

Pemrograman Berorientasi Objek

Jobsheet 4

Dosen Pengampu : Endah Septa Sintiya, S.Pd., M.Kom.



Nama : Annisa
NIM : 2341760032
Kelas : SIB 2C
Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024/2025

Berikut merupakan kode program dari class Pegawai

```
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() {
13         return nip;
14     }
15
16     public void setNip(String nip) {
17         this.nip = nip;
18     }
19
20     public String getNama() {
21         return nama;
22     }
23
24     public void setNama(String nama) {
25         this.nama = nama;
26     }
27
28     public String getInfo() {
29         return nama + " (" + nip + ")";
30     }
31 }
32
```

Berikut merupakan kode program dari class Pasien

```
1  package RumahSakit;
2
3  import java.time.LocalDate;
4  import java.util.ArrayList;
5
6  public class Pasien {
7      private String noRekamMedis;
8      private String nama;
9      private ArrayList<Konsultasi> riwayatKonsultasi;
10
11     public String getNoRekamMedis() {
12         return noRekamMedis;
13     }
14
15     public void setNoRekamMedis(String noRekamMedis) {
16         this.noRekamMedis = noRekamMedis;
17     }
18
19     public String getName() {
20         return nama;
21     }
22
23     public void setName(String nama) {
24         this.nama = nama;
25     }
26
27     public Pasien (String noRekamMedis, String nama) {
28         this.noRekamMedis = noRekamMedis;
29         this.nama = nama;
30         this.riwayatKonsultasi = new ArrayList<Konsultasi>();
31     }
32
33     public void tambahKonsultasi(LocalDate tanggal, Pegawai dokter, Pegawai perawat) {
34         Konsultasi konsultasi = new Konsultasi();
35         konsultasi.setTanggal(tanggal);
36         konsultasi.setDokter(dokter);
37         konsultasi.setPerawat(perawat);
38         riwayatKonsultasi.add(konsultasi);
39     }
40
41     public String getInfo() {
42         String info = "";
43         info += "No Rekam Medis : " + this.noRekamMedis + "\n";
44         info += "Nama          : " + this.nama + "\n";
45
46         if (!riwayatKonsultasi.isEmpty()) {
47             info += "\nRiwayat Konsultasi : \n";
48
49             for (Konsultasi konsultasi : riwayatKonsultasi) {
50                 info += konsultasi.getInfo();
51             }
52
53         } else {
54             info += "Belum ada riwayat konsultasi";
55         }
56
57         info += "\n";
58
59         return info;
60     }
61 }
```

Berikut merupakan kode program dari class konsultasi

```
1 package RumahSakit;
2 import java.time.LocalDate;
3
4 public class Konsultasi {
5     private LocalDate tanggal;
6     private Pegawai dokter;
7     private Pegawai perawat;
8
9
10    public LocalDate getTanggal() {
11        return tanggal;
12    }
13
14    public void setTanggal (LocalDate tanggal) {
15        this.tanggal = tanggal;
16    }
17
18    public Pegawai getDokter() {
19        return dokter;
20    }
21
22    public void setDokter(Pegawai dokter) {
23        this.dokter = dokter;
24    }
25
26    public Pegawai getPerawat() {
27        return perawat;
28    }
29
30    public void setPerawat(Pegawai perawat) {
31        this.perawat = perawat;
32    }
33
34    public String getInfo() {
35        String info = "";
36        info += "\tTanggal: " + tanggal;
37        info += "\tDokter: " + dokter.getInfo();
38        info += "\tPerawat: " + perawat.getInfo();
39        info += "\n";
40
41        return info;
42    }
43 }
```

Berikut merupakan kode program dari class RumahSakitDemo

```
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 }
```

Berikut merupakan output dari kode program di atas

```
Nama          : Puspa Widya
No Rekam Medis : 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)

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

D:\Jobsheet 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 ?

Method setter dan getter dalam kelas Pegawai, Pasien, dan Konsultasi memiliki peran penting dalam pengelolaan data dan enkapsulasi. Setter digunakan untuk mengatur atau memperbarui nilai dari atribut (variabel) yang bersifat privat, sehingga memungkinkan kontrol terhadap perubahan data. Dengan menggunakan setter, kita dapat menambahkan validasi atau logika tambahan sebelum menyimpan nilai baru. Di sisi lain, getter berfungsi untuk mengambil nilai dari atribut privat, sehingga memberikan akses kepada pengguna kelas tanpa mengizinkan modifikasi langsung.

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

kelas tersebut tetap memiliki constructor default yang disediakan secara otomatis oleh Java. Constructor default ini tidak menerima parameter dan tidak melakukan inisialisasi atribut apa pun. Namun, jika dilakukan inisialisasi atribut seperti tanggal, dokter, dan perawat saat membuat objek Konsultasi, maka perlu mendefinisikan constructor dengan parameter secara eksplisit. Tanpa constructor yang jelas, atribut tersebut akan tetap bernilai null (untuk objek) atau 0/false (untuk tipe primitif) sampai mengatur nilainya menggunakan setter atau metode lain.

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

Yang bertipe object meliputi pegawai dokter, dan juga pegawai perawat

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

Baris ini menunjukkan bahwa class Konsultasi memiliki dua atribut yang bertipe Pegawai, yaitu dokter dan perawat. Ini berarti bahwa class Konsultasi memiliki relasi dengan class Pegawai melalui atribut-atribut ini.

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

Method getInfo() ini kemungkinan besar digunakan untuk mengembalikan informasi tentang konsultasi yang dilakukan oleh pasien, seperti tanggal konsultasi, dokter yang menangani, perawat yang terlibat, dan lain-lain. Sehingga method getInfo() pada objek konsultasi, class Pasien dapat mengakses informasi tentang konsultasi yang dilakukan oleh pasien dan melakukan operasi yang diperlukan, seperti menampilkan informasi konsultasi atau melakukan analisis pada data konsultasi.

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

if (!riwayatKonsultasi.isEmpty())

Apakah yang dilakukan oleh baris tersebut?

Dengan menggunakan baris tersebut, kode akan memeriksa apakah pasien memiliki riwayat konsultasi. Jika daftar tidak kosong, maka kode di dalam blok if akan dieksekusi, yang biasanya mencakup menampilkan informasi tentang riwayat konsultasi. Jika daftar kosong, maka informasi terkait konsultasi tidak akan ditampilkan, dan dapat menampilkan pesan alternatif, seperti "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?**

Dengan menambahkan baris ini, dapat memastikan bahwa riwayatKonsultasi dimulai sebagai daftar kosong ketika objek Pasien dibuat. Ini penting agar dapat menggunakan metode seperti add() untuk menambahkan konsultasi ke dalam daftar tanpa mendapatkan NullPointerException.

Jika baris tersebut dihilangkan maka atribut riwayatKonsultasi tidak akan diinisialisasi, sehingga nilainya akan tetap null.

Akibatnya jika akan menambahkan konsultasi dengan menggunakan tambahKonsultasi maka harus mendapatkan NullPointerException karena tidak dapat memanggil method pada objek yang bernilai null, dan juga tidak bisa menyimpan atau mengelola riwayat konsultasi untuk pasien tersebut, yang akan menyebabkan fungsi dari kelas Pasien tidak berjalan dengan baik.

Tugas

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

Berikut merupakan kode program hasil implementasi dari tugas PBO teori.

Disini saya membuat 4 class, yaitu Class Pembeli, Class PegawaiKantin, Class Transaksi, dan juga Class KantinDemo.

1. Class Pembeli

```
1  package Kantin;
2
3  import java.util.ArrayList;
4
5  public class Pembeli {
6      private String idPembeli;
7      private String namaPembeli;
8      private ArrayList<Transaksi> transaksList;
9
10     //konstruktor pembeli
11     public Pembeli(String idPembeli, String namaPembeli) {
12         this.idPembeli = idPembeli;
13         this.namaPembeli = namaPembeli;
14         this.transaksList = new ArrayList<>();
15     }
16
17     public String getIdPembeli() {
18         return idPembeli;
19     }
20
21     public void setIdPembeli(String idPembeli) {
22         this.idPembeli = idPembeli;
23     }
24
25     public String getNamaPembeli() {
26         return namaPembeli;
27     }
28
29     public void setNamaPembeli(String namaPembeli) {
30         this.namaPembeli = namaPembeli;
31     }
32
33     public ArrayList<Transaksi> getTransaksList() {
34         return transaksList;
35     }
36
37     public void setTransaksList(ArrayList<Transaksi> transaksList) {
38         this.transaksList = transaksList;
39     }
40
41     //tambahkan transaksi
42     public void tambahTransaksi(Transaksi transaksi) {
43         transaksList.add(transaksi);
44     }
45
46     //hapus transaksi
47     public void batalkanPesanan(Transaksi transaksi) {
48         transaksList.remove(transaksi);
49     }
50 }
```

2. Class PegawaiKantin

```
1  package Kantin;
2
3  import java.util.ArrayList;
4
5  public class PegawaiKantin {
6      private String idPegawai;
7      private String namaPegawai;
8      private ArrayList<Transaksi> transaksList;
9
10     //konstruktor pegawai kantin
11     public PegawaiKantin(String idPegawai, String namaPegawai) {
12         this.idPegawai = idPegawai;
13         this.namaPegawai = namaPegawai;
14         this.transaksList = new ArrayList<>();
15     }
16
17     public String getIdPegawai() {
18         return idPegawai;
19     }
20
21     public void setIdPegawai(String idPegawai) {
22         this.idPegawai = idPegawai;
23     }
24
25     public String getNamaPegawai() {
26         return namaPegawai;
27     }
28
29     public void setNamaPegawai(String namaPegawai) {
30         this.namaPegawai = namaPegawai;
31     }
32
33     public ArrayList<Transaksi> getTransaksList() {
34         return transaksList;
35     }
36
37     public void setTransaksList(ArrayList<Transaksi> transaksList) {
38         this.transaksList = transaksList;
39     }
40
41     //tambahkan transaksi
42     public void layaniTransaksi(Transaksi transaksi) {
43         transaksList.add(transaksi);
44     }
45 }
```


3. Class Transaksi

```
1 package Kantin;
2
3 import java.text.DecimalFormat;
4 import java.util.ArrayList;
5
6 public class Transaksi {
7     private String idTransaksi;
8     private PegawaiKantin pegawai;
9     private Pembeli pembeli;
10    private ArrayList<String> items;
11    private int totalHarga;
12    private double uangPembayaran;
13
14    // Konstruktor untuk inisialisasi objek Transaksi
15    public Transaksi(String idTransaksi, PegawaiKantin pegawai, Pembeli pembeli) {
16        this.idTransaksi = idTransaksi;
17        this.pegawai = pegawai;
18        this.pembeli = pembeli;
19        this.items = new ArrayList<>();
20        this.totalHarga = 0;
21        this.uangPembayaran = 0;
22    }
23
24    public ArrayList<String>.getItems() {
25        return items;
26    }
27    public double getUangPembayaran() {
28        return uangPembayaran;
29    }
30
31    public void setUangPembayaran(double uangPembayaran) {
32        this.uangPembayaran = uangPembayaran;
33    }
34
35    public double hitungKembalian() {
36        return uangPembayaran - totalHarga;
37    }
38
39    public void hitungTotalHarga() {
40        totalHarga = items.size() * 20000;
41    }
42
43    @Override
44    public String toString() {
45        DecimalFormat df = new DecimalFormat("#,##0");
46        return "Transaksi ID      : " + idTransaksi + "\n" +
47            "Pembeli          : " + pembeli.getNamaPembeli() + "\n" +
48            "Total Harga     : Rp. " + df.format(totalHarga) + "\n" +
49            "Uang Pembayaran : Rp. " + df.format(uangPembayaran) + "\n" +
50            "Kembalian       : Rp. " + df.format(hitungKembalian()) + "\n";
51    }
52 }
```

4. Class KantinDemo

```
1 package Kantin;
2
3 public class KantinDemo {
4     public static void main(String[] args) {
5         // Membuat objek pegawai dan pembeli
6         PegawaiKantin pegawai1 = new PegawaiKantin("P01", "Niken");
7         Pembeli pembeli1 = new Pembeli("Pm01", "Annisa");
8
9         // Membuat transaksi
10        Transaksi transaksi1 = new Transaksi("TRX01", pegawai1, pembeli1);
11        transaksi1.getItems().add("Item 1");
12        transaksi1.getItems().add("Item 2");
13        transaksi1.hitungTotalHarga();
14
15        // Menambahkan uang pembayaran
16        transaksi1.setUangPembayaran(50000);
17
18        System.out.println("===== TRANSAKSI PEMBELIAN =====");
19        pegawai1.layaniTransaksi(transaksi1);
20        pembeli1.tambahTransaksi(transaksi1);
21        System.out.println(transaksi1);
22    }
23 }
```

Berikut merupakan output dari kode program di atas

```
===== TRANSAKSI PEMBELIAN =====
Transaksi ID    : TRX01
Pembeli         : Annisa
Total Harga     : Rp. 40,000
Uang Pembayaran : Rp. 50,000
Kembalian       : Rp. 10,000
```