****

**Nama : Fitri Cahyaniati**

**Kelas : SIB2G**

**Nim : 2341760198**

**Latihan :**

* **Buat sebuah studi kasus dengan minimal 3 class yang saling berasosiasi.**
* **Buat minimal 1 atribut bertipe ArrayList of objects**
* **Tentukan atribut dan method pada masing-masing class**
* **Gambarkan class diagramnya.**

**Studi Kasus: Sistem Manajemen Kebun Tanaman**

1. **Deskripsi:**  
   Sistem ini digunakan untuk mengelola kebun tanaman dengan memantau dan merawat tanaman secara efisien. Terdapat tiga class utama: Kebun, Tanaman, dan Pekerja. Class Kebun berisi informasi tentang kebun dan daftar tanaman serta pekerja, class Tanaman berisi informasi tentang tanaman, dan class Pekerja berisi informasi tentang pekerja kebun yang merawat tanaman.

**Class 1: Kebun**

* **Atribut:**
  + namaKebun: String – Nama kebun
  + lokasi: String – Lokasi kebun
  + daftarTanaman: ArrayList<Tanaman> – Daftar tanaman yang ada di kebun
  + daftarPekerja: ArrayList<Pekerja> – Daftar pekerja kebun
* **Method:**
  + tambahTanaman(Tanaman tanaman): Menambahkan tanaman baru ke kebun
  + tambahPekerja(Pekerja pekerja): Menambahkan pekerja baru ke kebun
  + tugaskanPekerjaKeTanaman(Tanaman tanaman, Pekerja pekerja): Menugaskan pekerja untuk merawat tanaman tertentu

**Class 2: Tanaman**

* **Atribut:**
  + namaTanaman: String – Nama tanaman
  + jenisTanaman: String – Jenis tanaman (contoh: "Tomat", "Kangkung")
  + statusKesehatan: String – Status kesehatan tanaman (contoh: "Sehat", "Butuh Perawatan")
  + pekerja: Pekerja – Pekerja yang bertanggung jawab untuk merawat tanaman
* **Method:**
  + updateStatusKesehatan(String status): Mengupdate status kesehatan tanaman
  + setPekerja(Pekerja pekerja): Menetapkan pekerja yang merawat tanaman

**Class 3: Pekerja**

* **Atribut:**
  + namaPekerja: String – Nama pekerja
  + idPekerja: String – ID pekerja
  + daftarTanaman: ArrayList<Tanaman> – Daftar tanaman yang dirawat oleh pekerja
* **Method:**
  + tambahTanaman(Tanaman tanaman): Menambahkan tanaman ke daftar tanaman yang dirawat
  + hapusTanaman(Tanaman tanaman): Menghapus tanaman dari daftar tanaman yang dirawat

1. Class Diagram :

* Class kebun :

|  |
| --- |
| Kebun |
| - namaKebun: String  - lokasi: String  - daftarTanaman: ArrayList<Tanaman>  - daftarPekerja: ArrayList<Pekerja> |
| + tambahTanaman()  + tambahPekerja()  + tugaskanPekerjaKeTanaman() |

* Class Tanaman :

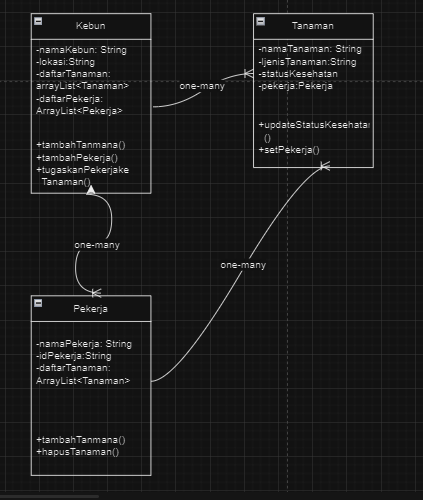
|  |
| --- |
| Tanaman |
| - namaTanaman: String  - jenisTanaman: String  - statusKesehatan: String  - pekerja: Pekerja |
| + updateStatusKesehatan()  + setPekerja() |

* Class Pekerja :

|  |
| --- |
| Pekerja |
| - namaPekerja: String  - idPekerja: String  - daftarTanaman: ArrayList<Tanaman> |
| + tambahTanaman()  + hapusTanaman() |

 **+ (Public)**: Metode/atribut bisa diakses dari luar kelas.

 **- (Private)**: Metode/atribut hanya bisa diakses dari dalam kelas.



1. Relasi :

* **Kebun -> Tanaman:**

Notasi: 1.. (One-to-Many)\*  
Setiap **Kebun** bisa memiliki banyak **Tanaman**, tetapi setiap **Tanaman** hanya dimiliki oleh satu **Kebun.**

Kebun: 1 Tanaman: 0..\*

* **Kebun -> Pekerja:**

Notasi: 1.. (One-to-Many)\*  
Setiap **Kebun** dapat memiliki banyak **Pekerja**, tetapi setiap **Pekerja** hanya bekerja di satu **Kebun**.

Kebun: 1 Pekerja: 0..\*

* Pekerja -> Tanaman :

Notasi: 1.. (One-to-Many)\*  
Setiap **Pekerja** bisa merawat banyak **Tanaman**, tetapi setiap **Tanaman** dirawat oleh satu **Pekerja.**

Pekerja: 1 Tanaman: 0..\*

1. Terdapat empat jenis relasi dasar antar kelas**: Inheritance, Association, Aggregation,** dan **Composition**. Berikut adalah penjelasan dan penerapannya:
2. **Inheritance (Pewarisan)**

* **Deskripsi**: Pewarisan digunakan ketika suatu kelas mengambil properti dan perilaku dari kelas lain.
* **Contoh pada sistem kebun**: Misalnya, jika kita memiliki kelas **Pekerja** yang merupakan superclass, kita dapat memiliki subclass seperti **PekerjaTetap** dan **PekerjaLepas**, di mana keduanya mewarisi atribut dan metode dari **Pekerja**.

### **Association (Asosiasi**)****

* **Deskripsi**: Asosiasi adalah relasi antar objek, di mana satu kelas berhubungan dengan kelas lainnya tanpa ada kepemilikan eksklusif.
* **Contoh pada sistem kebun**: **Kebun** berasosiasi dengan **Tanaman** dan **Pekerja.** Ini karena kebun memiliki daftar tanaman dan pekerja, namun tidak secara eksklusif "memiliki" objek-objek tersebut.

### **Aggregation (Agregasi)**

* **Deskripsi**: Agregasi adalah relasi yang lebih longgar dibandingkan komposisi, di mana satu kelas "memiliki" kelas lain, namun kelas yang dimiliki dapat tetap eksis tanpa harus bergantung pada kelas yang memilikinya.
* **Contoh pada sistem kebun: Kebun** mengagregasi **Tanaman** dan **Pekerja**, artinya meskipun **Kebun** dihapus, **Tanaman** dan **Pekerja** masih dapat ada secara mandiri di konteks lain.

### **Composition (Komposisi)**

* **Deskripsi**: Komposisi adalah hubungan kepemilikan yang lebih kuat, di mana jika objek yang "memiliki" dihapus, maka objek yang dimiliki juga dihapus.
* **Contoh pada sistem kebun:** Jika **Tanaman** memiliki bagian atau elemen internal, seperti **Akar** atau **Daun**, komposisi dapat digunakan. Jika **Tanaman** dihapus, maka bagian-bagian ini juga dihapus.