

**!! Catatan:** Diharapkan seluruh penggerjaan Assignment tidak sepenuhnya mengandalkan penggunaan AI !!

*"Proses belajar ibarat menanam pohon. Jika hanya mengandalkan AI tanpa memahami esensinya, yang berkembang bukan kompetensimu, melainkan ketergantungan yang melemahkan."*

- Learning Design Dibimbang

# Assignment Guidance: OOP

Data Science dan Data Analyst Bootcamp

## Periode Pembelajaran

Python Advanced

## Objectives

1. Student mampu memahami dan mengimplementasikan konsep class dan object.
2. Student mampu mengimplementasikan encapsulation dengan access modifiers.
3. Student mampu mengimplementasikan inheritance dan multiple inheritance.
4. Student mampu mengimplementasikan polymorphism dan method overriding.
5. Student mampu mengimplementasikan abstraction dan interfaces.

## Deskripsi Assignment

Assignment ini bertujuan untuk membantu student memahami dan mengimplementasikan konsep Object-Oriented Programming (OOP) dalam Python. Student akan mempraktikkan class & object, encapsulation, inheritance, polymorphism, dan abstraction melalui berbagai skenario pemrograman.

Dalam assignment ini, student akan membuat class dengan atribut dan metode, mengelola akses data menggunakan encapsulation, menerapkan inheritance, melakukan method overriding, serta menggunakan abstraction. Praktik ini akan memperkuat pemahaman OOP dan aplikasinya dalam pengembangan perangkat lunak.

## Detail Assignment

Untuk menyelesaikan assignment ini, kerjakan kelima soal berikut berikut:

### 1. Soal 1: Membuat Class dan Object

**Buatlah sebuah class Mahasiswa yang memiliki atribut nama, nim, dan jurusan. Tambahkan metode info\_mahasiswa yang akan menampilkan informasi mahasiswa (nama, nim, dan jurusan).** Perhatikan bagaimana mendeklarasikan class dan mendefinisikan atribut serta metode.

- Pastikan atribut dapat diakses dalam class dengan benar.
- Gunakan metode untuk menampilkan informasi dari instance yang dibuat.

**Goals:** Memahami konsep dasar class dan object, serta bagaimana cara mengakses atribut dan metode dalam class.

### 2. Soal 2: Mengimplementasikan Encapsulation

**Buat class Tabungan yang memiliki atribut saldo (bersifat private). Tambahkan metode deposit untuk menambah saldo dan tarik untuk mengambil saldo. Jika saldo tidak mencukupi, tampilkan pesan "Saldo tidak mencukupi."**

- Pahami konsep aksesibilitas atribut dalam OOP.
- Pastikan pengelolaan atribut private dapat dilakukan melalui metode yang sesuai.
- Terapkan validasi yang diperlukan untuk menjaga integritas data.

**Goals:** Memahami konsep encapsulation, serta bagaimana menggunakan access modifiers dalam class Python.

### 3. Soal 3: Mengimplementasikan Inheritance

**Buat class Pegawai yang memiliki atribut nama dan gaji. Buat subclass Manager yang mewarisi class Pegawai dan menambahkan atribut tunjangan. Tambahkan metode info\_manager yang menampilkan informasi nama, gaji, dan tunjangan.**

- Pahami bagaimana inheritance memungkinkan penggunaan kembali kode dari superclass.
- Pastikan subclass dapat memperluas fungsionalitas superclass dengan atribut/metode tambahan.
- Pastikan relasi antara superclass dan subclass terimplementasi dengan baik.

**Goals:** Memahami konsep inheritance dan bagaimana membangun hubungan antara superclass dan subclass.

#### 4. Soal 4: Polymorphism dan Method Overriding

Buat class Kendaraan dengan metode bahan\_bakar yang hanya menampilkan "Jenis bahan bakar tidak diketahui." Buat subclass Mobil dan Motor, lalu override metode bahan\_bakar pada masing-masing subclass untuk menampilkan jenis bahan bakar yang digunakan (contoh: "Bensin" untuk Mobil, "Solar" untuk Motor).

- Pahami konsep method overriding dan bagaimana menerapkannya dalam subclass.
- Pastikan method yang di-override memberikan hasil yang sesuai dengan subclass terkait.
- Gunakan metode yang sama tetapi dengan implementasi berbeda untuk setiap subclass.

**Goals:** Memahami konsep polymorphism dan method overriding, serta bagaimana menerapkannya dalam subclass.

#### 5. Soal 5: Abstraction dan Interfaces

Buat class abstrak Hewan dengan metode abstrak suara. Buat subclass Kucing dan Anjing yang mengimplementasikan metode suara untuk mengeluarkan suara spesifik masing-masing (contoh: "Meow" untuk Kucing, "Bark" untuk Anjing).

- Pahami konsep abstraction dalam OOP dan bagaimana class abstrak digunakan.
- Implementasikan metode abstrak yang diwajibkan pada subclass.
- Pastikan kode tetap terstruktur dan mengikuti prinsip OOP.

**Goals:** Memahami konsep abstraction serta bagaimana mengimplementasikan class abstrak dan metode abstrak dalam Python.

### Tools

Google Collaboratory

### Pengumpulan Assignment

**Deadline :**

Maksimal H+7 Kelas (Pukul 23.30 WIB)

**Details :**

Dikumpulkan dalam bentuk Link Google Collaboratory, secara INDIVIDU, di LMS

## Indikator Penilaian

No.	Aspek Penilaian	Parameter	Bobot Maksimal
1	Pemahaman Class dan Object	Mampu membuat class dan object yang sesuai	20
2	Implementasi Encapsulation	Mampu menggunakan access modifiers dengan benar	20
3	Implementasi Inheritance	Mampu menerapkan konsep inheritance	20
4	Implementasi Polymorphism	Mampu melakukan method overriding	20
5	Implementasi Abstraction	Mampu menggunakan class abstrak dan method abstrak	20

## Referensi/Template

1. <https://www.petanikode.com/javascript-array/>

### Sanksi Penggunaan AI:

Apabila student terdeteksi 100% menggunakan AI, maka hasil assignment akan diberikan skor 0.

### Ketentuan Pencapaian Nilai:

Nilai minimum Lulus Penyaluran Kerja: 75

Nilai minimum Lulus Bootcamp: 65

### Ketentuan Penilaian:

Mengumpulkan Assignment tepat waktu: Sesuai dengan nilai yang diberikan mentor

Mengumpulkan Assignment 12 jam setelah deadline: - 3 dari nilai yang diberikan mentor

Mengumpulkan Assignment 1x 24 Jam setelah deadline: - 6 dari nilai yang diberikan mentor

Mengumpulkan Assignment 2 x 24 Jam setelah deadline: - 12 dari nilai yang diberikan mentor

Mengumpulkan Assignment 3 x 24 Jam setelah deadline: - 18 dari nilai yang diberikan mentor

Mengumpulkan Assignment 4 x 24 Jam setelah deadline: - 24 dari nilai yang diberikan mentor

Mengumpulkan Assignment 5 x 24 Jam setelah deadline: - 30 dari nilai yang diberikan mentor

Mengumpulkan Assignment 6 x 24 Jam setelah deadline: - 36 dari nilai yang diberikan mentor

Mengumpulkan Assignment 7 x 24 Jam setelah deadline: - 42 dari nilai yang diberikan mentor