

PEMROGRAMAN BERBASIS OBJEK

Praktikum Relasi *Class*



Lavina/2341760062

SIB 2D

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

2024

A. PRAKTIKUM

Pasien.java

Class Pasien berisi method setter dan getter yang berguna untuk menginisialisasi dan mengakses data dari pasien. Pada class ini atribut-atributnya dibuat private agar tidak bisa digunakan secara sembarangan dari class lain.

```
1 package RumahSakit;
2
3 import java.time.LocalDate;
4 import java.util.ArrayList;
5
6 /**
7  * Pasien
8  */
9 public class Pasien {
10     private String noRekamMedis;
11     private String nama;
12     private ArrayList<Konsultasi> riwayatKonsultasi;
13
14     public Pasien(String noRekamMedis, String nama) {
15         this.noRekamMedis = noRekamMedis;
16         this.nama = nama;
17         this.riwayatKonsultasi = new ArrayList<Konsultasi>();
18     }
19
20     public String getNoRekamMedis() {
21         return noRekamMedis;
22     }
23
24     public void setNoRekamMedis(String noRekamMedis) {
25         this.noRekamMedis = noRekamMedis;
26     }
27
28     public String getNama() {
29         return nama;
30     }
31
32     public void setNama(String nama) {
33         this.nama = nama;
34     }
35
36     public void tambahKonsultasi(LocalDate tanggal, Pegawai dokter, Pegawai perawat) {
37         Konsultasi konsultasi = new Konsultasi();
38         konsultasi.setTanggal(tanggal);
39         konsultasi.setDokter(dokter);
40         konsultasi.setPerawat(perawat);
41         riwayatKonsultasi.add(konsultasi);
42     }
43
44     public String getInfo() {
45         String info = "";
46         info += "No Rekam Medis : " + this.noRekamMedis + "\n";
47         info += "Nama          : " + this.nama + "\n";
48
49         if (!riwayatKonsultasi.isEmpty()) {
50             info += "Riwayat Konsultasi : \n";
```

```

52     for (Konsultasi konsultasi : riwayatKonsultasi) {
53         info += konsultasi.getInfo();
54     }
55 } else {
56     info += "Belum ada riwayat konsultasi";
57 }
58
59 info += "\n";
60
61 return info;
62 }
63
64 }

```

Pada class pasien ini juga terdapat method tambahKonsultasi yang berfungsi untuk menambah dan mencatat riwayat konsultasi dari pasien yang pada method tersebut data konsultasi diinisialisasi dengan menggunakan setter dari class konsultasi lalu data yang sudah dibuat ditambahkan atau disimpan dengan method add yang berasal dari class konsultasi juga.

Pegawai.java

Class ini berisi atribut yang merupakan data diri dari pegawai rumah sakit yang dibuat private dan bisa diakses dengan setter dan getter.

```

1  package RumahSakit;
2
3  /**
4   * Pegawai
5   */
6  public class Pegawai {
7      private String nip;
8      private String nama;
9
10     public Pegawai(String nip, String nama) {
11         this.nip = nip;
12         this.nama = nama;
13     }
14
15     public String getNip() {
16         return nip;
17     }
18

```

```

19     public void setNip(String nip) {
20         this.nip = nip;
21     }
22
23     public String getNama() {
24         return nama;
25     }
26
27     public void setNama(String nama) {
28         this.nama = nama;
29     }
30
31     public String getInfo() {
32         return nama + " (" + nip + ")";
33     }
34
35 }

```

Konsultasi.java

Class konsultasi ini berisi atribut hasil instansiasi objek dari class pegawai yaitu dokter dan pegawai karena pada konsultasi terdapat data dokter dan pegawai yang menangani pasien, terdapat atribut tanggal konsultasi juga yang mana atribut-atribut tersebut bisa diakses dengan setter dan getter.

```

1  package RumahSakit;
2
3  import java.time.LocalDate;
4
5  /**
6   * Konsultasi
7   */
8  public class Konsultasi {
9      private LocalDate tanggal;
10     private Pegawai dokter;
11     private Pegawai perawat;
12
13     public LocalDate getTanggal() {
14         return tanggal;
15     }

```

```
17     public void setTanggal(LocalDate tanggal) {
18         this.tanggal = tanggal;
19     }
20
21     public Pegawai getDokter() {
22         return dokter;
23     }
24
25     public void setDokter(Pegawai dokter) {
26         this.dokter = dokter;
27     }
28
29     public Pegawai getPerawat() {
30         return perawat;
31     }
32
33     public void setPerawat(Pegawai perawat) {
34         this.perawat = perawat;
35     }
36
37     public String getInfo() {
38         String info = "";
39
40         info += "\tTanggal : " + tanggal;
41         info += ", Dokter : " + dokter.getInfo();
42         info += ", Perawat : " + perawat.getInfo();
43         info += "\n";
44
45         return info;
46     }
47
48 }
```

RumahSakitDemo.java

```
1 package RumahSakit;
2
3 import java.time.LocalDate;
4
5 /**
6  * RumahSakitDemo
7  */
8 public class RumahSakitDemo {
9
10     public static void main(String[] args) {
11         Pegawai ani = new Pegawai("1234", "dr. Ani");
12         Pegawai bagus = new Pegawai("4567", "dr. Bagus");
13
14         Pegawai desi = new Pegawai("1234", "Ns.Desi");
15         Pegawai eka = new Pegawai("4567", "Ns.Eka");
16
17         Pasien pasien1 = new Pasien("434678", "Puspa Widya");
18         pasien1.tambahKonsultasi(LocalDate.of(2024, 9, 16), ani, desi);
19         pasien1.tambahKonsultasi(LocalDate.of(2024, 10, 23), bagus, eka);
20
21         System.out.println(pasien1.getInfo());
22
23         Pasien pasien2 = new Pasien("998345", "Yenny Anggraeni");
24         System.out.println(pasien2.getInfo());
25     }
26 }
```

Output:

```
No Rekam Medis : 434678
Nama           : Puspa Widya
Riwayat Konsultasi :
    Tanggal : 2024-09-16, Dokter : dr. Ani (1234), Perawat : Ns.Desi (1234)
    Tanggal : 2024-10-23, Dokter : dr. Bagus (4567), Perawat : Ns.Eka (4567)

No Rekam Medis : 998345
Nama           : Yenny Anggraeni
Belum ada riwayat konsultasi
```

B. 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: Method setter berfungsi untuk menginisialisasi value dari atribut private dari class tersebut sedangkan getter untuk mengambil/mengakses/mendapatkan value yang sudah dibuat dari setter.

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

Jawab: Class tersebut tetap memiliki constructor default dari java yaitu constructor yang tidak berparameter.

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

Jawab: Ada dua, yaitu atribut dokter dan pasien yang merupakan instansiasi object dari Pegawai.

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

Jawab: Relasinya ditunjukkan oleh kode berikut:

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

yang menunjukkan bahwa pada class Konsultasi bisa membuat lebih dari satu object Pegawai.

5. Perhatikan pada class Pasien, apa yang dilakukan oleh kode konsultasi.getInfo()?

Jawab: Kode tersebut mengakses method getInfo() yang berasal dari class Konsultasi yang berfungsi untuk menampilkan informasi riwayat konsultasi pasien.

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

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

Apakah yang dilakukan oleh baris tersebut?

Jawab: riwayatKonsultasi adalah array dari object Konsultasi, jadi kode tersebut berfungsi untuk mengecek apakah arraynya memiliki isi atau tidak kosong.

7. Pada constructor class Pasien, terdapat baris kode:

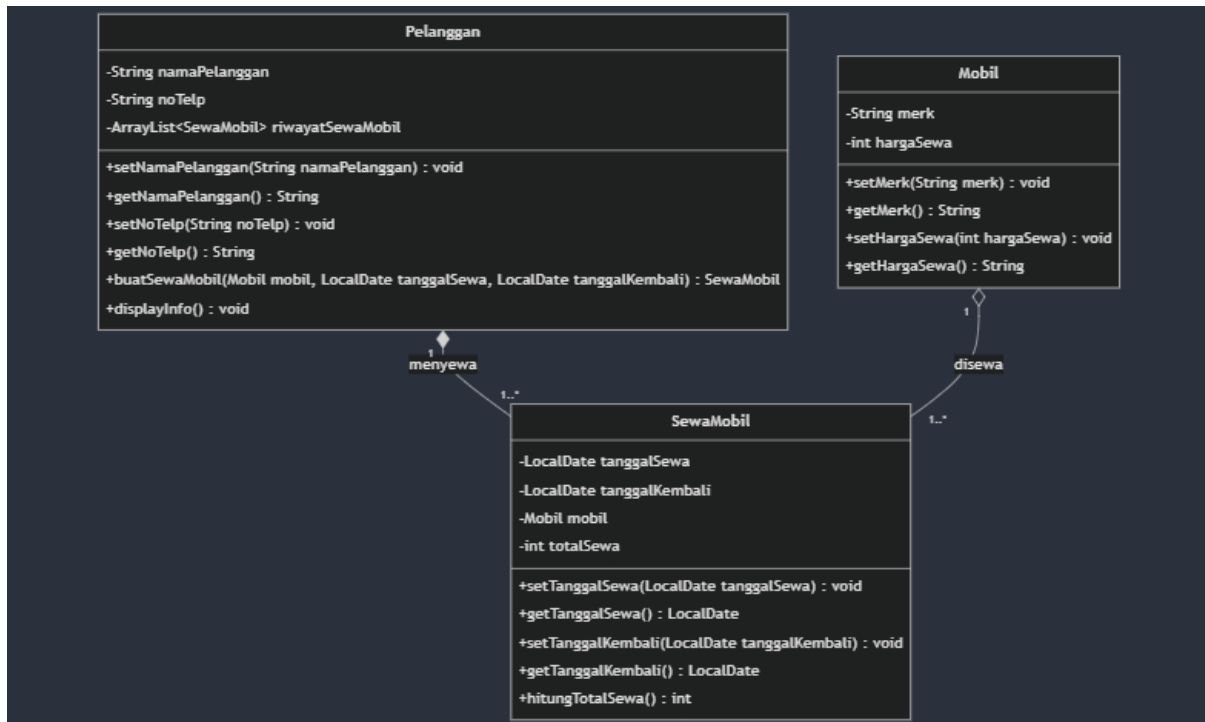
```
this.riwayatKonsultasi = new ArrayList<>();
```

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

Jawab: Kode tersebut melakukan inisialisasi terhadap atribut riwayatKonsultasi dengan object baru yang dapat digunakan untuk menyimpan riwayat konsultasi pasien. Jika baris ini dihilangkan, maka atribut riwayatKonsultasi tidak akan diinisialisasi, dan akan bernilai null secara default. Jika ada kode yang mencoba untuk menambahkan data ke dalam riwayatKonsultasi tanpa inisialisasi ini, maka akan terjadi error NullPointerException

Tugas

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



Mobil.java

```
1 package PenyewaanMobil;
2
3 public class Mobil {
4     private String merk;
5     private int hargaSewa;
6
7     public Mobil(String merk, int hargaSewa) {
8         this.merk = merk;
9         this.hargaSewa = hargaSewa;
10    }
11
12    public void setMerk(String merk) {
13        this.merk = merk;
14    }
15
16    public String getMerk() {
17        return merk;
18    }
19 }
```



```

20     public void setHargaSewa(int hargaSewa) {
21         if (hargaSewa != 0) {
22             this.hargaSewa = hargaSewa;
23         } else {
24             System.out.println("Harga sewa tidak valid");
25         }
26     }
27
28     public int getHargaSewa() {
29         return hargaSewa;
30     }
31 }
32

```

Pelanggan.java

```

1  package PenyewaanMobil;
2
3  import java.time.LocalDate;
4  import java.util.ArrayList;
5
6  public class Pelanggan {
7      private String namaPelanggan;
8      private String noTelp;
9      private ArrayList<SewaMobil> riwayatSewaMobil;
10     SewaMobil sewa = new SewaMobil();
11
12     public Pelanggan(String namaPelanggan, String noTelp) {
13         this.namaPelanggan = namaPelanggan;
14         this.noTelp = noTelp;
15         this.riwayatSewaMobil = new ArrayList<SewaMobil>();
16     }
17
18     public void setNamaPelanggan(String namaPelanggan) {
19         this.namaPelanggan = namaPelanggan;
20     }
21
22     public String getNamaPelanggan() {
23         return namaPelanggan;
24     }
25
26     public void setNoTelp(String noTelp) {
27         this.noTelp = noTelp;
28     }
29
30     public String getNoTelp() {
31         return noTelp;
32     }
33
34     public SewaMobil sewaMobil(Mobil mobil, LocalDate tanggalSewa, LocalDate tanggalKembali) {
35         sewa.setTanggalSewa(tanggalSewa);
36         sewa.setTanggalKembali(tanggalKembali);
37         sewa.setMobil(mobil);
38         riwayatSewaMobil.add(sewa);
39
40         return sewa;
41     }

```

```

42
43 public void displayInfo() {
44     System.out.println("Nama Pelanggan: " + this.namaPelanggan);
45     System.out.println("No Telepon: " + this.noTelp);
46     System.out.println("Jumlah Sewa: " + sewa.hitungTotalSewa());
47 }
48 }
49

```

SewaMobil.java

```

1 package PenyewaanMobil;
2
3 import java.time.LocalDate;
4 import java.time.temporal.ChronoUnit;
5
6 public class SewaMobil {
7     private LocalDate tanggalSewa;
8     private LocalDate tanggalKembali;
9     private Mobil mobil;
10    private int totalSewa;
11
12    public void setTanggalSewa(LocalDate tanggalSewa) {
13        this.tanggalSewa = tanggalSewa;
14    }
15
16    public LocalDate getTanggalSewa() {
17        return tanggalSewa;
18    }
19
20    public void setTanggalKembali(LocalDate tanggalKembali) {
21        this.tanggalKembali = tanggalKembali;
22    }
23
24    public LocalDate getTanggalKembali() {
25        return tanggalKembali;
26    }
27
28    public void setMobil(Mobil mobil) {
29        this.mobil = mobil;
30    }
31
32    public int hitungTotalSewa() {
33        long lamaSewa = ChronoUnit.DAYS.between(this.tanggalSewa, this.tanggalKembali);
34
35        this.totalSewa = mobil.getHargaSewa() * (int) lamaSewa;
36        return totalSewa;
37    }
38 }
39

```

PenyewaanMobilDemo.java

```
1 package PenyewaanMobil;  
2  
3 import java.time.LocalDate;  
4  
5 public class PenyewaanMobilDemo {  
6     public static void main(String[] args) {  
7         Mobil mobil1 = new Mobil("Avanza", 10_000);  
8         Mobil mobil2 = new Mobil("Toyota", 20_000);  
9  
10        Pelanggan pelanggan1 = new Pelanggan("Lavina", "08712345678");  
11        pelanggan1.sewaMobil(mobil2, LocalDate.of(2024, 9, 1), LocalDate.of(2024, 9, 5));  
12        pelanggan1.displayInfo();  
13    }  
14 }  
15
```

Output:

```
Nama Pelanggan: Lavina  
No Telepon: 08712345678  
Jumlah Sewa: 80000
```

Pelanggan menyewa mobil selama 4 hari sehingga jumlah yang harus dibayar yaitu $4 \times 20.000 = 80.000$.