



Nama : Fitri Cahyaniati

Kelas : SIB2G

ID : 2341760198

Latihan :

- Untuk studi kasus A dengan minimal 3 kelas yang rekan yang saling berhubungan.
- Buat setidaknya 1 tipe atribut ArrayList dari objek
- Tentukan atribut dan metode di setiap kelas
- Gambarkan diagram kelas.

Studi Kasus : Sistem Manajemen Kebun Tanaman

I. Keterangan : Sistem

ini digunakan untuk mengelola tanaman kebun dengan memonitor dan memelihara tanaman secara efisien. Ada tiga kelas utama : Kebun Tanaman dan Pekerja. Kelas Taman

berisi informasi mengenai daftar kebun dan tanaman serta pekerja yang berisi informasi kelas Tumbuhan mengenai tanaman, dan kelas Pekerja yang berisi informasi mengenai pekerja yang merawat tanaman kebun.

Kelas 1: Taman

• Atribut :

- o GardenName : String – Nama taman
- o lokasi : String – Lokasi taman
- o PlantList : ArrayList < Tanaman > – Daftar tanaman di taman
- o listWorkers : ArrayList < Workers > – Daftar pekerja taman

• Metode:

- o addPlants (Tanaman tanaman) : Tambahkan tanaman baru ke taman
- o addWorker (Pekerja) pekerja) : Tambahkan pekerja baru ke taman
- assignWorkersToPlants (Tanaman) Tanaman Pekerja pekerja) : Tetapkan pekerja Untuk perawat o menanam tertentu

Kelas 2: Tumbuhan

• Atribut :

- o PlantName : String – Nama tanaman
- o Jenis tanaman : Tanaman tipe tali (contoh : " Tomat, Kangkung)

- o HealthStatus : String – Status kesehatan tanaman (contoh : "Sehat", "Butuh Pemeliharaan ")
- o pekerja : Pekerja – Pekerja yang bertanggung jawab menjawab Untuk perawat tanaman

• **Metode:**

- o updateHealthStatus (String status) : Memperbarui status kesehatan tanaman
- o setWorker (Pekerja pekerja) : Menentukan pekerja perawatan tanaman

Kelas 3: Pekerja

• **Atribut :** o

- WorkerName : String – Nama pekerja o
- idWorker : String – ID pekerja o
- PlantList : ArrayList < Tanaman > – Daftar tanaman yang dirawat oleh pekerja

• **Metode:**

- o addPlants (Tanaman) plants) : Tambahkan tanaman ke dalam daftar tanaman yang dirawat
- o deletePlants (Tanaman plants) : Hapus tanaman dari daftar tanaman yang dirawat

Jil. Diagram Kelas : •
Kelas Taman :

Kebun
- NamaTaman : String - Lokasi : String - PlantList : ArrayList < Tanaman > - daftar pekerja : ArrayList < Pekerja > +
addPlant () + addWorker () + assignWorkersToPlants ()

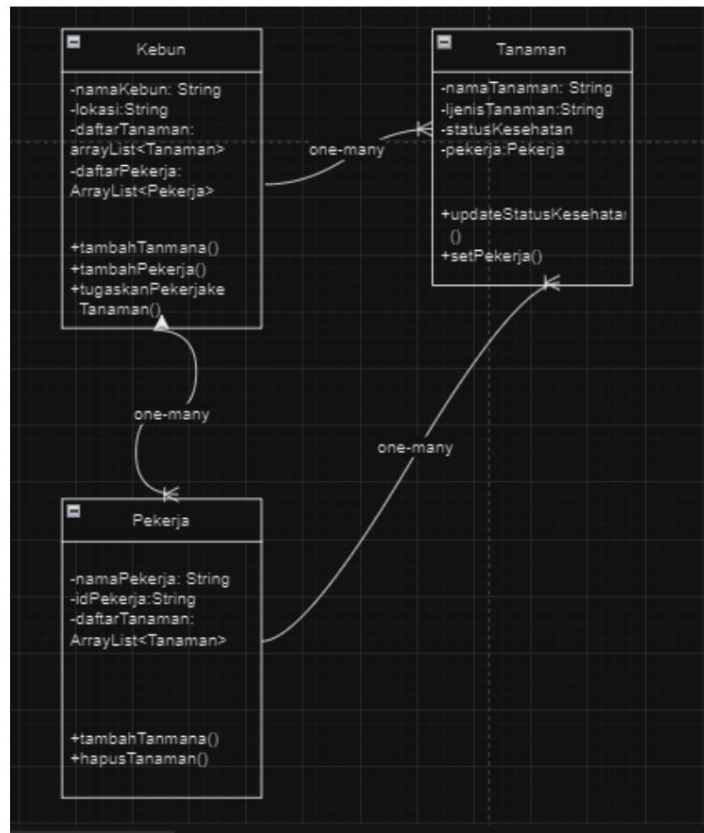
• Kelas Tumbuhan :

Tanaman
-NamaTanaman : String - Jenis tanaman : Tali - HealthStatus : String - pekerja : Pekerja +
updateHealthStatus() + setWorker()

• Kelas Pekerja :

Pekerja
- NamaPekerja : String - idPekerja : String - PlantList : ArrayList < Tanaman > +
tambahTanaman() + hapusTanaman()

- **+** (**Publik**) : Metode/atribut hanya dapat diakses dari dalam kelas.
- **-** (**Pribadi**) : Metode/atribut hanya dapat diakses dari dalam kelas.



III. Hubungan :

- Taman -> Tanaman :

Notasi : 1.. (Satu ke Banyak) * Setiap Kebun

Bisa Memiliki Banyak Tanaman Kebun : 1

Tanaman : 0.. *

tetapi setiap Tanaman hanya dimiliki oleh satu Kebun .

- Kebun -> Pekerja :

Notasi : 1.. (Satu ke Banyak) * Setiap Kebun

dapat memiliki Lot Pekerja Kebun : 1 Pekerja : 0.. *

tetapi setiap Pekerja hanya bekerja di satu Kebun.

- Pekerja -> Tanaman :

Notasi : 1.. (Satu ke Banyak) * Setiap Pekerja

Dapat Merawat Banyak Tanaman Pekerja : 1

Tanaman : 0.. *

tetapi setiap tanaman dirawat oleh satu pekerja .

IV. Terdapat empat tipe relasi dasar antar kelas : Inheritance Aggregation , dan Composition.

Asosiasi

Berikut penjelasan dan aplikasinya:

a. Pewarisan (Inheritance)

- Deskripsi : Pewarisan digunakan ketika suatu kelas mengambil properti dan perilaku dari suatu kelas. kelas lainnya.
- Contoh pada sistem garden : Misalnya, jika kita memiliki kelas Worker yang merupakan superclass, kita dapat memiliki subclass seperti Permanent Workers dan Casual Workers yang mana keduanya mewarisi atribut dan metode dari Worker.

b. Asosiasi (Association)

- Deskripsi : Asosiasi merupakan hubungan antar objek, dimana kelas yang satu berhubungan dengan kelas yang lain tanpa ada kepemilikan yang eksklusif.
- Contoh pada sistem kebun : kebun berasosiasi dengan Tanaman dan Pekerja . Ini tetapi Tidak dengan cara yang eksklusif " Karena taman mempunyai daftar tanaman dan pekerja " objek itu . memiliki

c. Agregasi (Agregasi)

- Deskripsi : Agregasi memiliki hubungan yang lebih longgar dibandingkan dengan komposisi tetapi kelas yang dimiliki masih dapat ada tanpa di mana satu kelas "memiliki" kelas lain harus tergantung pada kelas yang memilikinya
- Contoh pada sistem kebun : kebun agregat Tanaman dan Pekerja, artinya walaupun Kebun dihilangkan Tanaman dan Pekerjaanya Masih dapat Ada dengan cara yang independen dalam konteks yang lain.

d. Komposisi (Composition)

- Keterangan : Komposisi merupakan hubungan kepemilikan yang lebih kuat, dimana apabila objek yang "dimiliki" dihapus maka objek yang dimiliki juga ikut dihapus.
- Contoh pada sistem kebun : Bila tanaman mempunyai bagian atau unsur dalam seperti Akar atau Daun, dapat digunakan komposisi. seperti Jika Tanaman dihapus maka bagian ini juga ikut dihapus.

V. Kode Program : a.
Kebun

```

Welcome  J Kebun11.java 1 X  J Tanaman11.java 1  J Pekerja11.java 1  J Main11.java
J Kebun11.java > ...
1  import java.util.ArrayList;
2
3  public class Kebun11 {
4      private String namaKebun;
5      private String lokasi;
6      private ArrayList<Tanaman11> daftarTanaman;
7      private ArrayList<Pekerja11> daftarPekerja;
8
9      public Kebun11(String namaKebun, String lokasi) {
10         this.namaKebun = namaKebun;
11         this.lokasi = lokasi;
12         this.daftarTanaman = new ArrayList<>();
13         this.daftarPekerja = new ArrayList<>();
14     }
15
16     public void tambahTanaman(Tanaman11 tanaman) {
17         daftarTanaman.add(tanaman);
18     }
19
20     public void tambahPekerja(Pekerja11 pekerja) {
21         daftarPekerja.add(pekerja);
22     }
23
24     public void tugaskanPekerjaKeTanaman(Tanaman11 tanaman, Pekerja11 pekerja) {
25         tanaman.setPekerja(pekerja);
26         pekerja.tambahTanaman(tanaman);
27     }
28
29     // Getter untuk namaKebun
30     public String getNamaKebun() {
31         return namaKebun;
32     }
33
34     // Getter untuk daftarTanaman
35     public ArrayList<Tanaman11> getDaftarTanaman() {
36         return daftarTanaman;
37     }
38 }
39

```

b. Tanaman

```

Welcome  J Kebun11.java 1  J Tanaman11.java 1 X  J Pekerja11.java 1  J Main11.java
J Tanaman11.java > J Tanaman11 > getPekerjaNama0
1  public class Tanaman11 {
2      private String namaTanaman;
3      private String jenisTanaman;
4      private String statusKesehatan;
5      private Pekerja11 pekerja;
6
7      public Tanaman11(String namaTanaman, String jenisTanaman) {
8         this.namaTanaman = namaTanaman;
9         this.jenisTanaman = jenisTanaman;
10         this.statusKesehatan = "Sehat"; // Default status
11     }
12
13     public void updateStatusKesehatan(String status) {
14         this.statusKesehatan = status;
15     }
16
17     public void setPekerja(Pekerja11 pekerja) {
18         this.pekerja = pekerja;
19     }
20
21     public String getStatusKesehatan() {
22         return statusKesehatan;
23     }
24
25     public String getNamaTanaman() {
26         return namaTanaman;
27     }
28     public String getPekerjaNama() {
29         return pekerja != null ? pekerja.getNamaPekerja() : "Belum ditugaskan";
30     }
31 }
32

```

c. Pekerja

```

Pekerja11.java > Pekerja11
1 import java.util.ArrayList;
2
3 public class Pekerja11 {
4     private String namaPekerja;
5     private String idPekerja;
6     private ArrayList<Tanaman11> daftarTanaman;
7
8     public Pekerja11(String namaPekerja, String idPekerja) {
9         this.namaPekerja = namaPekerja;
10        this.idPekerja = idPekerja;
11        this.daftarTanaman = new ArrayList<>();
12    }
13
14    public void tambahTanaman(Tanaman11 tanaman) {
15        daftarTanaman.add(tanaman);
16    }
17
18    public void hapusTanaman(Tanaman11 tanaman) {
19        daftarTanaman.remove(tanaman);
20    }
21
22    public String getNamaPekerja() {
23        return namaPekerja;
24    }
25    public void tampilkanDaftarTanaman() {
26        if (daftarTanaman.isEmpty()) {
27            System.out.println(namaPekerja + " belum merawat tanaman.");
28        } else {
29            System.out.println(namaPekerja + " merawat tanaman berikut:");
30            for (Tanaman11 tanaman : daftarTanaman) {
31                System.out.println("- " + tanaman.getNamaTanaman());
32            }
33        }
34    }
35 }
36

```

d. Utama

```

Main11.java > Main11 > main(String[])
1 public class Main11 {
2     public static void main(String[] args) {
3
4         Kebun11 kebun = new Kebun11(namaKebun:"Kebun Sayur", lokasi:"Bogor");
5
6         Tanaman11 tomat = new Tanaman11(namaTanaman:"Tomat", jenisTanaman:"Sayuran");
7         Tanaman11 kangkung = new Tanaman11(namaTanaman:"Kangkung", jenisTanaman:"Sayuran");
8
9         // Membuat Pekerja
10        Pekerja11 pekerja1 = new Pekerja11(namaPekerja:"Budi", idPekerja:"P001");
11        Pekerja11 pekerja2 = new Pekerja11(namaPekerja:"Siti", idPekerja:"P002");
12
13        // Menambahkan Tanaman ke Kebun
14        kebun.tambahTanaman(tomat);
15        kebun.tambahTanaman(kangkung);
16
17        // Menambahkan Pekerja ke Kebun
18        kebun.tambahPekerja(pekerja1);
19        kebun.tambahPekerja(pekerja2);
20
21        // Menugaskan Pekerja untuk merawat Tanaman
22        kebun.tugaskanPekerjaKeTanaman(tomat, pekerja1);
23        kebun.tugaskanPekerjaKeTanaman(kangkung, pekerja2);
24
25        // Menampilkan daftar tanaman dan pekerja yang merawatnya
26        System.out.println("Daftar tanaman di kebun " + kebun.getNamaKebun() + ":");
27        for (Tanaman11 tanaman : kebun.getDaftarTanaman()) {
28            System.out.println(tanaman.getNamaTanaman() + " dirawat oleh: " + tanaman.getPekerjaNama());
29        }
30
31        // Menampilkan status tanaman
32        System.out.println("\nStatus kesehatan Tomat: " + tomat.getStatusKesehatan());
33
34        // Mengubah status kesehatan tanaman
35        tomat.updateStatusKesehatan(status:"Butuh Perawatan");
36        System.out.println("Status kesehatan Tomat setelah diperbarui: " + tomat.getStatusKesehatan());
37
38        // Menampilkan daftar tanaman yang dirawat oleh pekerja1
39        pekerja1.tampilkanDaftarTanaman();
40        pekerja2.tampilkanDaftarTanaman();
41    }
42 }
43

```

e. Keluaran

```
Messages' '-cp' 'C:\Users\HP\AppData\Roaming\Code\User\workspac
Daftar tanaman di kebun Kebun Sayur:
Tomat dirawat oleh: Budi
Kangkung dirawat oleh: Siti

Status kesehatan Tomat: Sehat
Status kesehatan Tomat setelah diperbarui: Butuh Perawatan
Budi merawat tanaman berikut:
- Tomat
Siti merawat tanaman berikut:
- Kangkung
PS D:\SEMESTER 3\PBO\Java\Jobsheet04\Perkebunan> █
```

Daftar tanaman di kebun: Program ini menampilkan setiap tanaman di kebun dan pekerja yang merawatnya.

Status kesehatan tanaman: Status kesehatan tanaman tomat sebelum dan sesudah diperbarui (default)

Daftar tanaman yang dirawat pekerja: Menampilkan tanaman yang dirawat oleh Budi dan Siti