

Nama : Risang Haryo Pamungkas Bimani Sakti

NPM : 5230411290

Kelas : XIII

Mata Kuliah : Pemrograman Berbasis Objek

Soal Teori :

1. Jelaskan perbedaan use case diagram dengan class diagram!
2. Jelaskan jenis-jenis dependensi!
3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek, jelaskan!
4. Jelaskan konsep objek dan beri contohnya!
5. Jelaskan jenis-jenis access modifier beri contohnya dalam baris pemrograman!
6. Gambarkan contoh pewarisan dalam class diagram!

Jawab :

1. Use case diagram berfokus pada gambaran fungsi dan interaksi dari pengguna terhadap sistem, sedangkan Class diagram berfokus pada gambaran struktur class dan hubungan antar objek.
2. **Dependency Inversion**, Dependency Inversion adalah mengubah ketergantungan suatu class ke suatu abstraksi.  
**Dependency Injection**, Dependency Injection adalah membuat satu class baru yang bertanggung jawab untuk menginstansi class-class tersebut dan memberikannya ke class-class lain yang membutuhkan dalam waktu runtime.  
**Association**, Association adalah objek yang independen tetapi berelasi terhadap objek lain dan tanpa saling memiliki.  
**Aggregation**, Aggregation adalah objek yang memiliki objek yang lain tetapi objek lain tersebut masih bisa berdiri sendiri tanpa adanya objek pemilik objek ini.  
**Composititon**, Composition adalah objek yang sepenuhnya memiliki objek lain, jadi jika objek pemilik tidak ada maka objek lain tidak bisa berdiri sendiri.
3. Pemrograman terstruktur artinya program akan berjalan secara sequential atau dengan kata lain program akan dijalankan secara berurutan dari A-Z atau Top-Down, sedangkan Pemrograman Berorientasi Objek artinya program akan dijalankan berdasarkan class yang dibangun di dalam program untuk dibuat sistem yang lebih kompleks lagi yang berarti program akan dijalankan secara Bottom-Top.
4. Objek adalah sebuah entitas yang memiliki beberapa attribute dan method selayaknya di kehidupan nyata, seperti manusia yang memiliki attribute nama, tanggal lahir, dan sebagainya serta memiliki method seperti berjalan, berlari, makan, minum, tidur, dan sebagainya. Dengan kata lain objek adalah hasil bentukan dari cetak biru yang bernama class supaya objek tersebut dapat dimodifikasi, dipasangkan, diatur, dll dengan objek lain.

5. **Private**, Private adalah jenis access modifier yang berguna untuk membatasi attribute tertentu di suatu class supaya tidak bisa digunakan oleh class lain maupun class turunannya dan hanya bisa digunakan oleh class itu sendiri.

**Protected**, Protected adalah jenis access modifier yang berguna untuk membatasi attribute tertentu di suatu class supaya hanya bisa digunakan oleh class itu sendiri dan class turunannya saja

**Public**, Public adalah jenis access modifier yang berguna untuk memberikan akses attribute tertentu pada suatu class supaya dapat digunakan oleh class manapun.

**Contoh Kode :**

```
class Account:
    def __init__(self):
        self.email = None # Public
        self.__password = None # Private
        self._captcha = None # Protected

    def setEmail(self, email):
        self.email = email

    def setPassword(self, password):
        self.__password = password

    def setCaptcha(self, captcha):
        self._captcha = captcha

    def getPassword(self):
        return self.__password

    def displayAcc(self):
        print(f"Email : {self.email} | Password : {self.__password} | Captcha : {self._captcha}")

Acc1 = Account()
Acc1.setEmail('trombol@mail.com')
Acc1.setPassword('helcurut')
Acc1.setCaptcha('aTuf56')
Acc1.displayAcc()
```

6.

