

**RESPONSI MATAKULIAH PEMROGRAMAN BERBASIS  
OBJEK  
TAHUN 2024**



Disusun oleh:

Rafi Farrij Al

5230411274

**FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
02/11/2024**

## SOAL TEORI

1. Jelaskan perbedaan antara class diagram dan use case diagram
2. Jelaskan jenis jenis dependensi
3. Apa perbedaan pemrograman terstruktur dengan berorientasi obje. Jelaskan
4. Jelaskan konsep objek dan beri contohnya
5. Jelaskan jenis jenis access modifier beri contohnya dalam baris pemrograman
6. Gambarkan contoh pewarisan dalam diagram class

## JAWABAN SOAL

1. Use Case Diagram: Gambaran tentang bagaimana pengguna berinteraksi dengan sistem, menunjukkan fitur apa saja yang bisa dipakai oleh pengguna.

Class Diagram: Gambaran tentang struktur sistem, termasuk kelas-kelas, atribut, metode, dan hubungan antar kelas. Ini lebih fokus ke desain sistem.

2. Association: Hubungan biasa antar objek (contoh: murid belajar di sekolah).  
Aggregation: Hubungan "bagian dari" tapi bisa berdiri sendiri (contoh: tim berisi pemain, pemain bisa pindah ke tim lain).  
Composition: Hubungan "bagian dari" yang kuat, tidak bisa berdiri sendiri (contoh: rumah dan kamar, kamar tidak ada kalau rumah dihapus).  
Dependency: Hubungan sementara, satu objek butuh bantuan dari objek lain.
3. Pemrograman Terstruktur: Fokus ke langkah-langkah atau prosedur. Program dibagi jadi fungsi atau prosedur.

Pemrograman Berorientasi Objek (OOP): Fokus ke objek yang punya data dan perilaku. Misalnya, konsep objek, pewarisan, dan polimorfisme.

4. Konsep objek dalam pemrograman merujuk pada entitas yang memiliki data dan perilaku. Dalam pemrograman berorientasi objek (OOP), objek adalah instansi dari kelas, yang merupakan blueprint atau template untuk membuat objek. Setiap objek memiliki atribut (data) dan metode (fungsi atau perilaku) yang terkait.

Contoh:

- Class Mobil :
  - Atribut: warna, merek, tahun, kecepatan
  - Metode: berjalan(), berhenti(), tambahKecepatan()
- Objek:
  - Mobil1: warna = merah, merek = Toyota, tahun = 2020, kecepatan = 0
  - Mobil2: warna = biru, merek = Honda, tahun = 2021, kecepatan = 0

5. Access modifier adalah pengatur visibilitas atau aksesibilitas atribut dan metode dalam suatu kelas. Dalam pemrograman berorientasi objek, terdapat beberapa jenis access modifier yang umum digunakan:

- Publik  
class Mobil:  
    def \_\_init\_\_(self, warna):  
        self.warna = warna # public attribute  
  
    def berjalan(self):  
        print(f'{self.warna} mobil sedang berjalan.')  
  
mobil1 = Mobil("merah")  
mobil1.berjalan()  
  
• Private  
class Mobil:  
    def \_\_init\_\_(self, nomor\_mesin):  
        self.\_\_nomor\_mesin = nomor\_mesin # private attribute  
  
    def \_\_periksa\_mesin(self):  
        print("Mesin diperiksa.")  
  
    def cek\_mesin(self):  
        self.\_\_periksa\_mesin()  
  
mobil2 = Mobil("1234AB")  
mobil2.cek\_mesin()

- Protected

```
class Mobil:
    def __init__(self, merek):
        self._merek = merek # protected attribute

    def tampilkan_merek(self):
        print(f'Merek mobil: {self._merek}')

class MobilBalap(Mobil):
    def __init__(self, merek):
        super().__init__(merek)

    def tunjukkan_merek(self):
        print(f'Merek balap: {self._merek}')

mobil_balap = MobilBalap("Ferrari")
mobil_balap.tunjukkan_merek()
```

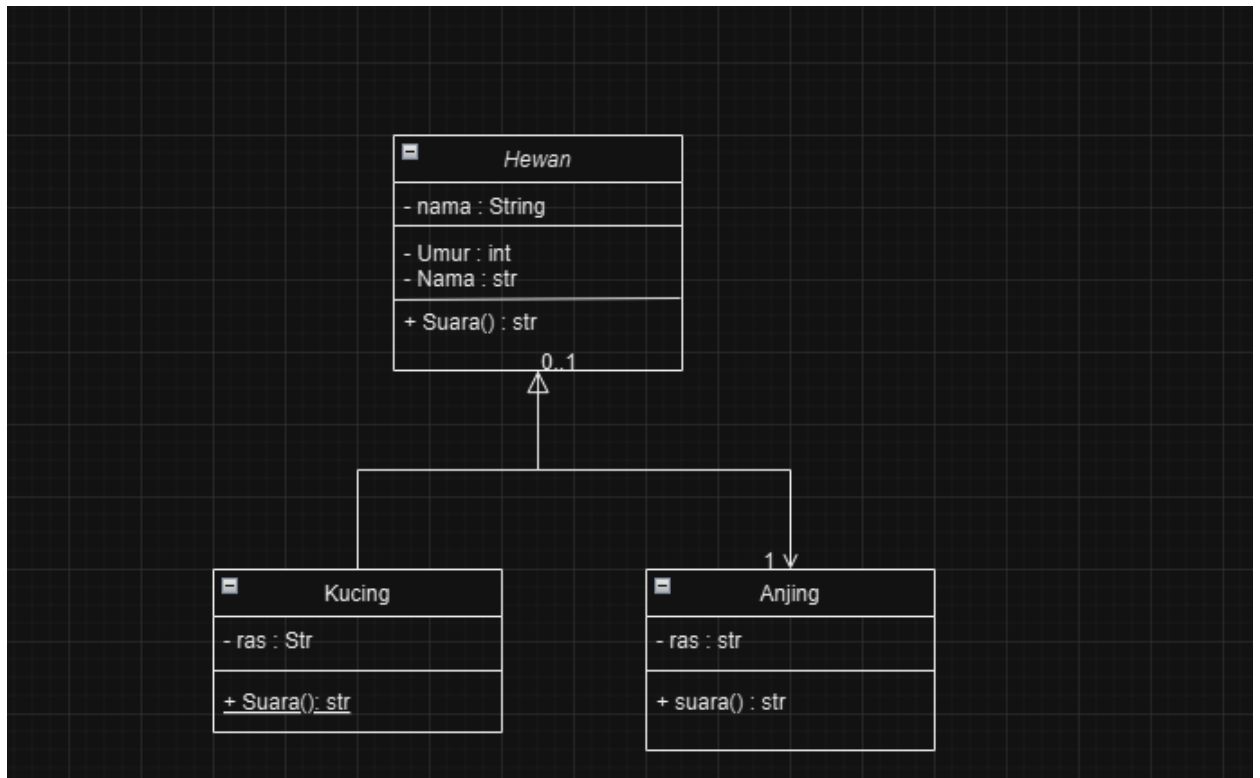
- Default

```
class Mobil:
    def __init__(self, model):
        self.model = model # default access

    def tampilkan_model(self):
        print(f'Model mobil: {self.model}')

mobil3 = Mobil("Civic")
mobil3.tampilkan_model()
```

6.



#### PENJELASAN :

##### 1. Kelas Induk (Hewan):

- Memiliki atribut nama dan umur.
- Memiliki metode `Suara ()` yang dapat dioverride oleh kelas anak.

##### 2. Kelas Anak (Kucing dan Anjing):

- Mewarisi atribut dan metode dari kelas `Hewan`.
- Masing-masing memiliki atribut tambahan `ras` dan mengimplementasikan metode `suara ()` sesuai dengan jenis hewan.