

# **RESPONSI PEMROGRAMAN BERORIENTASI OBJEK**



**Nama: Yoga Rusdi Hantoro**

**NPM : 5230411266**

**Kelas :VII**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIV TEKNOLOGI YOGYAKARTA**

**2024/2025**

## 1. Menjelaskan Perbedaan *Use Case Diagram* dan *Class Diagram*

- **Use Case Diagram** : Menggambarkan interaksi pengguna dan program/system. Tujuannya adalah menunjukkan bagaimana sistem akan digunakan dan apa saja yang mungkin terjadi dalam penggunaan sistem. Diagram ini biasanya fokus pada fungsi sistem dan tidak mendalami struktur internal sistem.
- **Class Diagram** : Menggambarkan struktur internal dari sistem dengan menunjukkan kelas-kelas, atribut, metode, dan hubungan antar kelas (seperti pewarisan atau asosiasi). Diagram ini lebih menjelaskan bagaimana data dan fungsi diatur dalam sistem dan bagaimana data dan fungsi tersebut berhubungan satu dengan yang lain

## 2. Menjelaskan Jenis-Jenis *Dependensi*

Dalam UML, *dependensi* adalah hubungan dimana satu elemen bergantung pada elemen lain untuk berfungsi. Beberapa jenis ketergantungan adalah:

- **Ketergantungan Penggunaan** : Satu elemen membutuhkan elemen lain untuk menjalankan fungsinya. Misalnya, sebuah metode menggunakan metode lain dari kelas yang berbeda.
- **Ketergantungan Warisan** : Salah satu element mewarisi elemen lain. Ini sering terlihat dalam hubungan pewarisan.
- **Ketergantungan Asosiasi** : Hubungan antara dua kelas yang melibatkan interaksi satu sama lain (seperti asosiasi dua arah antara dua kelas).
- **Agregasi & Komposisi** : Menyatakan bahwa satu kelas berisi atau terdiri dari kelas lain; pada *komposisi* , kelas yang satu bergantung penuh pada yang lain untuk eksis.

## 3. Apa Perbedaan Pemrograman Terstruktur dengan Berorientasi Objek? jelaskan.

- **Pemrograman Terstruktur** : Pendekatan pemrograman yang fokus pada urutan proses dan pembagian program menjadi sub-program atau fungsi yang lebih kecil. Contohnya adalah bahasa seperti C. Biasanya menekankan alur prosedural tanpa adanya objek.
- **Pemrograman Berorientasi Objek (OOP)** : Pendekatan yang fokus pada objek yang memiliki atribut dan metode. Program dibangun di sekitar entitas yang disebut objek,

yang mewakili konsep dunia nyata. Contohnya adalah Java, Python, dan C++. OOP memiliki konsep utama seperti *pewarisan* , *polimorfisme* , *enkapsulasi* , dan *abstraksi* .

#### 4. Menjelaskan Konsep Objek dan Beri Contohnya

- **Objek** : Adalah entitas yang memiliki data (disebut atribut/properti) dan perilaku (disebut metode/fungsi). Dalam pemrograman yang berorientasi pada objek, objek adalah representasi dari hal-hal dunia nyata.

**Contoh** : *Mobil*, *ibuMobil1* dengan *atrberjalan()* dan *abberhenti()* .

#### 5. Menjelaskan Jenis-Jenis *Access Modifier* dan

- **Publik** : Di
- **Pribadi** : Atr
- **Dilindungi** : A
- **Default (Tanpa Pengubah)** : Akse

**Contoh Kode dalam python:**

```
class hewan:
    def __init__(self, nama, berjalan):
        self.nama = nama # public attribute
        self._berjalan = berjalan # protected attribute
        self.__jenis_hewan = "herbivora" # private attribute

    def get__jenis_hewan(self): # public method
        return self.__jenis_hewan

    def _info_berjalan(self): # protected method
        return f"Model: {self._berjalan}"

    def __cek_kesehatan(self): # private method
        return "hewan anda dalam keadaan sehat"

# Instansiasi objek Mobil
hewan_saya = hewan("kerbau", "paru-paru")

# Mengakses atribut public
```

```

print(hewan_saya.nama) # Output: kerbau

# Mengakses metode public
print(hewan_saya.get__jenis_hewan()) # Output: herbivora

# Mengakses atribut protected (tidak dianjurkan, tapi masih bisa diakses)
print(hewan_saya._info_berjalan) # Output: SUV

# Mengakses metode protected (tidak dianjurkan, tapi masih bisa diakses)
print(hewan_saya.__jenis_hewan()) # Output: Model: SUV

# Mengakses atribut private (tidak bisa langsung diakses)
# print(hewan_saya._info_berjalan) # Akan menyebabkan AttributeError

# Mengakses metode private (tidak bisa langsung diakses)
# print(hewan_saya.__jenis_hewan()) # Akan menyebabkan AttributeError

```

## 6. Gambarkan Contoh Pewarisan dalam *Class Diagram*

data yaitu data pegawai dengan class diantara nya class induk, class dosen dan class pegawai non dosen. Dengan tribute yang berisi id\_pegawai, Nama, Gol, dan tanggal lahir

