

Laporan Praktikum Pemrograman Berorientasi Objek Dosen
Pengampu : Vit Zuraida, S.Kom., M.Kom.



Nama : Syaqira Nazaretna
NIM : 2341760123
Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024

Jobsheet 04 - Relasi Kelas

I. Kompetensi

Setelah menempuh pokok bahasan ini, mahasiswa mampu:

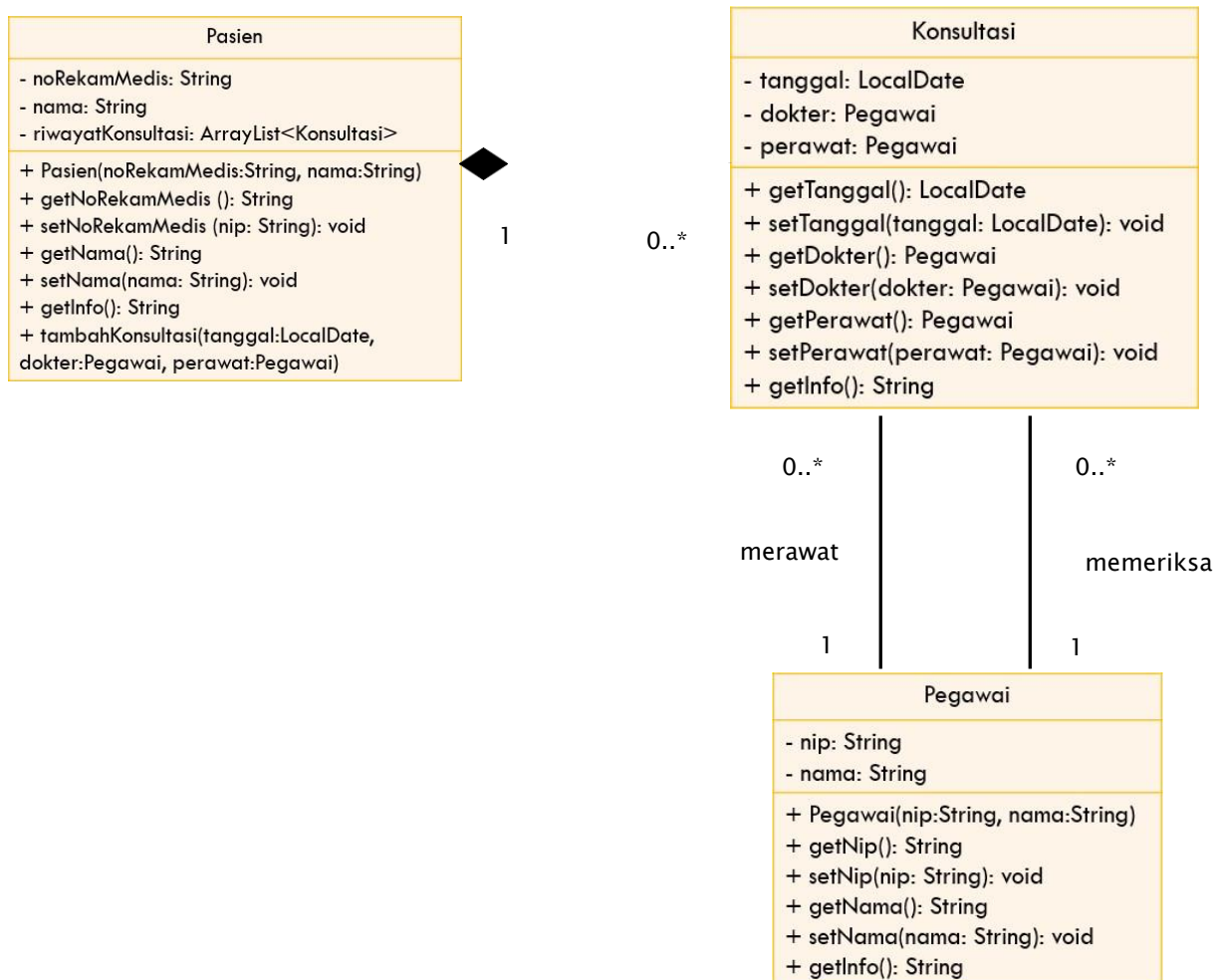
1. Memahami konsep relasi kelas; 2. Mengimplementasikan relasi asosiasi ke dalam program.

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:



- 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;
```

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

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

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

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

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

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

- 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

```
import java.time.LocalDate;
import java.util.ArrayList;

public class Pasien {
    private String noRekamMedis;
    private String nama;
    //atribut untuk menyimpan data riwayat konsultasi
    private ArrayList<Konsultasi> riwayatKonsultasi;

    public String getNoRekamMedis(){
        return noRekamMedis;
    }
    public void setNoRekamMedis(){
        this.noRekamMedis = noRekamMedis;
    }
    public String getNama(){
        return nama;
    }
    public void setNama(String nama){
        this.nama = nama;
    }
}
```

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

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

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

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

- 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;  
}
```

- 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;
}

```

- l. Implementasikan method getInfo() sebagai berikut:

```

public String getInfo(){
    String info = "";
    info += "Tanggal Konsultasi: " + tanggal;
    info += "\nDokter: " + dokter.getInfo();
    info += "\nPerawat: " + perawat.getInfo();
    info += "\n";

    return info;
}

```

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

```

public class Pasien {
    private String noRekamMedis;
    private String nama;
    //atribut untuk menyimpan data riwayat konsultasi
    private ArrayList<Konsultasi> riwayatKonsultasi;
}

```

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


```

}
//Konstruktor Berparameter
public Pasien(String noRekamMedis, String nama){
    this.noRekamMedis = noRekamMedis;
    this.nama = nama;
    this.riwayaKonsultasi = new ArrayList<Konsultasi>();
}

```

- 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);
    riwayaKonsultasi.add(konsultasi);
}

```

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

```

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

    //mengembalikan info pasien dan daftar konsultasi yang pernah dilakukan
    if (!riwayaKonsultasi.isEmpty()) {
        info += "Riwayat Konsultasi :\n";

        for (Konsultasi konsultasi : riwayaKonsultasi) {
            info += konsultasi.getInfo();
        }
    }
    else{
        info += "Belum ada riwayat konsultasi";
    }
    info += "\n";
    return info;
}

```

- q. Test program yang sudah dibuat dengan membuat objek-objek pada class `RumahSakitDemo`. Lakukan `import java.time.LocalDate` agar dapat mendeklarasikan atribut tanggal bertipe `LocalDate`. 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:

```

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

```

- 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

```

```

9a78517\bin' 'RumahSakitDemo'
No Rekam Medis : 343298
Nama          : Puspa Widya
Riwayat Konsultasi :
Tanggal Konsultasi: 2021-08-11
Dokter: dr. Ani (1234)
Perawat: Ns. Desi (1234)
Tanggal Konsultasi: 2021-09-11
Dokter: dr. Bagus (4567)
Perawat: Ns. Eka (4567)

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

PS D:\KULIAH\SEMESTER 3\PBO\JOBSHEET\PRAKTIKUM 4>

```


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 ?

Jawab : pada class tersebut setter gunanya untuk mengubah atau menetapkan nilai dari atribut privat, sedangkan getter digunakan untuk mengambil nilai dari atribut privat.

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

Jawab : Jika di dalam kelas *Konsultasi* tidak terdapat konstruktor yang didefinisikan secara eksplisit, maka Java secara otomatis akan menyediakan konstruktor default. Konstruktor default ini adalah konstruktor tanpa parameter yang menginisialisasi objek tanpa melakukan tindakan khusus.

3. Perhatikan *class* Konsultasi, atribut mana saja yang bertipe *object*?

Jawab : dalam class konsultasi atribut yang bertipe object adalah tanggal, dokter, perawat

4. Perhatikan *class* Konsultasi, pada baris manakah yang menunjukkan bahwa *class* Konsultasi memiliki relasi dengan *class* Pegawai?

Jawab : pada attribute dan method pada kode ini Atribut

```
private Pegawai dokter;  
private Pegawai perawat;
```

Method

```
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;  
}
```

5. Perhatikan pada *class* Pasien, apa yang dilakukan oleh kode `konsultasi.getInfo()`? **Jawab :** akan melakukan pemanggilan method `konsultasi.getInfo()` yang dilakukan adalah mengakses informasi yang ada dalam method tersebut

6. Pada method `getInfo()` dalam class Pasien, terdapat baris kode:
`if (!riwayatKonsultasi.isEmpty())`

Apakah yang dilakukan oleh baris tersebut?

Jawab : mengecek apakah riwayat konsultasi kosong atau tidak, jika tidak maka akan mencetak riwayat konsultasi pasien. Jika iya maka akan mencetak info "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?

Jawab : melakukan inisialisasi atribut riwayatKonsultasi, yang kemungkinan bertipe ArrayList, dengan objek ArrayList baru. Artinya, saat objek Pasien dibuat, atribut riwayatKonsultasi akan langsung memiliki nilai berupa daftar kosong yang siap digunakan untuk menampung riwayat konsultasi.

IV. Tugas

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

```
1  package tugas;
2  import java.util.ArrayList;
3
4  public class Produk {
5      private String namaProduk;
6      private int harga;
7      private String kategori;
8      private ArrayList<Order> orders;
9
10     public Produk(String namaProduk, int harga, String kategori) {
11         this.namaProduk = namaProduk;
12         this.harga = harga;
13         this.kategori = kategori;
14         this.orders = new ArrayList<>();
15     }
16
17     public String getNamaProduk() {
18         return namaProduk;
19     }
20
21     public void setNamaProduk(String namaProduk) {
22         this.namaProduk = namaProduk;
23     }
24
25     public int getHarga() {
26         return harga;
27     }
28
29     public void setHarga(int harga) {
30         this.harga = harga;
31     }
32
33     public String getKategori() {
34         return kategori;
35     }
36
37     public void setKategori(String kategori) {
38         this.kategori = kategori;
39     }
40
41     public String infoProduk() {
42         return "Produk: " + namaProduk + ", Harga: " + harga + ", Kategori: " + kategori;
43     }
44 }
45
```



```
1  package tugas;
2  import java.time.LocalDate;
3  import java.util.ArrayList;
4  import java.util.Date;
5
6  public class Order {
7      private Date tanggal;
8      private String idOrder;
9      private ArrayList<Produk> orderDetails;
10
11     public Order(Date tanggal, String idOrder) {
12         this.tanggal = tanggal ;
13         this.idOrder = idOrder;
14
15         this.orderDetails = new ArrayList<>();
16     }
17     public Date getTanggalOrder() {
18         return tanggal;
19     }
20     public void setTanggal(Date tanggal) {
21         this.tanggal = tanggal;
22     }
23
24     public String getIdOrder() {
25         return idOrder;
26     }
27
28     public void setIdOrder(String idOrder) {
29         this.idOrder = idOrder;
30     }
31
32     public void addProduk(Produk produk) {
33         orderDetails.add(produk);
34     }
35
36     public String infoOrder() {
37         return "Order ID: " + idOrder + ", Tanggal: " + tanggal;
38     }
39 }
40
41
```

```

1  package tugas;
2  import java.util.Date;
3
4  public class OrderDetail {
5      private Produk namaProduk;
6      private Date tanggal;
7      private int jumlah;
8
9      public OrderDetail(Produk namaProduk, Date date, int jumlah) {
10         this.namaProduk = namaProduk;
11         this.tanggal = tanggal;
12         this.jumlah = jumlah;
13     }
14     public double getSubtotal(double hargaPerUnit) {
15         return jumlah * hargaPerUnit;
16     }
17
18     public int getJumlah() {
19         return jumlah;
20     }
21
22     public Produk getProduk() {
23         return namaProduk;
24     }
25
26     public void setProduk(Produk namaProduk) {
27         this.namaProduk = namaProduk;
28     }
29
30     public Date getTanggal() {
31         return tanggal;
32     }
33
34     public void setTanggal(Date tanggal) {
35         this.tanggal = tanggal;
36     }
37
38     public String infoOrderDetail() {
39
40         String info = "";
41         info += "\nDetail " + namaProduk.infoProduk();
42         info += "\nJumlah: " + jumlah;
43         info += "\n";
44
45         return info;
46     }
47
48
49 }
50
51

```



```

1 package tugas;
2 import java.util.Date;
3
4 public class MarketDemo {
5     public static void main(String[] args) {
6         // Membuat produk
7         Produk produk1 = new Produk("Laptop", 10000000, "Elektronik");
8         Produk produk2 = new Produk("Mouse", 150000, "Aksesoris");
9         Date tanggal = new Date();
10
11         // Membuat order
12         Order order1 = new Order(new Date(), "123");
13         Order order2 = new Order(new Date(), "456");
14
15         // Membuat order detail
16         OrderDetail detail1 = new OrderDetail(produk1, new Date(), 2);
17         OrderDetail detail2 = new OrderDetail(produk2, new Date(), 1);
18         OrderDetail detail3 = new OrderDetail(produk2, new Date(), 1);
19
20         // Menambahkan order detail ke order
21         order1.setTanggal(new Date());
22         System.out.println(order1.infoOrder());
23
24         // Menampilkan informasi produk
25         System.out.println(produk1.infoProduk());
26         System.out.println(produk2.infoProduk());
27
28         System.out.println(detail1.infoOrderDetail());
29         System.out.println(detail2.infoOrderDetail());
30
31         System.out.println("-----");
32         order2.setTanggal(new Date());
33         order2.getTanggalOrder();
34         System.out.println(order2.infoOrder());
35
36         // Menampilkan informasi produk
37         System.out.println(produk2.infoProduk());
38         System.out.println(detail3.infoOrderDetail());
39     }
40 }
41
42

```

Outputnya

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

ws\jdt.ls-java-project\bin' 'tugas.MarketDemo'
Order ID: 123, Tanggal: Wed Oct 02 13:40:24 WIB 2024
Produk: Laptop, Harga: 10000000, Kategori: Elektronik
Produk: Mouse, Harga: 150000, Kategori: Aksesoris

Detail Produk: Laptop, Harga: 10000000, Kategori: Elektronik
Jumlah: 2

Detail Produk: Mouse, Harga: 150000, Kategori: Aksesoris
Jumlah: 1

-----
Order ID: 456, Tanggal: Wed Oct 02 13:40:24 WIB 2024
Produk: Mouse, Harga: 150000, Kategori: Aksesoris

Detail Produk: Mouse, Harga: 150000, Kategori: Aksesoris
Jumlah: 1

PS D:\KULIAH\SEMESTER 3\PBO\JOBSHEET\PRAKTIKUM 4\tugas>
```