RESPONSI 1
PEMROGRAMAN BERORIENTASI OBJEK PRAKTIK
KELAS VIII

MOH. SU'AIDI

5230411271

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024

Soal:

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 dan class diagram adalah dua jenis diagram yang sering digunakan dalam pemodelan sistem perangkat lunak. Use case diagram fokus pada interaksi antara pengguna (aktor) dengan sistem, menggambarkan fitur dan fungsionalitas apa saja yang disediakan oleh sistem. Diagram ini memberikan gambaran tingkat tinggi tentang apa yang dilakukan sistem.

   Class diagram, di sisi lain, menggambarkan struktur internal sistem, seperti kelas-kelas, atribut, dan hubungan antar kelas. Diagram ini memberikan gambaran rinci tentang objek-objek apa yang membentuk sistem dan bagaimana mereka saling berinteraksi.

   Ibaratnya, use case diagram menjawab pertanyaan "Apa yang dilakukan sistem?", sedangkan class diagram menjawab pertanyaan "Bagaimana sistem dibangun?".

2. Dependensi merupakan hubungan ketergantungan antara satu elemen dengan elemen lainnya dalam suatu sistem. Secara umum, dependensi dapat dikategorikan menjadi beberapa jenis:

   - Dependensi Fungsional: Menunjukkan hubungan sebab-akibat antara atribut dalam basis data.
   - Dependensi Transitif: Merupakan hubungan tidak langsung antara dua atribut melalui atribut ketiga.
   - Dependensi Parsial: Menunjukkan hubungan sebagian antara dua atribut.
   - Dependensi Logis: Berkaitan dengan hubungan antara konsep atau ide.
   - Dependensi Fisik: Berhubungan dengan ketergantungan pada sumber daya fisik.

3. Pemrograman terstruktur dan berorientasi objek adalah dua paradigma pemrograman yang berbeda. Pemrograman terstruktur menekankan pada pemecahan masalah dengan cara membagi program menjadi fungsi-fungsi yang lebih kecil, dengan aliran kontrol yang

jelas. Pemrograman berorientasi objek lebih fokus pada pemodelan dunia nyata dengan menggunakan konsep objek, kelas, atribut, dan metode.

Pemrograman terstruktur melihat program sebagai sekumpulan langkah-langkah yang harus dijalankan secara berurutan, sedangkan pemogragraman berorientasi objek melihat program sebagai kumpulan objek yang saling berinteraksi. Singkatnya, pemograman terstruktur lebih procedural, sementara pemograman berorientasi objek lebih konseptual.

4. Konsep objek dalam pemrograman berorientasi objek (OOP) adalah representasi dari entitas di dunia nyata yang memiliki karakteristik (atribut) dan perilaku (metode). Objek ini seperti cetakan kue, di mana setiap cetakan (kelas) dapat menghasilkan banyak kue (objek) yang serupa namun memiliki isi yang berbeda.

   Contohnya, mobil adalah sebuah objek. Atribut mobil bisa meliputi merek, warna, tahun pembuatan, dan kecepatan maksimum. Metode mobil bisa berupa menyalakan mesin, mengerem, atau berbelok. Jadi, setiap mobil yang kita lihat adalah sebuah objek dengan atribut dan metode yang spesifik.

5. Access modifier adalah atribut yang menentukan tingkat akses terhadap anggota (atribut atau metode) suatu kelas dalam pemrograman berorientasi objek. Python menyediakan tiga jenis access modifier utama:

   - Public: Anggota yang dideklarasikan sebagai public dapat diakses dari mana saja, baik dari dalam kelas itu sendiri, kelas turunan, maupun di luar kelas.
   - Protected: Anggota yang dideklarasikan sebagai protected hanya dapat diakses dari dalam kelas itu sendiri dan kelas turunannya.
   - Private: Anggota yang dideklarasikan sebagai private hanya dapat diakses dari dalam kelas itu sendiri.

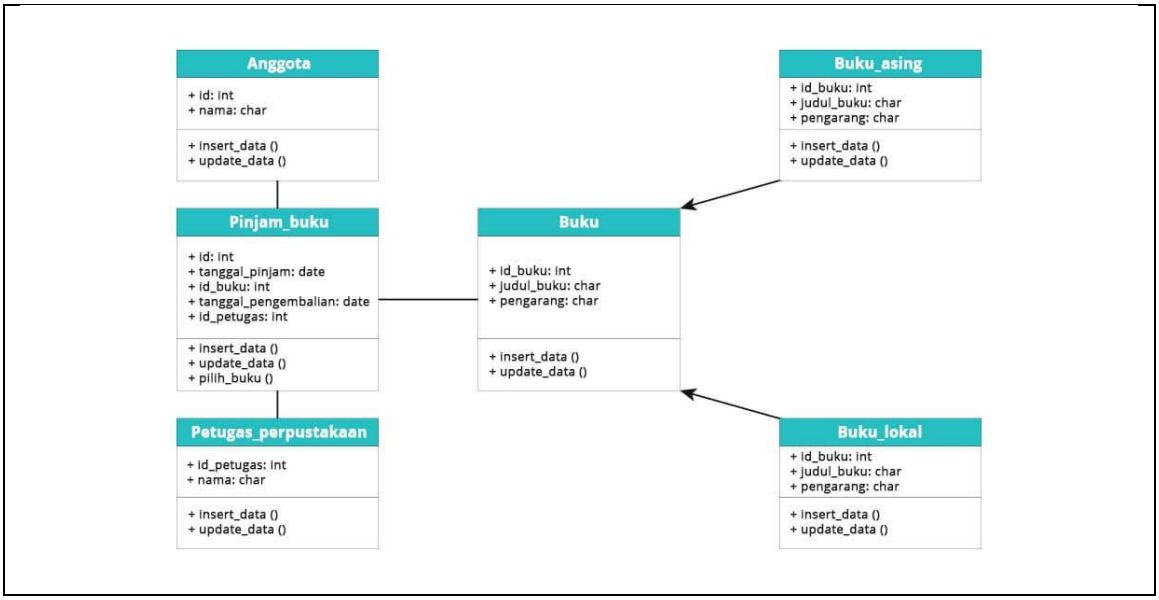
   Contoh:

```
class Mobil:
    def __init__(self, merek, warna, tahun):
        self.merek = merek  # public
        self._warna = warna  # protected
        self.__tahun = tahun  # private

    def get_tahun(self):  # public method untuk mengakses atribut
private
        return self.__tahun
```

- “merek” adalah atribut public, dapat diakses dari mana saja.
- “_warna” adalah atribut protected, hanya bisa diakses dari dalam kelas Mobil dan kelas turunannya.
- “__tahun” adalah atribut private, hanya bisa diakses dari dalam kelas Mobil itu sendiri. Metode “get_tahun()” digunakan untuk mengakses nilai “__tahun” dari luar kelas.

6. Diagram Class:



Penjelasan:

- Buku: Kelas ini bertindak sebagai kelas induk atau superclass. Kelas ini memiliki atribut dasar yang dimiliki oleh semua jenis buku, yaitu id_buku, judul_buku, dan pengarang.
- Buku_asing dan Buku_lokal: Kedua kelas ini merupakan kelas anak atau subclass dari kelas Buku. Ini berarti kelas Buku_asing dan Buku_lokal mewarisi semua atribut dan metode yang dimiliki oleh kelas Buku.