

SISTEM OPERASI G

**Jawaban Responsi Pemrograman
Berorientasi Objek**



Disusun Oleh :

Fradeka Nur Choerun Kristiyanto 5230411294

**PROGRAM STUDI INFORMATIKA FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
TAHUN AJARAN 2024/2025**

1. - Use case diagram adalah jenis diagram yang digunakan dalam pemodelan sistem berbasis objek untuk menggambarkan interaksi antara pengguna (atau aktor) dengan sistem yang sedang di kembangkan. Tujuan use case diagram adalah untuk mendokumentasikan fungsi sistem dari perspektif pengguna, mendokumentasikan ruang lingkup sistem, mendokumentasikan interaksi antara pengguna dan sistem menggunakan deskripsi use case pendukung (spesifikasi perilaku).
 - Class diagram adalah jenis diagram struktur statis yang menggambarkan struktur sistem dengan menunjukkan sistem class, atributnya, metode dan hubungan antar objek. Fungsi utama dari class diagram adalah menggambarkan struktur sebuah sistem pemrograman.
 - Perbedaannya : Use Case Diagram fokus pada fungsi sistem dari perspektif user, sedangkan Class Diagram menunjukkan bagaimana sistem akan diimplementasikan. Jadi, Use Case Diagram membantu memahami apa yang sistem lakukan bagi penggunanya, sementara Class Diagram membantu memahami bagaimana sistem beroperasi untuk mencapai fungsi tersebut.
2. 1). Dependensi Fungsional
Dependensi fungsional menggambarkan relasi/hubungan, batasan, dan keterkaitan antara atribut-atribut dalam suatu relasi. Suatu atribut dikatakan bergantung pada atribut lain secara fungsional jika kita menggunakan harga atribut yang lain.
2). Dependensi Penuh
Dependensi penuh menunjukkan jika terdapat atribut A dan B dalam suatu relasi dimana :
 - B memiliki dependensi fungsional secara penuh pada A
 - B bukan memiliki dependensi terhadap subset A3). Dependensi Parsial
Dependensi Parsial merupakan ketergantungan fungsional dimana beberapa atribut dapat dihilangkan dari A dengan ketergantungan tetap dipertahankan.
 - B memiliki dependensi terhadap subset A4). Dependensi Transitif
Transitif dependensi adalah kondisi dimana A, B, C merupakan relasi dimana $A \Rightarrow B$ dan $B \Rightarrow C$.
 - maka C dikatakan dependensi transitif terhadap A melalui B.
3. ~ Pemrograman Terstruktur
Pemrograman terstruktur adalah pemrograman yang fokus pada alur logika yang menggunakan konsep prosedural. Program dibagi menjadi fungsi atau prosedur yang memiliki tugas tertentu.
~ Pemrograman Berorientasi Objek
Pemrograman berorientasi objek adalah pemrograman yang fokus pada objek. Objek adalah suatu benda/entitas yang memiliki atribut (data) dan metode (fungsi) yang menggambarkan keadaan objek tersebut.
~ Perbedaan
Pemrograman Terstruktur dan Pemrograman Berorientasi Objek memiliki fokus, organisasi kode, pembagian kode, reusability dan penggunaan bahasa yang berbeda satu sama lain. Secara umum, pemrograman terstruktur lebih cocok untuk program sederhana dan langsung, sementara pemrograman berbasis objek lebih cocok untuk proyek besar yang membutuhkan modularitas, pemeliharaan dan penggunaan ulang kode.

4. Konsep objek dalam pemrograman berorientasi objek (Object-Oriented Programming, OOP) adalah salah satu konsep dasar yang mewakili entitas dalam dunia nyata atau representasi data yang memiliki karakteristik dan perilaku tertentu. Dalam OOP, objek adalah unit utama yang menggabungkan data (disebut atribut atau properties) dan fungsi atau perilaku (disebut metode atau methods) dalam satu entitas.

Contoh :

- Mobil memiliki atribut:
 - ~ warna
 - ~ merk
 - ~ kecepatan_maksMethod nya:
 - ~ melaju
 - ~ rem
 - Manusia memiliki atribut :
 - ~ nama
 - ~ umur
 - ~ tinggi badanMethodnya :
 - ~ berjalan
 - ~ makan
 - Laptop memiliki atribut :
 - ~ merk
 - ~ spesifikasi
 - ~ beratMethodnya :
 - ~ menjalankan aplikasi
 - ~ mematikan laptop
5. Access Modifier adalah konsep dalam pemrograman berorientasi objek yang digunakan untuk menentukan aksesibilitas atau visibilitas anggota class, seperti atribut dan metode, dari luar kelas. Dalam banyak bahasa pemrograman, terdapat beberapa jenis access modifier, antara lain: public, protected, dan private.
- Public
Access modifier public memiliki hak akses paling luas dibanding lainnya. Karena aksesnya sangat luas, maka access modifier ini biasanya digunakan untuk method setter getter sesuai konsep OOP.
 - Protected
Access modifier protected biasanya digunakan untuk mewariskan variabel yang ada di super class terhadap child class.
 - Private
Access modifier private bersifat tertutup. Sesuai dengan konsep OOP, maka setiap variabel wajib untuk dilindungi hak aksesnya secara langsung dari luar. Oleh karena itu, variabel diberikan hak akses private dan untuk melakukan pengaksesan/perubahan data digunakan setter getter.

6.

