

RESPONSI UTS PEMROGRAMAN BERIORENTASI

OBJEK PRAKTIK VIII

T.A Semester Ganjil 2024/2025



Najwa Shafira Zahra
(5230411293)

PROGRAM STUDI INFORMATIKA

**FAKULTAS SAINS DAN TEKNOLOGI UNIVERSITAS TEKNOLOGI
YOGYAKARTA**

2024

1. Jelaskan perbedaan use case diagram dengan class diagram

Jawab :

- **Use Case Diagram:**

Menunjukkan apa yang bisa dilakukan oleh sistem dari sudut pandang pengguna. Fokusnya adalah interaksi antara pengguna (aktor) dan layanan atau aktivitas (use case) yang tersedia di sistem.

- **Class Diagram:**

Menunjukkan struktur internal sistem, seperti "blueprint" yang berisi kelas-kelas, atribut, dan metode. Fokusnya adalah susunan dan hubungan antar bagian (kelas) dalam sistem.

2. Jelaskan jenis jenis dependensi

Jawab :

1. **Usage Dependency:**

Satu elemen membutuhkan elemen lain untuk berfungsi. Contoh: kelas A menggunakan kelas B untuk menjalankan metode tertentu.

2. **Include Dependency:**

Dalam *use case*, suatu proses membutuhkan proses lain sebagai bagian yang wajib. Contoh: pembayaran membutuhkan verifikasi.

3. **Extend Dependency:**

Dalam *use case*, suatu proses memiliki proses tambahan opsional. Contoh: setelah pemesanan barang selesai, ada *use case* opsional untuk mengirim notifikasi.

4. **Realization Dependency:**

Kelas mengimplementasikan antarmuka tertentu. Contoh: kelas A mengimplementasikan antarmuka B.

5. **Association Dependency:**

Dua elemen terhubung sebagai kolaborator. Contoh: kelas *Order* terhubung dengan kelas *Customer*.

6. **Instantiation Dependency:**

Satu elemen membuat objek dari elemen lain. Contoh: *OrderFactory* membuat objek *Order*.

7. **Permission Dependency:**

Satu elemen diberi akses ke elemen lain. Contoh: modul A memiliki izin untuk menggunakan modul B.

Dependensi ini menunjukkan bagaimana elemen-elemen dalam sistem saling bergantung.

3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek, jelaskan

Pemrograman Terstruktur

- **Cara Kerja:** Pemrograman ini berfokus pada langkah-langkah atau prosedur untuk menyelesaikan tugas. Program dibagi menjadi fungsi-fungsi kecil yang mengerjakan tugas tertentu.
- **Data dan Fungsi Terpisah:** Data dan fungsi tidak digabungkan dalam satu tempat; data disimpan sendiri, dan fungsi bertugas untuk memproses data itu.
- **Cocok Untuk:** Program yang sederhana, yang tidak membutuhkan banyak pengelompokan data.
- **Contoh Bahasa:** C dan Pascal.

Pemrograman Berorientasi Objek (OOP)

- **Cara Kerja:** Berfokus pada objek, yang artinya data dan fungsi disatukan dalam satu wadah atau kelas. Jadi, dalam satu kelas kita punya data dan juga fungsi-fungsi yang bisa mengolah data tersebut.
- **Pengelompokan Data dan Fungsi:** Data dan fungsi yang berkaitan dikumpulkan dalam objek, jadi lebih terorganisir.
- **Cocok Untuk:** Program yang lebih kompleks dan besar, karena lebih mudah untuk diatur dan dikembangkan.
- **Contoh Bahasa:** Java, Python, dan C++.

4. jelaskan konsep objek dan beri contohnya

Jawab :

Objek bisa diibaratkan seperti benda di dunia nyata. Misalnya, "mobil" adalah sebuah objek. Mobil memiliki data seperti warna, merek, dan kecepatan (atribut), serta tindakan seperti berjalan, berhenti, dan mempercepat (metode).

Dalam kode, objek adalah "versi" nyata dari kelas. **Kelas** adalah template atau cetakan, sedangkan **objek** adalah hasil nyata dari cetakan tersebut.

5. jelaskan jenis jenis acces modifier beri contohnya dalam baris pemrograman

Jawab :

- **Public**

Akses: Bisa diakses dari mana saja, baik dari dalam kelas, kelas lain, atau di luar kelas.

Contoh Penggunaan: Biasanya digunakan untuk metode atau atribut yang perlu diakses secara luas.

python

- **Protected**

Akses: Bisa diakses dalam kelas dan oleh kelas turunan (subclass), tetapi tidak disarankan untuk diakses di luar kelas.

Penulisan: Atribut atau metode protected diawali dengan satu garis bawah (_) di depan nama.

Contoh Penggunaan: Digunakan untuk data atau metode yang perlu dijaga tetapi masih perlu diakses oleh kelas turunan.

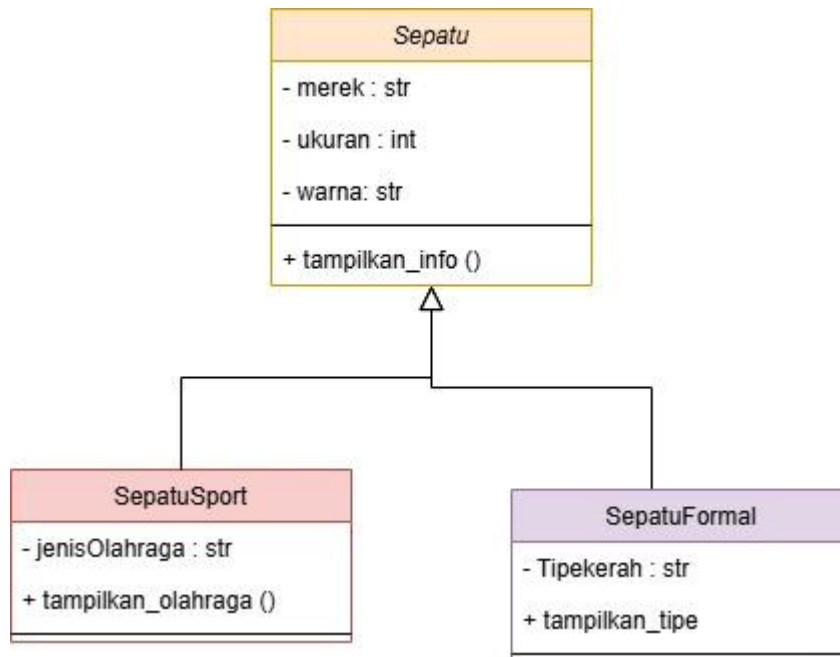
- **Private**

Akses: Hanya bisa diakses di dalam kelas itu sendiri, tidak bisa diakses oleh kelas turunan atau dari luar kelas.

Penulisan: Atribut atau metode private diawali dengan dua garis bawah (__) di depan nama.

Contoh Penggunaan: Digunakan untuk data atau metode yang benar-benar harus tersembunyi dan tidak boleh diakses dari luar.

6. Gambarkan contoh pewarisan dalam diagram class?



Penjelasan Diagram Kelas

1. Kelas Induk (Sepatu):

- Sepatu adalah kelas induk yang memiliki atribut umum untuk semua jenis sepatu, seperti:
 - merek: Merek sepatu (contoh: Nike, Adidas).
 - ukuran: Ukuran sepatu (contoh: 39, 40).
 - warna: Warna sepatu (contoh: merah, biru).
- Sepatu juga memiliki metode bernama `tampilkan_info()`, yang dapat menampilkan informasi dasar tentang sepatu.

2. Kelas Anak (SepatuSport):

- SepatuSport mewarisi atribut dan metode dari kelas Sepatu.
- Atribut tambahan yang dimiliki SepatuSport adalah `jenisOlahraga`, yang menunjukkan jenis olahraga yang cocok untuk sepatu tersebut (contoh: lari, basket).
- Metode baru `tampilkan_olahraga()` untuk menampilkan jenis olahraga yang dapat dilakukan dengan sepatu ini.

3. Kelas Anak (SepatuFormal):

- SepatuFormal juga mewarisi atribut dan metode dari kelas Sepatu.
- Atribut tambahan yang dimiliki SepatuFormal adalah tipeKerah, yang menunjukkan tipe kerah dari sepatu formal tersebut (contoh: kerah rendah, kerah tinggi).
- Metode baru tampilkan_tipe() untuk menampilkan informasi tentang tipe kerah sepatu formal.

Inti Pewarisan

Dalam diagram ini, kelas SepatuSport dan SepatuFormal mewarisi atribut dan metode dari kelas Sepatu, tetapi masing-masing juga menambahkan atribut dan metode spesifik yang sesuai dengan jenis sepatu tersebut. Ini menggambarkan bagaimana pewarisan memungkinkan pengorganisasian dan penggunaan kembali kode dalam pemrograman berorientasi objek.