

**Nama : Aji Kharisma Atmaja**

**NPM : 5230411292**

**Mata Kuliah : Pemrograman Berorientasi Objek**

**Kelas : VIII**

**Responsi**

1. Jelaskan perbedaan use case diagram dengan class diagram?

Use case diagram adalah sebuah desain eksternal yang menggambarkan proses jalannya sebuah system yang dapat di mengerti oleh pengguna atau bukan pengembang dan biasanya use case diagram ini diperuntukkan untuk pemilik program atau owner yang ingin kita buat sistemnya sedangkan class diagram adalah struktur internal didalam program yang isinya ada kelas, atribut, method, relasi, yang biasanya diperuntukkan untuk pengembang atau pembuat program karena dapat membantu programmer dalam membuat dan menyelesaikan program.

2. Jelaskan jenis-jenis dependensi?

- Association (Asosiasi)

Adalah sebuah hubungan yang menghubungkan antar kelas yang ada pada program, untuk jumlah kelasnya bisa terdiri dari 2 atau lebih kelas yang berhubungan yang artinya kelas kelas tersebut dapat berinteraksi secara langsung. Contoh : Ada kelas Pembeli dan kelas order, kelas pembeli ini akan berhubungan dengan kelas order karena seorang pembeli dapat mengorder satu atau bahkan lebih.

- Agregation (Agregasi)

Hampir sama dengan asosiasi namun agregasi memiliki hubungan lebih khusus antar kelasnya dan dalam agregasi ini kelas atau bagian itu dapat berdiri sendiri tanpa ada keseluruhan. Contoh nya Kelas buku dan perpustakaan, kelas perpustakaan ini memiliki hubungan agregasi dengan kelas buku karena perpustakaan menyimpan banyak koleksi buku, namun kelas buku ini dapat berdiri sendiri tanpa adanya kelas perpustakaan.

- Composition (Komposisi)

Jenis lebih khusus lagi dari pada agregasi, karena memiliki hubungan-hubungan yang sangat kuat dan tidak dapat berdiri sendiri tanpa keseluruhan. Contohnya ada sebuah rumah yang didalamnya ada ruangan, nah ruangan itu tidak dapat berdiri sendiri tanpa adanya rumah.

- Dependency (Ketergantungan)

Artinya hubungan setiap kelas yang saling bergantung pada kelas lain. Suatu kelas yang harus menggunakan fungsi dari kelas lain karena definisi nya kelas ini saling bergantung sehingga tidak dapat berjalan sendiri tanpa adanya kelas lain. Contoh nya ada kelas pesan makanan dan kelas pembayaran, kelas pesan makanan sangat membutuhkan layanan pembayaran yang merupakan fungsi yang diambil dari kelas pembayaran.

- Realization (Realisasi)

Adalah hubungan antar kelas yang ada pada antarmuka (interface) serta pengimplementasiannya. Menunjukkan bahwa kelas menyediakan implementasi dari metode-metode yang didefinisikan pada antarmuka. Contoh Pada tampilan pembayaran itu diimplementasikan oleh kelas pembayaran dengan kartu kredit dan kelas paypalpayment atau saldo dari aplikasi atau program itu sendiri.

- Generalization (Pewarisan atau Inheritance)

Adalah dimana sebuah kelas atau objek mewariskan sifat dan perilaku kepada kelas lainnya atau hubungan kelas induk dan kelas turunan. Kelas turunan akan memiliki sifat atau perilaku yang sama pada kelas induk namun pada kelas induk belum tentu memiliki sifat atau perilaku yang sama dengan kelas turunan. Contoh Kelas hewan diwarisi oleh kelas burung dan kucing.

3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek, jelaskan?

Pemrograman terstruktur adalah pemrograman yang berguna untuk memecahkan masalah dengan prosedur yang runtut dan membangun program dengan serangkaian modul dan fungsi, pemrograman ini masih terbilang simpel. Sedangkan pemrograman berorientasi objek adalah pemrograman yang berbasis pada objek dan kelas yang menggabungkan data menjadi satu untuk dapat membuat aplikasi besar dan berorientasi pada pengembangan berkelanjutan, pemrograman ini dapat dikatakan lebih kompleks dari pada pemrograman terstruktur.

4. Jelaskan konsep objek dan berikan contohnya?

Objek adalah representasi dari entitas yang memiliki atribut dan perilaku. Objek ini berfungsi untuk media cetakan dan pada setiap objek memiliki keadaan dan bisa melakukan tindakan tertentu melalui metode yang dimilikinya didalam kelas. Contoh sederhana adalah sistem pembelian motor, ada kelas motor yang memiliki atribut warna merk seri dan kecepatan. Dari kelas ini kita dapat membuat objek motor ini berwarna merah, merk nya honda, serinya supra 125 dan kecepatannya 80km/jam. objek itu dapat memiliki metode berjalan dan berhenti yang mengekspresikan perilaku motor.

5. Jelaskan jenis-jenis access modifier beri contohnya dalam baris pemrograman?

- A. Public : hak akses public bisa diakses dari mana saja baik dari luar kelas maupun dari dalam kelas. Contoh dalam program : self.pemilik = pemilik
- B. Private : hanya bisa diakses pada kelasnya sendiri. Dalam python ditandai dengan 2 garis bawah (\_\_) diikuti nama atribut. Contoh dalam pemrograman = self.\_\_saldo = saldo\_awal.

Contoh lebih jelas:

```

class user():
    def __init__(self, username, password):
        self.username = username
        self.__password = password

    def get_password(self):
        return self.__password

    def set_password(self, password):
        self.__password = password

    def display(self):
        print(f"Username: {self.username} Password: {self.__password}")

user1 = user("Joko", "123")
print(user1._user__password) #CARA MANGGIL ATAU PRINT CLASS PRIVAT DILUAR KELAS TANPA PAKAI GET
user1.display()
print(user1.get_password()) #CARA PRINT PASSWORD YANG PRIVAT DI LUAR CLASS TANPA DISPLAY TAPI PAKAI GET

```

- C. Protected : Hanya bisa diakses secara terbatas oleh dirinya sendiri (yaitu didalam internal kelas), dan juga dari kelas turunannya. Dalam python ditandai dengan tanda 1 garis bawah (\_) diikuti nama atribut. Contoh dalam program = self.\_nomo\_rek = '123456789'

Contoh lebih jelas:

```

class hewan():
    def __init__(self, nama):
        self.nama = nama

class darat(hewan):
    def __init__(self, nama, kaki):
        super().__init__(nama)
        self._kaki = kaki

    def display(self):
        print (f"{self.nama} berkaki {self._kaki}")

class air(hewan):
    def __init__(self, nama, nafas):
        super().__init__(nama)
        self._nafas = nafas

    def display(self):
        print (f"{self.nama} bernafas dengan {self._nafas}")

kucing = darat("kucing", "4")
hiu = air("Hiu", "Insang")
kucing.display()
hiu.display()

```

6. Gambarkan contoh pewarisan dalam diagram class?

