macam jenis database seperti MySQL, PostgreSQL, dan SQLite. Salah satu library yang populer digunakan untuk berkomunikasi dengan MySQL adalah MySQL Connector. Berikut ini adalah contoh sederhana dari koneksi ke MySQL dan eksekusi query SELECT:

```
import mysql.connector

# koneksi ke database
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="db_uas_pbo"
)

# membuat cursor
cursor = db.cursor()

# menjalankan query SELECT
cursor.execute("SELECT * FROM mahasiswa")

# menampilkan hasil query
result = cursor.fetchall()
for data in result:
    print(data)

# menutup koneksi
cursor.close()
db.close()
```

```
Command Prompt
Microsoft Windows [Version 10.0.19045.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aldi Maulana>C:\Python311\python.exe "c:/Users/Aldi Maulana/Documents/Coolyeah/SMT 3/PBO/UAS/Code/No6(belum kel
ar)/no6.py"
('001', 'John Doe', 'Laki-Laki')
('002', 'Jane Doe', 'Perempuan')
('003', 'Mark Smith', 'Laki-Laki')
('004', 'Emily Johnson', 'Perempuan')
('005', 'Michael Brown', 'Laki-Laki')

C:\Users\Aldi Maulana>
```

Dalam contoh di atas, kita mengimport `mysql.connector` dan menggunakannya untuk terhubung ke database MySQL yang berada di localhost dengan user root dan password yang telah ditentukan. Kemudian kita membuat cursor untuk mengeksekusi query `SELECT` dan menampilkan hasilnya. Seiring dengan itu, kita juga harus menutup koneksi agar tidak terjadi memory leak.

**Buatlah Aplikasi sederhana yang terkoneksi dengan database push ke github source codenya dan capture hasilnya**

Source Code:

```
import tkinter as tk
import sqlite3

# Buat koneksi ke database
conn = sqlite3.connect('db_uas_pbo.db')
cursor = conn.cursor()

# Eksekusi query untuk mengambil data mahasiswa
query = 'SELECT nim, nama, jenis_kelamin FROM mahasiswa'
cursor.execute(query)
data = cursor.fetchall()

# Buat GUI
root = tk.Tk()
root.title('Data Mahasiswa')

# Buat tabel
table = tk.Treeview(root, columns=(
    'nim', 'nama', 'jenis_kelamin'), show='headings')
table.heading('nim', text='NIM')
table.heading('nama', text='Nama')
table.heading('jenis_kelamin', text='Jenis Kelamin')

# Tambahkan data ke tabel
for item in data:
    table.insert('', 'end', values=item)
```



```
# Tampilkan tabel
table.pack()
root.mainloop()

# Tutup koneksi ke database
conn.close()
```