

**RESPONSI UTS PEMROGRAMAN
BERORIENTASI OBJEK PRAKTIK VIII**

T. A. Semester Ganjil 2024/2025



5230411285

OKTABERLIAN DIVA GRACIELA

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

2024

1. Jelaskan perbedaan Use Case Diagram dan Class Diagram?

Use Case Diagram berfungsi untuk menggambarkan interaksi antara pengguna atau aktor dengan sistem. Diagram ini menunjukkan apa saja yang bisa dilakukan oleh pengguna dalam sistem. Fokus utama dari use case diagram adalah menunjukkan perilaku dan kebutuhan yang diharapkan dari sistem berdasarkan peran tiap aktor.

Class Diagram, di sisi lain, lebih menekankan pada struktur dalam sistem itu sendiri. Diagram ini merepresentasikan kelas-kelas dalam sistem beserta atribut, metode, dan hubungan di antara kelas tersebut, seperti pewarisan (inheritance), asosiasi (association), atau ketergantungan (dependency). Dengan kata lain, class diagram berfokus pada desain internal dan detail teknis dari sistem.

2. Jelaskan jenis-jenis dependensi dalam pemrograman berorientasi objek? Dalam pemrograman berorientasi objek, ada beberapa jenis dependensi yang umum digunakan, yaitu:

- Association: Hubungan antara dua kelas yang bersifat tetap, di mana satu kelas mengenal kelas lain dan bisa berinteraksi dengannya.
- Aggregation: Hubungan "memiliki" atau "has-a" yang menunjukkan bahwa satu objek mengandung objek lain tetapi tidak mengendalikan siklus hidupnya.
- Composition: Mirip dengan agregasi, namun objek yang dimiliki sepenuhnya bergantung pada objek yang memilikinya, sehingga jika objek induk dihapus, objek anak juga akan ikut terhapus.
- Inheritance: Hubungan antara kelas induk dan kelas turunan, di mana kelas turunan mengambil sifat atau fitur dari kelas induknya.
- Dependency: Ketergantungan di mana satu kelas menggunakan fungsi atau properti dari kelas lain, biasanya bersifat sementara atau hanya selama fungsi tertentu berlangsung.

3. Apa perbedaan antara pemrograman terstruktur dan berorientasi objek?

Pemrograman Terstruktur adalah metode pemrograman yang fokus pada prosedur atau langkah-langkah untuk menyelesaikan masalah secara top-down. Setiap fungsi dan data didefinisikan terpisah. Biasanya cocok untuk program yang tidak terlalu kompleks.

Pemrograman Berorientasi Objek (OOP) menitikberatkan pada konsep objek yang memiliki atribut dan metode tertentu. Dengan prinsip seperti enkapsulasi, pewarisan, dan polimorfisme, OOP membantu dalam membuat

program yang lebih modular, sehingga lebih mudah dikelola dan dikembangkan, terutama untuk program yang kompleks.

4. Jelaskan konsep objek dan berikan contohnya?

Konsep Objek dalam pemrograman merujuk pada entitas yang merupakan instance dari suatu kelas. Objek memiliki data atau atribut tertentu dan fungsi atau perilaku. Misalnya, dalam kelas Mobil, objeknya bisa berupa Mobil A atau Mobil B yang memiliki atribut seperti warna dan merek, serta metode seperti jalan() untuk bergerak dan berhenti() untuk berhenti.

5. Jelaskan jenis-jenis access modifier dalam pemrograman berorientasi objek, beserta contohnya?

- Public: Akses terbuka untuk semua kelas. Contoh: public int data;
- Private: Hanya dapat diakses dalam kelas itu sendiri. Contoh: private int data;
- Protected: Dapat diakses oleh kelas turunan atau kelas dalam paket yang sama. Contoh: protected int data;
- Default (tanpa modifier): Hanya dapat diakses oleh kelas-kelas dalam paket yang sama. Contoh: int data;

6. Gambarkan contoh pewarisan dalam diagram class?

Contoh pewarisan dalam diagram class Contoh pewarisan dalam class diagram dapat digambarkan dengan memiliki kelas utama, misalnya Produk, yang menjadi superclass untuk kelas-kelas turunan seperti Makanan, Minuman, dan Snack. Kelas turunan ini akan mewarisi atribut dan metode dari Produk, namun bisa juga menambahkan fitur atau perilaku yang lebih spesifik sesuai dengan kebutuhan masing-masing jenis produk.

