

Responsi 1



Disusun oleh:

Nama : Taufik Yoga Pratama

NPM : 5230411269

Kelas : VIII

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

2023/2024

Soal Teori

1. Jelaskan perbedaan use case diagram dengan class diagram?

Use Case Diagram: Digunakan untuk menggambarkan interaksi antara aktor (pengguna atau sistem lain) dengan sistem yang akan dikembangkan. Diagram ini menyoroti fungsionalitas utama sistem dari sudut pandang pengguna. Use case diagram berfokus pada apa yang bisa dilakukan oleh sistem, misalnya: login, membuat akun, dll.

Class Diagram: Digunakan untuk menggambarkan struktur statis dari sistem, seperti kelas-kelas yang ada dalam sistem, atribut, metode, dan hubungan antar kelas. Diagram ini berfokus pada bagaimana sistem bekerja secara internal, dan digunakan dalam pemodelan Object-Oriented (OOP) untuk merancang struktur kelas-kelas dan relasinya dalam sistem.

2. Jelaskan jenis jenis dependensi?

Association: Hubungan yang menggambarkan keterkaitan antar objek. Misalnya, objek Mahasiswa memiliki asosiasi dengan objek Universitas.

Aggregation: Hubungan yang menunjukkan bahwa satu objek adalah bagian dari objek lain, tetapi bisa berdiri sendiri. Contoh: Mobil memiliki Mesin, tetapi Mesin bisa berdiri sendiri tanpa Mobil.

Composition: Hubungan yang lebih kuat dari aggregation, di mana objek bagian tidak bisa berdiri sendiri tanpa objek utamanya. Contoh: Rumah memiliki Ruang, dan jika Rumah dihapus, maka Ruang juga ikut terhapus.

Inheritance: Hubungan pewarisan di mana suatu kelas dapat mewarisi atribut dan metode dari kelas lain. Contoh: MobilBalap adalah subclass dari Mobil.

3. Apa perbedaan pemrograman terstruktur dengan berorientasi object, jelaskan!

Pemrograman Terstruktur: Berfokus pada fungsi atau prosedur, dengan cara berpikir yang linear dan top-down. Dalam pemrograman terstruktur, program dibagi menjadi modul-modul kecil (fungsi) untuk menyelesaikan masalah. Contoh bahasa: C.

Pemrograman Berorientasi Objek (OOP): Berfokus pada objek yang merupakan gabungan dari data dan fungsi dalam satu kesatuan. OOP menggunakan konsep seperti inheritance, encapsulation, dan polymorphism untuk memodelkan sistem. Bahasa berorientasi objek: Java, Python, C++.

4. Jelaskan konsep object dan beri contohnya!

Objek adalah entitas dari dunia nyata yang memiliki karakteristik (atribut) dan perilaku (metode/fungsi). Dalam OOP, objek adalah instance dari sebuah kelas.

Contoh:

Kelas Mobil memiliki atribut seperti warna, merk, dan kecepatan. Metode dari kelas Mobil dapat mencakup berjalan() dan berhenti(). Setiap mobil yang dibuat dari kelas Mobil adalah objek, misalnya Mobil merah dengan merk Toyota.

```
class Mobil {  
    String warna;  
    String merk;  
  
    void berjalan() {  
        System.out.println("Mobil berjalan");  
    }  
}
```

5. Jelaskan jenis jenis acces modifier beri contohnya dalam baris pemrograman?

Public: Atribut atau metode yang bisa diakses dari mana saja, baik di dalam kelas itu sendiri, kelas lain dalam paket yang sama, atau dari luar paket.

Contoh: class persegi_panjang:

```
def __init__(self, panjang, lebar):
    self.panjang = panjang
    self.lebar = lebar

def luas(self):
    return (self.panjang * self.lebar)
```

```
prsg = persegi_panjang(10, 5)
print(prsg.panjang)
```

Protected: Atribut atau metode yang bisa diakses oleh kelas di dalam paket yang sama dan subclass dari kelas tersebut.

Contoh:

class persegi_panjang:

```
def __init__(self, panjang, lebar):
    self.panjang = panjang
    self._lebar = lebar

def luas(self):
    return (self.panjang * self.lebar)
```

```
prsg = persegi_panjang(10, 5)
print(prsg.luas)
```

Private: Atribut atau metode hanya bisa diakses dari dalam kelas itu sendiri.

Contoh:

class Hewan:

```
def __init__(self, nama, gender):
    self.nama = nama
    self.__gender = gender

def tampilkan_hewan(self):
    return (f"nama: {self.nama}, gender: {self.gender}")
```

class darat(Hewan):

```
#atribut gender dari class Hewan tidak bisa diakses di sini karena atribut privat/
hanya bisa diakses di class nya sendiri
pass
```

```
kucing = Hewan("Kucing", "jantan")
kucing.tampilkan_hewan()
```

6. Gambarkan contoh pewarisan dalam diagram class!

