

Jobsheet 04 - Relasi Kelas

I. Kompetensi

Setelah menempuh pokok bahasan ini, mahasiswa mampu:

1. Memahami konsep relasi kelas;
2. Mengimplementasikan relasi asosiasi ke dalam program.

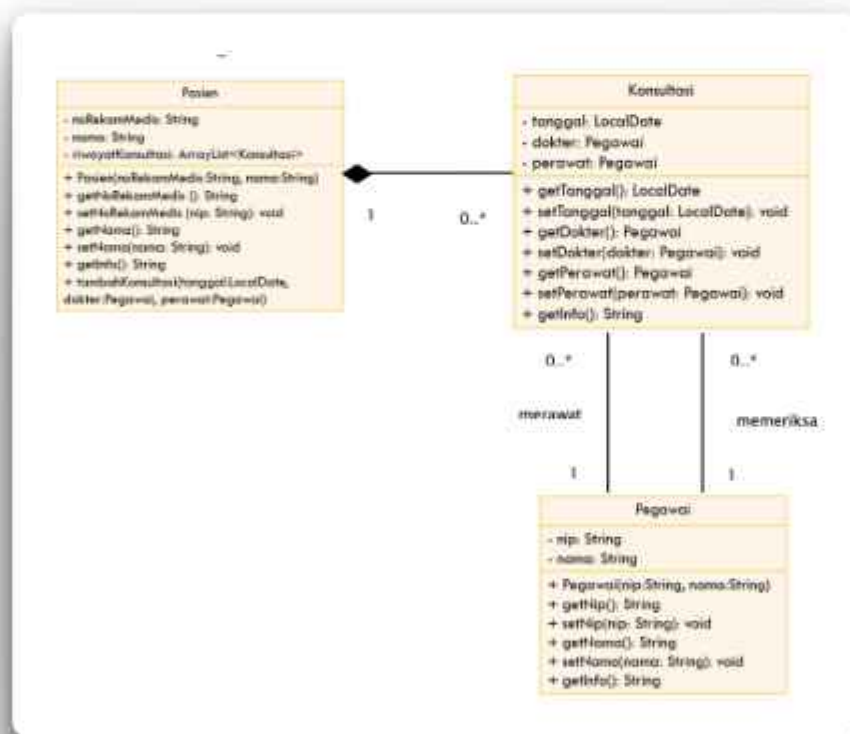
II. Pendahuluan

Pada kasus yang lebih kompleks, dalam suatu sistem akan ditemukan lebih dari satu *class* yang saling memiliki keterkaitan antara *class* satu dengan yang lain. Pada percobaan-percobaan sebelumnya, mayoritas kasus yang sudah dikerjakan hanya fokus pada satu *class* saja. Pada jobsheet ini akan dilakukan percobaan yang melibatkan beberapa *class* yang saling berkaitan.

III. Praktikum

Pada praktikum ini akan dikembangkan suatu sistem informasi rumah sakit yang menyimpan data riwayat konsultasi pasien.

Perhatikan diagram *class* berikut:



Screenshot by Xnapper.com

- a. Buatlah folder baru dengan nama RumahSakit
- b. Buatlah class Pegawai. Tambahkan atribut nip dan nama pada class Pegawai dengan akses modifier private

```
public class Pegawai {  
    private String nip;  
    private String nama;  
}
```

Screenshot by Xnapper.com

- c. Buatlah *constructor* untuk class Pegawai dengan parameter nip dan nama.

```
public Pegawai(String nip, String nama) {  
    this.nip = nip;  
    this.nama = nama;  
}
```

Screenshot by Xnapper.com

- d. Implementasikan **setter** dan **getter** untuk class Pegawai.

```
public String getNip() {  
    return nip;  
}  
  
public void setNip(String nip) {  
    this.nip = nip;  
}  
  
public String getNama() {  
    return nama;  
}  
  
public void setNama(String nama) {  
    this.nama = nama;  
}
```

Screenshot by Xnapper.com

```
public String getInfo(){  
    return nama + " (" + nip + ")";  
}
```

Screenshot by Xnapper.com

e. Implementasikan *method* `getInfo()` sebagai berikut:

```
public String getInfo(){  
    return nama + " (" + nip + ")";  
}
```

Screenshot by Xhapper.com

Hasil:

```
1 package RumahSakit;  
2  
3 public class Pegawai {  
4     private String nip;  
5     private String nama;  
6  
7     public Pegawai(String nip, String nama){  
8         this.nip = nip;  
9         this.nama = nama;  
10    }  
11  
12    public String getNip (String nip){  
13        return nip;  
14    }  
15  
16    public void setNip(){  
17        this.nip = nip;  
18    }  
19  
20    public String getNama(){  
21        return nama;  
22    }  
23  
24    public void setNama(){  
25        this.nama = nama;  
26    }  
27  
28    public String getInfo(){  
29        return nama + " (" + nip + ")";  
30    }  
31  
32 }
```

- f. Selanjutnya buatlah class Pasien kemudian tambahkan atribut noRekamMedis dan nama pada class Pasien dengan access level modifier private. Sediakan pula setter dan getter untuk kedua atribut tersebut.

```
public class Pasien {  
    private String noRekamMedis;  
    private String nama;  
  
    public String getNoRekamMedis() {  
        return noRekamMedis;  
    }  
  
    public void setNoRekamMedis(String noRekamMedis) {  
        this.noRekamMedis = noRekamMedis;  
    }  
  
    public String getName() {  
        return nama;  
    }  
  
    public void setName(String nama) {  
        this.nama = nama;  
    }  
}
```

Screenshot by Xnapper.com

```
1 public class Pasien {  
2     private String noRekamMedis;  
3     private String nama;  
4     private ArrayList<Konsultasi> riwayatKonsultasi;  
5  
6     public Pasien(String noRekamMedis, String nama){  
7         this.noRekamMedis = noRekamMedis;  
8         this.nama = nama;  
9         this.riwayatKonsultasi = new ArrayList<Konsultasi>();  
10    }  
11  
12    public String getNoRekamMedis() {  
13        return noRekamMedis;  
14    }  
15  
16    public void setNoRekamMedis(String noRekamMedis) {  
17        this.noRekamMedis = noRekamMedis;  
18    }  
19  
20    public String getName() {  
21        return nama;  
22    }  
23  
24    public void setName(String nama) {  
25        this.nama = nama;  
26    }  
}
```

- g. Buatlah constructor untuk class Pasien dengan parameter noRekamMedis dan nama

```
public Pasien(String noRekamMedis, String nama) {  
    this.noRekamMedis = noRekamMedis;  
    this.nama = nama;  
}
```

Screenshot by Xnapper.com

- h. Implementasikan *method* getInfo() sebagai berikut:

```
public String getInfo() {  
    String info = "";  
    info += "No Rekam Medis : " + this.noRekamMedis + "\n";  
    info += "Nama : " + this.nama + "\n";  
    info += "\n";  
    return info;  
}
```

Screenshot by Xnapper.com

- i. Sistem ini akan menyimpan data setiap konsultasi yang dilakukan pasien. Pasien bisa melakukan konsultasi lebih dari sekali. Oleh karena itu, data konsultasi akan disimpan dalam bentuk ArrayList dari objek-objek yang bertipe Konsultasi.
- j. Buatlah class dengan nama Konsultasi dengan atribut tanggal bertipe LocalDate, dokter bertipe Pegawai, dan perawat bertipe Pegawai. Set access level modifier private untuk seluruh atribut. Lakukan import java.time.LocalDate agar dapat mendeklarasikan atribut tanggal bertipe LocalDate.

```
import java.time.LocalDate;  
  
public class Konsultasi {  
    private LocalDate tanggal;  
    private Pegawai dokter;  
    private Pegawai perawat;  
}
```

Screenshot by Xnapper.com

k. Sediakan setter dan getter untuk masing-masing atribut pada class Konsultasi

```
public LocalDate getTanggal() {  
    return tanggal;  
}  
  
public void setTanggal(LocalDate tanggal) {  
    this.tanggal = tanggal;  
}  
  
public Pegawai getDokter() {  
    return dokter;  
}  
  
public void setDokter(Pegawai dokter) {  
    this.dokter = dokter;  
}  
  
public Pegawai getPerawat() {  
    return perawat;  
}  
  
public void setPerawat(Pegawai perawat) {  
    this.perawat = perawat;  
}
```

Screenshot by Xnapper.com

```
1 public class Konsultasi {  
2     private LocalDate tanggal;  
3     private Pegawai dokter;  
4     private Pegawai perawat;  
5  
6     public LocalDate getTanggal() {  
7         return tanggal;  
8     }  
9  
10    public void setTanggal(LocalDate tanggal) {  
11        this.tanggal = tanggal;  
12    }  
13  
14    public Pegawai getDokter() {  
15        return dokter;  
16    }  
17  
18    public void setDokter(Pegawai dokter) {  
19        this.dokter = dokter;  
20    }  
21  
22    public Pegawai getPerawat() {  
23        return perawat;  
24    }  
25  
26    public void setPerawat(Pegawai perawat) {  
27        this.perawat = perawat;  
28    }  
}
```

l. Implementasikan method getInfo() sebagai berikut:


```

public String getInfo(){
    String info = "";
    info += "\tTanggal: " + tanggal;
    info += ", Dokter: " + dokter.getInfo();
    info += ", Perawat: " + perawat.getInfo();
    info += "\n";

    return info;
}

```

Screenshot by Xnapper.com

```

1 public String getInfo() {
2     String info = "";
3     info += "\tTanggal: " + tanggal;
4     info += ", Dokter: " + dokter.getInfo();
5     info += ", Perawat: " + perawat.getInfo();
6     info += "\n";
7
8     return info;
9 }

```

- m. Untuk menyimpan data riwayat konsultasi pasien, maka tambahkan atribut riwayatKonsultasi pada class Pasien dengan tipe arrayList<Konsultasi>. Atribut ini akan menyimpan serangkaian objek bertipe Konsultasi. Import java.util.ArrayList agar dapat mendeklarasikan atribut bertipe ArrayList of object.

```

private String noRekamMedis;
private String nama;
private ArrayList<Konsultasi> riwayatKonsultasi;

```

Screenshot by Xnapper.com

- n. Buatlah constructor berparameter untuk class Pasien. Inisiasi nilai atribut noRekamMedis dan nama berdasarkan atribut nama. Instansiasi/buat ArrayList baru untuk atribut riwayatKonsultasi;

```
public Pasien(String noRekamMedis, String nama) {  
    this.noRekamMedis = noRekamMedis;  
    this.nama = nama;  
    this.riwayatKonsultasi = new ArrayList<Konsultasi>();  
}
```

Screenshot by Xnapper.com

- o. Lakukan import java.time.LocalDate agar dapat mendeklarasikan atribut tanggal bertipe LocalDate pada class Pasien. Selanjutnya, implementasikan method tambahKonsultasi() sebagai berikut:

```
public void tambahKonsultasi(LocalDate tanggal, Pegawai dokter, Pegawai perawat) {  
    Konsultasi konsultasi = new Konsultasi();  
    konsultasi.setTanggal(tanggal);  
    konsultasi.setDokter(dokter);  
    konsultasi.setPerawat(perawat);  
    riwayatKonsultasi.add(konsultasi);  
}
```

Screenshot by Xnapper.com

```
1: public void tambahKonsultasi(LocalDate tanggal, Pegawai dokter, Pegawai perawat) {  
2:     Konsultasi konsultasi = new Konsultasi();  
3:     konsultasi.setTanggal(tanggal);  
4:     konsultasi.setDokter(dokter);  
5:     konsultasi.setPerawat(perawat);  
6:     riwayatKonsultasi.add(konsultasi);  
7: }
```

- p. Modifikasi method getInfo() untuk mengembalikan info pasien dan daftar konsultasi yang pernah dilakukan


```

public String getInfo(){
    String info = "";
    info += "No Rekam Medis : " + this.noRekamMedis + "\n";
    info += "Nama : " + this.nama + "\n";

    if (!riwayatKonsultasi.isEmpty()) {
        info += "Riwayat Konsultasi :\n";

        for (Konsultasi konsultasi : riwayatKonsultasi) {
            info += konsultasi.getInfo();
        }
    }
    else{
        info += "Belum ada riwayat konsultasi";
    }

    info += "\n";

    return info;
}

```

Screenshot by Xnapper.com

```

1  public String getInfo() {
2      String info = "";
3      info += "No Rekam Medis : " + this.noRekamMedis + "\n";
4      info += "Nama : " + this.nama + "\n";
5
6      if (!riwayatKonsultasi.isEmpty()) {
7          info += "Riwayat Konsultasi :\n";
8          for (Konsultasi konsultasi : riwayatKonsultasi) {
9              info += konsultasi.getInfo();
10             }
11         } else {
12             info += "Belum ada riwayat konsultasi";
13         }
14
15         info += "\n";
16         return info;
17     }

```

- q. Lakukan import `java.time.LocalDate` agar dapat mendeklarasikan atribut tanggal bertipe `LocalDate` pada class `RumahSakitDemo`. Test program yang sudah dibuat dengan membuat objek-objek pada class `RumahSakitDemo`. Instansiasi objek baru bertipe `Pegawai` dengan nama `ani` menggunakan constructor `Pegawai(String nip, String nama)` dengan nilai argumen nip `"1234"` dan nama `"dr. Ani"`. Lanjutkan instansiasi objek sebagai berikut:

```
import java.time.LocalDate;

public class RumahSakitDemo {
    public static void main(String[] args) {
        Pegawai ani = new Pegawai("1234", "dr. Ani");
        Pegawai bagus = new Pegawai("4567", "dr. Bagus");

        Pegawai desi = new Pegawai("1234", "Ns. Desi");
        Pegawai eka = new Pegawai("4567", "Ns. Eka");

        Pasien pasien1 = new Pasien("343298", "Puspa Widya");
        pasien1.tambahKonsultasi(LocalDate.of(2021, 8, 11), ani, desi);
        pasien1.tambahKonsultasi(LocalDate.of(2021, 9, 11), bagus, eka);

        System.out.println(pasien1.getInfo());

        Pasien pasien2 = new Pasien("997744", "Yenny Anggraeni");
        System.out.println(pasien2.getInfo());
    }
}
```

Screenshot by Xnapper.com

```
1 package RumahSakit;
2 import java.time.LocalDate;
3
4 public class RumahSakitDemo {
5     public static void main(String[] args) {
6         Pegawai ani = new Pegawai("1234", "dr. Ani");
7         Pegawai bagus = new Pegawai("4567", "dr. Bagus");
8
9         Pegawai desi = new Pegawai("1234", "Ns. Desi");
10        Pegawai eka = new Pegawai("4567", "Ns. Eka");
11
12        Pasien pasien1 = new Pasien("343298", "Puspa Widya");
13        pasien1.tambahKonsultasi(LocalDate.of(2021, 8, 11), ani, desi);
14        pasien1.tambahKonsultasi(LocalDate.of(2021, 9, 11), bagus, eka);
15
16        System.out.println(pasien1.getInfo());
17
18        Pasien pasien2 = new Pasien("997744", "Yenny Anggraeni");
19        System.out.println(pasien2.getInfo());
20    }
21 }
22
23
```

- r. *Compile* kemudian *run* `RumahSakitDemo` dan didapatkan hasil seperti berikut:

```

No Rekam Medis : Puspa Widya
Nama : 343298
Riwayat Konsultasi :
    Tanggal: 2021-08-11, Dokter: dr. Ani (1234), Perawat: Ns. Desi (1234)
    Tanggal: 2021-09-11, Dokter: dr. Bagus (4567), Perawat: Ns. Eka (4567)

No Rekam Medis : Yenny Anggraeni
Nama : 997744
Belum ada riwayat konsultasi

```

Screenshot by Xnapper.com

```

No Rekam Medis : 343298
Nama : Puspa Widya
Riwayat Konsultasi :
    Tanggal: 2021-08-11, Dokter: dr. Ani (1234), Perawat: Ns. Desi (1234)
    Tanggal: 2021-09-11, Dokter: dr. Bagus (4567), Perawat: Ns. Eka (4567)

No Rekam Medis : 997744
Nama : Yenny Anggraeni
Belum ada riwayat konsultasi
nazril@Muhammads-MacBook-Air Jobsheet4 %

```

Screenshot by Xnapper.com

Pertanyaan

Berdasarkan percobaan 1, jawablah pertanyaan-pertanyaan yang terkait:

1. Di dalam *class* Pegawai, Pasien, dan Konsultasi, terdapat method *setter* dan *getter* untuk masing-masing atributnya. Apakah gunanya *method setter* dan *getter* tersebut ?

Gunanya *setter* dan *getter*:

- **Setter:** Mengubah nilai atribut privat.
- **Getter:** Mengambil nilai atribut privat tanpa mengubahnya

2. Di dalam *class* Konsultasi tidak secara eksplisit terdapat constructor dengan parameter. Apakah ini berarti *class* Konsultasi tidak memiliki constructor?

Class Konsultasi memiliki **constructor default** tanpa parameter, meskipun tidak dideklarasikan secara eksplisit.

3. Perhatikan *class* Konsultasi, atribut mana saja yang bertipe *object*? dokter dan perawat (keduanya bertipe Pegawai), serta tanggal (tipe *LocalDate*)
4. Perhatikan *class* Konsultasi, pada baris manakah yang menunjukkan bahwa *class* Konsultasi memiliki relasi dengan *class* Pegawai?

Terlihat pada atribut Pegawai dokter; dan Pegawai perawat;

5. Perhatikan pada *class* Pasien, apa yang dilakukan oleh kode `konsultasi.getInfo()`? Mengambil informasi tanggal, dokter, dan perawat dari setiap konsultasi.

6. Pada method `getInfo()` dalam *class* Pasien, terdapat baris kode:

```
if (!riwayatKonsultasi.isEmpty())
```

Apakah yang dilakukan oleh baris tersebut?

Memeriksa apakah ada riwayat konsultasi. Jika kosong, tampilkan pesan "Belum ada

riwayat konsultasi".

7. Pada constructor class Pasien, terdapat baris kode:

`this.riwayatKonsultasi = new ArrayList<>();`

Apakah yang dilakukan oleh baris tersebut? Apakah yang terjadi jika baris tersebut dihilangkan?

Menginisialisasi riwayatKonsultasi. Tanpa ini, akan terjadi NullPointerException saat menambah konsultasi.

IV. Tugas

Implementasikan studi kasus yang telah dibuat pada tugas PBO Teori ke dalam program

Reservasi ditambahkan untuk tamu: Tamu: Andi, Nomor Kamar: 101, Tipe Kamar: Deluxe, Tanggal: 2024-09-17
Reservasi ditambahkan untuk tamu: Tamu: Budi, Nomor Kamar: 102, Tipe Kamar: Suite, Tanggal: 2024-09-18
Hotel: Hotel Bintang Lima, Jumlah Reservasi: 2
Daftar Reservasi di Hotel Hotel Bintang Lima:
Tamu: Andi, Nomor Kamar: 101, Tipe Kamar: Deluxe, Tanggal: 2024-09-17
Tamu: Budi, Nomor Kamar: 102, Tipe Kamar: Suite, Tanggal: 2024-09-18
nazril@Muhammads-MacBook-Air Jobsheet4 %

Screenshot by Xnapper.com