

**RESPONSI**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**KELAS VIII**

T. A. Semester Genap 2024/2025



**Disusun Oleh:**  
**Mochammad Adi Prawira (5230411329)**

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**2024**

1. **Use Case Diagram** digunakan pada saat ingin mengetahui apa interaksi pengguna dengan sebuah sistem yang akan dirancang bekeja nantinya, sedangkan **Class Diagram** adalah jenis diagram yang menggambarkan struktur sistem dengan menunjukkan sistem class, attribut, method, dan hubungan antara class dan objek.
2. - **Dependency Injection**  
Dependency Injection adalah pola di mana suatu objek menerima dependencies-nya dari luar, bukan membuatnya sendiri. Dependency ini biasanya di-inject melalui konstruktor (Constructor Injection), metode (Setter Injection), atau langsung melalui properti.
  - **Association**  
Association adalah jenis dependency yang menunjukkan hubungan antara dua kelas yang bekerja bersama. Dalam konteks ini, satu kelas menggunakan atau mengakses fungsi dari kelas lain.
  - **Composition**
  - Composition adalah hubungan yang kuat antara dua kelas. Dalam *composition*, kelas "parent" (induk) bertanggung jawab atas siklus hidup kelas "child" (anak).
  - **Inheritance (Pewarisan)**  
Inheritance memungkinkan satu kelas (subkelas) untuk mewarisi atribut dan metode dari kelas lainnya (superclass). Subkelas bergantung pada superclass untuk mewarisi sifat-sifat tertentu.
  - **Interface Implementation**  
Class yang mengimplementasikan suatu interface juga memiliki dependency terhadap interface tersebut, karena harus memenuhi kontrak dari metode atau atribut yang didefinisikan oleh interface.
3. **Pemrograman yang terstruktur** adalah Pemrograman terstruktur memecahkan masalah dengan prosedur dan tata cara yang teratur. Cenderung lebih fokus pada cara pemodelan data dan fungsi atau prosedur prosedur yang harus dibuat. Sedangkan Pemrograman Berorientasi Objek yaitu memecahkan masalah dengan melihat objek apa yang bekerja. Program dikembangkan menggunakan serangkaian objek dan interaksinya.
4. **Konsep objek** merupakan bagian dari Pemrograman Berorientasi Objek (OOP). Objek adalah instanse dari kelas, yaitu blueprint atau template dari objek. Contohnya seperti Class, Object, Encapsulation, Inheritance, dan Polymorphism.
5. -**Public**  
Memberikan akses kepada kelas untuk diakses oleh kelas manapun. Misalnya, jika kita memiliki kelas "Mobil" dengan properti "warna" yang diatur sebagai public, kita dapat mengaksesnya dari kelas lain tanpa batasan.  
-**Protected**  
Memberikan akses kepada kelas untuk diakses oleh kelas lain yang berada dalam satu paket dan class turunan.  
-**Default**  
Memberikan akses kepada kelas untuk diakses oleh kelas lain yang berada dalam satu paket. Namun anggota Default tidak dapat diakses dari kelas di luar paket.

### -Private

Memberikan akses kepada kelas untuk diakses oleh class itu sendiri. Artinya, properti atau metode tidak bisa diakses dari luar kelas Private.

## 6. Contoh Pewarisan

