

Nama : Muhammad Ilham Yunanto

NPM : 5230411277

Kelas : VIII

Matkul : Pemrograman Berorientasi Objek Praktik

Soal Teori

1. Jelaskan perbedaan use case diagram dengan class diagram!
2. Jelaskan jenis-jenis dependensi!
3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek, 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:

1. Use Case Diagram

Adalah jenis yang digunakan dalam pemodelan sistem berbasis objek untuk menggambarkan interaksi pengguna(aktor) dengan sistem yang sedang dikembangkan. Diagram ini bagian dari Unified Modeling Language (UML) dan digunakan untuk menggambarkan fungsionalitas sistem dari perspektif pengguna.

Notasi yang ada di Use Case Diagram:

1. Aktor
2. Use Case
3. Asosiasi Komunikasi
4. Subjek (subsystem)
5. Dependensi

Class Diagram

Adalah Jenis diagram struktur statis yang menggambarkan struktur sistem dengan menunjukkan sistem class, atributnya, metode, dan hubungan antar objek, sekelompok objek ini terdiri atas fitur struktural yang mendefinisikan apa yang diketahui class dan fitur operasional yang mendefinisikan apa yang bisa dilakukan oleh class. Fungsi utama dari class diagram adalah menggambarkan struktur sebuah sistem pemrograman.

Notasi yang ada di Class Diagram:

1. Kelas
2. Interface
3. Association
4. Directed Association
5. Generalisasi
6. Dependency
7. Agregation

2. Jenis-Jenis Dependency:

a) Include Dependency

Digunakan untuk menunjukkan bahwa suatu use case mengikutsertakan (include) dengan use case yang lain.

b) Extend Dependency

Digunakan use case untuk memperluas(extend) fungsionalitas use case lain dalam kondisi tertentu.

c) Usage Dependency

Digunakan untuk menunjukkan hubungan satu elemen menggunakan elemen lain untuk menjalankan fungsinya. Biasa digunakan untuk dalam class diagram untuk menunjukkan satu kelas menggunakan metode atau atribut dari kelas lain.

d) Realization Dependency

Digunakan untuk menunjukkan bahwa satu elemen direalisasikan oleh elemen lain.

e) Abstraction Dependency

Hubungan antar elemen yang mendefinisikan konsep abstrak dan elemen yang menerapkannya.

3. Perbedaan pemrograman terstruktur dengan berorientasi objek

a. Pemrograman Terstruktur

Adalah proses untuk mengimpletasikan urutan langkah untuk menyelesaikan suatu masalah dalam bentuk program. Pada saat menyelesaikan suatu masalah akan menjadi sub-masalah yang lebih kecil dan menyelesaikannya dengan prosedur atau fungsi.

b. Pemrograman Berorientasi Objek (OOP)

Adalah suatu pemrograman dimana program dapat distrukturkan dalam kumpulan objek. Kumpulan objek akan dibungkus dalam sebuah kelas, yang merupakan kerangka atau template untuk membuat objek

4. Jelaskan konsep objek dan beri contohnya

Objek adalah salah satu elemen utama pada pemrograman berorientasi objek. Objek menggambarkan entitas dari dunia nyata atau konsep tertentu yang memiliki atribut (data) dan perilaku (fungsi/metode). Objek dibuat berdasarkan kelas. **Contoh:** Menu pada rumah makan dapat memiliki atribut: harga, nama, deskripsi, dan memiliki fungsi untuk: ditambah, dihapus, dilihat, dll.

5. Jenis-jenis access modifier, beri contohnya dalam baris pemrograman!

- **Public :** Dapat diakses secara bebas dan dimana saja baik dalam kelas maupun dari luar kelas

```
• class Kendaraan:
•     def __init__(self, merk, kecepatan):
•         # Atribut public
•         self.merk = merk # Atribut ini dapat diakses dari mana saja
```

- **Protected:** Ditandai dengan satu underscore (_), hak akses protected hanya bisa diakses secara terbatas oleh dirinya sendiri (yaitu dalam internal kelas), dan juga dari kelas turunannya.

```
• class Kendaraan:
•     def __init__(self, merk, kecepatan):
•         # Atribut protected
•         self._kecepatan = kecepatan # Atribut ini sebaiknya hanya
diakses dalam kelas ini dan turunannya
```

- **Private:** Ditandai dengan dua underscore (__), hak akses hanya bisa diakses pada kelasnya sendiri.

```
• class Kendaraan:
•     def __init__(self, merk, kecepatan):
•         # Atribut private
•         self.__bensin = 100 # Atribut ini hanya bisa diakses dari
dalam kelas ini
```

6. Gambarkan contoh pewarisan dalam diagram class!

