

**RESPONSI UTS
PEMPROGRAMAN BERORIENTASI OBJEK
KELAS VII**

T. A. Semester Ganjil 2024/2025



5230411286

FIRMANTI ALHILMA S.

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

SOAL TEORI (POINT 50)

1. Jelaskan perbedaan use case diagram dengan class diagram?

Jawaban :

Use case diagram menunjukkan interaksi antara pengguna atau sistem lain dengan sistem, dengan fokus pada layanan atau fungsi yang disediakan. Elemen utamanya adalah aktor (pengguna) dan use case (fungsi sistem) yang dihubungkan dengan garis. Visualisasinya menggunakan bentuk oval untuk use case dan stick figure untuk aktor, memberikan gambaran umum tentang apa yang diinginkan pengguna dari sistem. Sementara itu, class diagram menggambarkan struktur internal sistem, termasuk kelas-kelas, atribut, metode, dan hubungan antar kelas. Elemen utamanya adalah kotak untuk tiap kelas yang memuat bagian atribut dan metode, yang menampilkan detail teknis bagaimana sistem dibangun. Jadi, use case diagram berfokus pada apa yang dilakukan sistem untuk pengguna, sedangkan class diagram berfokus pada bagaimana sistem tersebut disusun secara internal.

2. Jelaskan jenis-jenis dependensi?

Jawaban :

Jenis-jenis dependensi dalam UML yaitu:

- Usage : Menunjukkan suatu elemen menggunakan elemen lain, seperti pemanggilan metode.
- Abstraction : Menunjukkan hubungan antara elemen umum dengan elemen spesifik, misalnya antarmuka.
- Permission : Mengatur izin akses terbatas dari satu elemen ke elemen lain.
- Instantiation : Satu elemen membuat instans dari elemen lain.
- Trace : Melacak asal-usul antara elemen-elemen dari fase pengembangan yang berbeda.
- Refinement : Memurnikan elemen lain, biasanya dari tingkat abstraksi yang lebih tinggi.

3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek. Jelaskan?

Jawaban :

Pemrograman terstruktur fokus pada fungsi dan prosedur untuk menyelesaikan tugas dengan membagi program menjadi blok-blok fungsi. Data dan fungsi terpisah, jadi data bisa diakses dari mana saja di program. Sementara itu, pemrograman berorientasi objek (OOP) menggunakan kelas dan objek untuk merepresentasikan entitas dunia nyata. Data dan fungsi terkait digabungkan dalam satu unit (objek), dan OOP punya fitur khas seperti enkapsulasi (menyembunyikan data), pewarisan (kelas bisa mewarisi sifat kelas lain), dan polimorfisme (metode yang bisa berperilaku berbeda tergantung konteks).

4. Jelaskan konsep objek dan beri contohnya?

Jawaban :

Konsep objek dalam pemrograman merujuk pada entitas yang memiliki sifat (atribut) dan perilaku (metode). Objek adalah instansiasi dari kelas, yang merupakan blueprint atau template untuk menciptakan objek.

Contohnya dalam pemrograman :

```
class Pasma:
    def __init__(self, warna, bahan, ukuran):
        self.warna = warna      # Atribut: warna
        self.bahan = bahan      # Atribut: bahan
        self.ukuran = ukuran    # Atribut: ukuran

    def deskripsi(self):
        return f"Pasma ini berwarna {self.warna}, terbuat dari {self.bahan}, dan berukuran {self.ukuran}."

daftar_pasma = []

while True:
    print("\nMenu:")
    print("1. Tambah Pasma")
    print("2. Lihat Daftar Pasma")
    print("3. Keluar")

    pilihan = input("Pilih opsi (1/2/3): ")

    if pilihan == '1':
        # Meminta input dari pengguna
        warna = input("Masukkan warna pasma: ")
        bahan = input("Masukkan bahan pasma: ")
        ukuran = input("Masukkan ukuran pasma: ")

        # Membuat objek Pasma dan menambahkannya ke daftar
```

```

    pasmina_baru = Pasmina(warna, bahan, ukuran)
    daftar_pasmina.append(pasmina_baru)
    print("Pasmina berhasil ditambahkan!")

elif pilihan == '2':
    if not daftar_pasmina:
        print("Belum ada pasmina yang ditambahkan.")
    else:
        print("\nDaftar Pasmina:")
        for idx, pasmina in enumerate(daftar_pasmina, start=1):
            print(f'{idx}. {pasmina.deskripsi()}')

elif pilihan == '3':
    print("Terima kasih! Program selesai.")
    break

else:
    print("Pilihan tidak valid. Silakan coba lagi.")

```

5. Jelaskan jenis jenis access modifier beri contohnya dalam baris pemrograman?

Jawaban :

Access modifier adalah pengaturan akses untuk atribut dan metode dalam kelas guna mengontrol visibilitasnya dari luar kelas. Ada tiga jenis utama yaitu :

- Public : Anggota kelas yang bersifat public bisa diakses dari mana saja, baik dari dalam maupun luar kelas.
- Private : Anggota private hanya bisa diakses dari dalam kelas itu sendiri, menjaga agar data tidak diubah langsung dari luar.
- Protected : Anggota protected bisa diakses oleh kelas itu sendiri, kelas turunan (subclass), dan kelas dalam paket yang sama, tapi tidak dari luar.

Contoh dalam baris pemrograman nya :

```

class Kerudung:
    def __init__(self, warna, jenis, harga):
        self.__warna = warna # Private
        self._jenis = jenis # Protected
        self.harga = harga # Public

    def get_warna(self):
        return self.__warna # Mengakses warna melalui metode

```

```

def tampil_info(self):
    print(f'Warna: {self.get_warna()}')
    print(f'Jenis: {self._jenis}')
    print(f'Harga: {self.harga}')

class KerudungPremium(Kerudung):
    def __init__(self, warna, jenis, harga, bahan):
        super().__init__(warna, jenis, harga)
        self.bahan = bahan # Public

    def tampil_info_premium(self):
        self.tampil_info() # Mengakses metode public dari
        kelas induk
        print(f'Bahan: {self.bahan}')

def main():
    daftar_kerudung = []

    while True:
        print("\nMenu:")
        print("1. Tambah Kerudung Biasa")
        print("2. Tambah Kerudung Premium")
        print("3. Tampilkan Semua Kerudung")
        print("4. Keluar")

        pilihan = input("Pilih menu (1/2/3/4): ")

        if pilihan == '1':
            warna = input("Masukkan warna kerudung: ")
            jenis = input("Masukkan jenis kerudung: ")
            harga = int(input("Masukkan harga kerudung: "))
            kerudung = Kerudung(warna, jenis, harga)
            daftar_kerudung.append(kerudung)
            print("Kerudung biasa berhasil ditambahkan.")

        elif pilihan == '2':
            warna = input("Masukkan warna kerudung
            premium: ")
            jenis = input("Masukkan jenis kerudung premium:
            ")
            harga = int(input("Masukkan harga kerudung
            premium: "))
            bahan = input("Masukkan bahan kerudung
            premium: ")

```

```
        kerudung_premium = KerudungPremium(warna,
jenis, harga, bahan)
        daftar_kerudung.append(kerudung_premium)
        print("Kerudung premium berhasil ditambahkan.")

    elif pilihan == '3':
        print("\nDaftar Kerudung:")
        for kerudung in daftar_kerudung:
            kerudung.tampil_info()
            if isinstance(kerudung, KerudungPremium):
                print(f'Bahan: {kerudung.bahan}')
            print("---")

    elif pilihan == '4':
        print("Terimakasih telah menggunakan program
ini.")
        break # Menghentikan loop

    else:
        print("Pilihan tidak valid. Silakan coba lagi.")

if __name__ == "__main__":
    main()
```

6. Gambarkan contoh pewarisan dalam diagram class?

Jawaban :



