

TUGAS
PEMOGRAMAN BERORIENTASI OBJEK PRAKTIK



DISUSUN

OLEH

Daffa Nayottama

5230411234

Prodi : Informatika – E

UNIVERSITAS TEKNOLOGI YOGYAKARTA

FAKULTAS SAINS & TEKNOLOGI

Tahun Akademik 2023

1. Jelaskan perbedaan use case diagram dengan class diagram?
 - Use Case Diagram menggambarkan interaksi antara pengguna dan sistem. Diagram ini biasanya berfokus pada fungsi yang disediakan sistem dari sudut pandang pengguna. Use case diagram biasanya menjelaskan apa saja fitur utama sistem dan siapa saja yang menggunakannya.
 - Class Diagram menunjukkan struktur statis dari sistem, termasuk kelas-kelas yang ada beserta atribut dan metodenya. Diagram ini menggambarkan ikatan antara kelas-kelas yang ada dalam sistem dan lebih berfokus pada desain struktur data serta interaksi antara objek.
2. Jelaskan jenis-jenis dependensi?
 - Association: Hubungan antara dua kelas yang menunjukkan keterkaitan antara satu sama dengan yang lain.
 - Aggregation: Jenis asosiasi di mana satu kelas berfungsi sebagai bagian dari kelas lain, tetapi bisa berdiri sendiri tanpa adanya hubungan antara kelas lain.
 - Composition: Hubungan di mana satu kelas tidak dapat berdiri sendiri tanpa kelas yang lain.
 - Inheritance: Kelas mewarisi atribut dan metode dari kelas induknya sehingga bisa memakai atribut dari kelas lain.
 - Dependency: Ketergantungan satu kelas pada kelas lain, biasanya terjadi jika suatu kelas membutuhkan layanan dari kelas lain.
3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek? Jelaskan?
 - Pemrograman Terstruktur: Berfokus pada prosedur dan fungsinya. Program dipecah menjadi beberapa sub-program yang lebih kecil/child seperti fungsi atau prosedur.
 - Pemrograman Berorientasi Objek: Berfokus pada objek yang mewakili data dan metode untuk mengelola data tersebut. Pendekatan ini berpusat pada konsep kelas dan objek, dan menerapkan konsep pewarisan, enkapsulasi, dan polimorfisme. Contoh bahasa pemrograman berorientasi objek adalah Java dan Python.
4. Jelaskan konsep objek dan beri contohnya?
 - Objek adalah instance dari kelas yang mewakili entitas nyata atau abstrak dalam pemrograman berorientasi objek. Objek memiliki atribut (data) dan metode (fungsi) yang menentukan bagaimana sikapnya. Misalnya, pada sistem perpustakaan, buku bisa menjadi objek. Atribut objek buku mungkin bisa kita permisalkan berisi judul, penulis, dan tahun terbit, sedangkan metode yang dapat dimiliki termasuk pinjam dan kembalikan.

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

- Public: Atribut atau metode yang tanpa tanda khusus dianggap public dan dapat diakses dari luar kelas diberi lambang + kalau di uml.

Contoh:

```
class Person:
    def __init__(self, name):
        self.name = name
```

```
person = Person("Daffa")
print(person.name)
```

- Protected: Di Python, atribut yang diawali dengan satu garis di bawah seperti “_nik” dianggap protected dan hanya disarankan untuk diakses di dalam kelas dan subclass-nya di berilambang #.

Contoh :

```
class Person:
    def __init__(self, name, _nik):
        self._nik = _nik
```

```
person = Person("Daffa", 19)
print(person._age)
```

- Private: Atribut yang diawali dengan dua garis bawah “__nik” dianggap private dan tidak bisa diakses langsung dari luar kelas di beri lambing - .

Contoh:

```
class Person:
    def __init__(self, name, __age):
        self.__age = __age
```

```
    def get_age(self):
        return self.__age
```

```
person = Person("Daffa", 19)
print(person.get__age())
print(person.__age)
```