

# TUGAS

## PEMROGRAMAN BERORIENTASI OBJEK



Disusun oleh:

Fauzi Kurniawan (5230411251)

PROGRAM STUDI INFORMATIKA  
UNIVERSITAS TEKNOLOGI YOGYAKARTA

2024

## Soal Teori

### 1. Jelaskan perbedaan use case diagram dengan class diagram

Class diagram dan use case diagram adalah diagram yang berada di dalam UML (Unified Modeling Language) untuk membantu merancang sistem perangkat lunak namun dengan focus yang berbeda

- Tujuan Utama
  - Class Diagram:  
Untuk menunjukkan struktur system seperti kelas-kelas, atribut dan fungsi dalam kelas serta hubungan antar kelas
  - Use Case Diagram:  
Menunjukkan bagaimana pengguna atau sistem berinteraksi dengan sistem yang dibuat. Hal ini membantu memahami apa yang harus dilakukan oleh system berdasarkan kebutuhan pengguna
- Apa yang ditampilkan
  - Class Diagram  
Menampilkan kelas, atribut, metode, dan hubungan seperti "kelas A berhubungan dengan kelas B".
  - Use Case Diagram  
Menampilkan pengguna dan daftar tugas atau fitur/use case yang mereka lakukan, serta interaksi antara aktor dan sistem.
- Digunakan untuk Apa?
  - Class Diagram:  
Digunakan untuk mendesain struktur internal system, sebagai contoh saat membuat database atau merancang program
  - Use Case Diagram:  
Digunakan untuk mengumpulkan kebutuhan fungsional system supaya pengembang tahu fitur apa saja yang harus disediakan

Dengan kata lain perbedaan antara use case diagram dan class diagram adalah class diagram menunjukkan struktur bagian dalam system sedangkan use case diagram menunjukkan interaksi antara user dan system.

### 2. Jelaskan jenis-jenis dispenses?

Dispensesi atau pengenduran (exception) adalah pengelolaan kesalahan atau pengecualian.

Jenis-jenis Dispensasi atau Exception dalam Pemrograman:

- Checked Exception  
Adalah kesalahan yang terjadi pada saat kompilasi. Dan sebagai programmer checed exception dengan cara try-catch atau mendeklarasikan dengan throws. Dengan contoh kesalahan pada operasi input dan output
- Unchecked Exception  
Kesalahan saat runtime biasanya terjadi karena ada logika atau program yang salah sebagai contoh 'NullPointerException'
- Error

Seperti namanya eror atau ada kesalahan biasa terjadi dan berada diluar kendali program. Hal ini sering berhubungan dengan lingkungan runtime atau memori

- Custom Exception

Adalah eception khusus yang dibuat user atau programmer untuk menangani situasi tertentu yang tidak bisa digapai oleh exception bawaan.

### 3. Perbedaan Pemrograman Terstruktur dengan Berorientasi Objek

- Pemrograman Terstruktur

Memiliki fokus dengan prodesur dan alur logika program dengan menggunakan fungsi dan prosedur menyebarkan atau mengkategorikan masalah menjadi kecil. Dan program ini bersifat linier serta menggunakan struktur seperti pengulangan dan seleksi.

- Pemrograman Berorientasi Objek

Berfokus dengan objek yang merupakan gabungan dari data dan metode. Hal ini menggunakan konsep class,objek,pewarisan dll, yang membuat program lebih mudah dikelola serta dikembangkan

### 4. Konsep Objek dan Contohnya

#### Pengertian

Objek adalah entitas yang berisi data dan perilaku. Setiap objek adalah instens dari setiap kelas.

Sebagai contoh dalam pemrograman pemesanan kendaraan 'Mobil' bisa menjadi kelas, dab spesifik mobil seperti merek ataupun nama model "Gt Supra MK 4" adalah objek dalam kelas 'Mobil'

### 5. Jenis-jenis Access Modifier dan Contohnya dalam Pemrograman

- Jenis-jenis access Modifier ada 3 yaitu public,Potect dan private sebagai contohnya dalam pemrograman python

```
1 #Contoh Publik, Provate, dan Protect dalam python
2 class Kendaraan:
3     def __init__(self, merk, tahun):
4         self.merk = merk          # Public attribute
5         self._tahun = tahun       # Protected attribute
6         self.__kecepatan = 0     # Private attribute
7
8     def set_kecepatan(self, kecepatan):
9         self.__kecepatan = kecepatan
10        print(f"Kecepatan diatur ke {self.__kecepatan} km/jam")
11
12    def get_kecepatan(self):
13        return self.__kecepatan
14
15 class Mobil(Kendaraan):
16     def __init__(self, merk, tahun, jenis):
17         super().__init__(merk, tahun)
18         self.jenis = jenis
19
20     def info(self):
21         print(f"Mobil {self.merk}, Tahun {self._tahun}, Jenis {self.jenis}")
22
23 mobil = Mobil("Toyota", 2020, "SUV")
24 mobil.info()
25
26 #Menampilkan
27 print(mobil.merk)
28 print(mobil._tahun) |
29 print(mobil.get_kecepatan())
30
```

Disini diketahui perbedaan dari ketiga hal tersebut dan perbedaannya terletak pada

Public = .merk

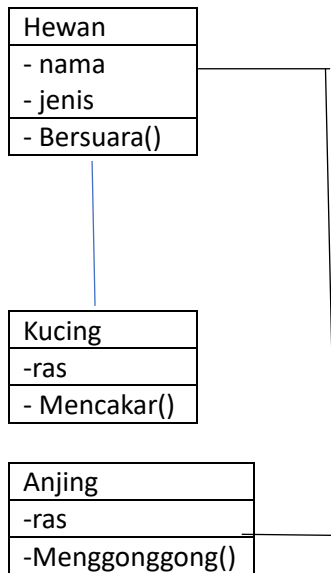
Private = \_\_kecepatan

Protect = \_tahun

Dengan intinya adalah banyaknya underscore.

#### 6. Contoh Pewarisan dalam Diagram Kelas

Berikut adalah contoh dari diagram pewarisan pada Hewan:



Dimana kucing dan anjing mewarisi sifat dari hewan