Nama: Firmansyah hasibuan

NPM : 5230411283

MK : PBO

Soal Teori:

1. Jelaskan perbedaan use case diagram dengan class diagram?
2. Jelaskan jenis-jenis dependensis?
3. Apa perbedaan pemrograman terstruktur dengan berorientasi objek, jelaskan?
4. Jelaskan konsep objek dan beri contohnya?
5. Jelaskan jenis-jenis access modifiet beri contohnya dalam baris program?
6. Gambarkan contoh pewaris dalam diagram class?

Jawaban:

1. **Use Case Diagram** adalah jenis diagram yang digunakan dalam pemodelan sistem berbasis objek untuk menggambarkan interaksi antara pengguna (atau aktor) dengan sistem yang sedang dikembangkan.

   Class diagram adalah salah satu diagram UML yang memperlihatkan struktur sistem seperti kelas, atribut, metode, dan hubungan antar kelas.

2. **Dependensi Fungsional** Ketika satu atribut (atau kumpulan atribut) dalam suatu tabel menentukan atribut lain. Misalnya, jika kita memiliki tabel dengan kolom A dan B, dan untuk setiap nilai A yang unik.

   Dependensi Transitif Ini terjadi ketika ada dependensi fungsional tidak langsung. Jika A $\rightarrow$ B dan B $\rightarrow$ C, maka A $\rightarrow$ C adalah dependensi transitif.

   Dependensi Multivalued Terjadi ketika suatu atribut dapat memiliki lebih dari satu nilai untuk satu nilai dari atribut lain.

   Dependensi Kandidat Ini adalah subset dari atribut yang dapat digunakan untuk secara unik mengidentifikasi suatu entitas dalam tabel.

   Dependensi Partial Terjadi ketika atribut non-kunci bergantung pada hanya sebagian dari kunci utama. Misalnya, jika kunci utama terdiri dari dua atribut (A, B), dan ada atribut C yang hanya bergantung pada A.

   Dependensi Bertentangan Ini terjadi ketika ada konflik antara dua atau lebih dependensi, sering kali dalam konteks yang lebih kompleks seperti dalam sistem yang sangat dinamis.

3. **Pemrograman Terstruktur** paradigma pemrograman yang berfokus pada penggunaan struktur kontrol yang jelas dan terorganisisr untuk menulis program yang efisien dan mudah dipahami.

Pemrograman berorientasi objek paradigma dari pemrograman yang menggunakan "objek" dan "kelas" sebagai dasar untuk membangun perangkat lunak.

4. **Konsep objek** adalah inti dari pemrograman berorientasi objek (OOP), yang merupakan paradigma pemrograman yang mengorganisir kode dalam bentuk objek.

   Kelas Kelas adalah cetak biru atau template untuk membuat objek. Kelas mendefinisikan atribut dan metode yang akan dimiliki oleh objek tersebut.

   Objek Objek adalah instansi konkret dari kelas. Setiap objek memiliki keadaan (state) yang ditentukan oleh nilai-nilai atributnya.

   Atribut Atribut adalah variabel yang menyimpan data dalam objek. Mereka menggambarkan keadaan objek.

   Metode Metode adalah fungsi yang didefinisikan dalam kelas dan dapat melakukan operasi pada objek atau mengubah atributnya.

5. **Access Modifiet:**
   - **Public** Atribut atau metode yang dideklarasikan sebagai public dapat diakses dari mana saja, baik dari dalam kelas itu sendiri, dari kelas lain di paket yang sama, maupun dari kelas di paket yang berbeda.

```
public class Mobil { public String warna; public void tampilkanWarna() { System.out.println("Warna mobil: " + warna); } }
```

   - **Private** Anggota kelas yang dideklarasikan sebagai `private` hanya dapat diakses dari dalam kelas itu sendiri. Ini membantu menjaga enkapsulasi.

```
public class Akun { private String password; public void setPassword(String password) { this.password = password; } public String getPassword() { return password; // Mengakses password melalui metode publik } }
```

   - **Protected** Anggota kelas yang dideklarasikan sebagai `protected` dapat diakses oleh kelas itu sendiri, kelas turunan (subclass), dan kelas dalam paket yang sama.

```
public class Hewan { protected String jenis; protected void suara() { System.out.println("Suara hewan"); } } public class Kucing extends Hewan { public void suaraKucing() { suara(); // Mengakses metode protected dari superclass } }
```

- **Default (Package-Private)** Jika tidak ada modifier yang ditentukan, anggota kelas tersebut bersifat default, yang berarti hanya dapat diakses dalam paket yang sama.

```
class Buku { String judul; // Default access modifier void tampilkanJudul() { System.out.println("Judul buku: " + judul); } }
```

6. <u>Contoh gambar diagram class</u>