

**Pemrograman Berorientasi Objek
Jobsheet 4**



Nama : Maulana Ahmad Bukhori
NIM : 244107060133
Kelas : SIB-2C
No.Absen : 13

**D IV –SISTEM INFORMASI BISNIS
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI MALANG
JL. Soekarno Hatta No.9 Kota Malang Kode Pos: 65141
2025**

Verifikasi hasil akhir percobaan

Praktikum :

```
Contoh tampilan dari RumahSakit.java dan outputnya akan 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	

Hasil percobaan :

```
Nama : Puspa Widya
No Rekam Medis : 343298
Riwayat Konsultasi :
    Tanggal: 2021-08-11, Dokter: dr. Ani (123), Perawat: Ns.Desi (123)
    Tanggal: 2021-09-15, Dokter: dr. Bagus (4567), Perawat: Ns.Eka (4567)

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

PS D:\Semester 3\Pbo\Praktikum\P4\RumahSakit> 
```

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** digunakan untuk memberikan atau mengubah nilai atribut yang bersifat *private*, sedangkan **method getter** digunakan untuk mengambil atau membaca nilai atribut tersebut. Dengan ini prinsip enkapsulasi tetap terjaga karena atribut tidak bisa diakses secara langsung dari luar kelas, melainkan harus melalui method yang disediakan.

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

Jawab : *Class* Konsultasi tetap memiliki constructor. Jika sebuah constructor tidak didefinisikan secara langsung, maka Java otomatis menyediakan default constructor (constructor tanpa parameter). Jadi walaupun tidak ditulis constructor akan tetap ada.

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

Jawab : Dokter(Pegawai),Perawat(Pegawai),Tanggal(Local Date)

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

Jawab :

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

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

Jawab : `konsultasi.getInfo()` dipakai untuk memanggil method `getInfo()` milik objek Konsultasi. Tujuannya adalah menampilkan informasi detail dari konsultasi pasien, seperti tanggal konsultasi, dokter, dan perawat yang terlibat.

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

Apakah yang dilakukan oleh baris tersebut?

Jawab : Baris tersebut berfungsi untuk memeriksa apakah daftar `riwayatKonsultasi` masih kosong atau sudah berisi data. Jika tidak kosong, maka informasi riwayat konsultasi akan ditampilkan. Jika kosong, artinya akan menampilkan pasien belum pernah melakukan konsultasi.

7. Pada constructor *class* Pasien, terdapat baris kode:

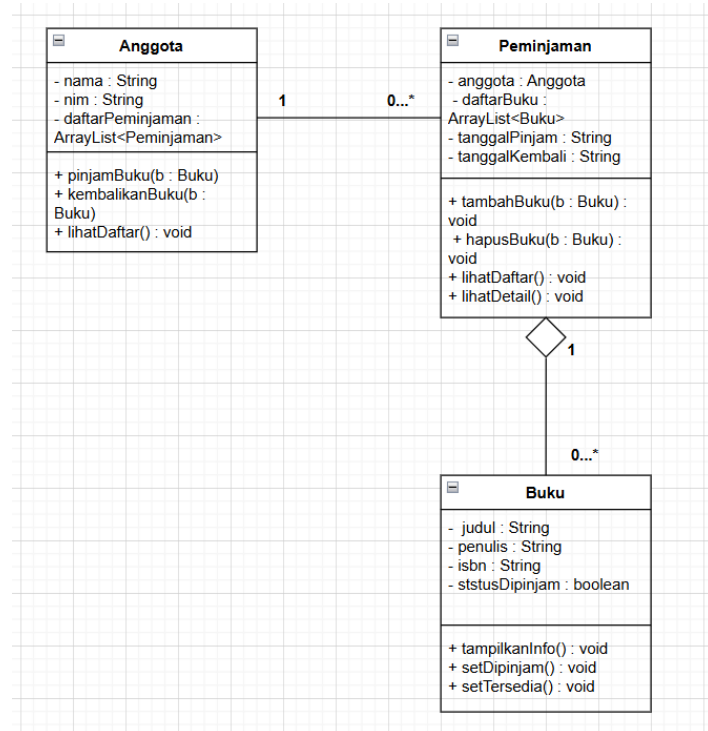
```
this.riwayatKonsultasi = new ArrayList<>();
```

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

Jawab : Baris kode ini digunakan untuk menginisialisasi `riwayatKonsultasi` sebagai sebuah objek `ArrayList` baru. Dengan begitu, setiap kali objek Pasien dibuat, sudah tersedia wadah untuk menyimpan data konsultasi. Apabila baris tersebut dihapus, maka atribut `riwayatKonsultasi` akan bernilai `null`. Akibatnya, saat program mencoba menambahkan data konsultasi, akan terjadi error `NullPointerException`.

Implementasikan studi kasus tugas teori kedalam program

Studi kasus



Jawab :

Source code

Class Anggota

```
1  import java.util.ArrayList;
   Windsurf: Refactor | Explain
2  public class Anggota {
3      String nama;
4      String nim;
5      ArrayList<Peminjaman> daftarPeminjaman;
6
7      public Anggota(String nama, String nim) {
8          this.nama = nama;
9          this.nim = nim;
10         this.daftarPeminjaman = new ArrayList<>();
11     }
12
13     Windsurf: Refactor | Explain | Generate Javadoc | X
14     public void pinjamBuku(Buku b, String tglPinjam, String tglKembali) {
15         Peminjaman peminjaman = new Peminjaman(this, tglPinjam, tglKembali);
16         peminjaman.tambahBuku(b);
17         daftarPeminjaman.add(peminjaman);
18     }
19
20     Windsurf: Refactor | Explain | Generate Javadoc | X
21     public void kembalikanBuku(Buku b) {
22         for (Peminjaman p : daftarPeminjaman) {
23             if (p.daftarBuku.contains(b)) {
24                 p.hapusBuku(b);
25                 break;
26             }
27         }
28     }
29
30     Windsurf: Refactor | Explain | Generate Javadoc | X
31     public void lihatDaftar() {
32         System.out.println("Daftar peminjaman " + nama + " :");
33         for (Peminjaman p : daftarPeminjaman) {
34             p.lihatDetail();
35         }
36     }
37 }
```

Class Peminjaman

```
Tugas > J Peminjaman.java > Language Support for Java(TM) by Red Hat > Peminjaman
1  import java.util.ArrayList;
   Windsurf: Refactor | Explain
2  public class Peminjaman {
3      Anggota anggota;
4      ArrayList<Buku> daftarBuku;
5      String tanggalPinjam;
6      String tanggalKembali;
7
8      public Peminjaman(Anggota anggota, String tanggalPinjam, String tanggalKembali) {
9          this.anggota = anggota;
10         this.tanggalPinjam = tanggalPinjam;
11         this.tanggalKembali = tanggalKembali;
12         this.daftarBuku = new ArrayList<>();
13     }
14     // Menambahkan buku ke daftar peminjaman
15     public void tambahBuku(Buku b) {
16         daftarBuku.add(b);
17         b.setDipinjam();
18     }
19     // Menghapus buku dari daftar peminjaman
20     public void hapusBuku(Buku b) {
21         daftarBuku.remove(b);
22         b.setTersedia();
23     }
24     public void lihatDetail(){
25         System.out.println("Peminjaman oleh: " + anggota.nama);
26         System.out.println("Tanggal Pinjam : " + tanggalPinjam);
27         System.out.println("Tanggal Kembali: " + tanggalKembali);
28         System.out.println("Daftar Buku: ");
29         for (Buku b : daftarBuku) {
30             System.out.println("- " + b.judul);
31         }
32     }
33 }
```

Class Buku

```
Tugas > J Buku.java > Language Support for Java(TM) by Red Hat > Buku
   Windsurf: Refactor | Explain
1  public class Buku {
2      String judul;
3      String penulis;
4      String isbn;
5      boolean statusDipinjam;
6      // Konstruktor
7      public Buku(String judul, String penulis, String isbn) {
8          this.judul = judul;
9          this.penulis = penulis;
10         this.isbn = isbn;
11         this.statusDipinjam = false; // Buku awalnya tidak dipinjam
12     }
13     public void tampilkanInfo() {
14         System.out.println("Judul: " + judul);
15         System.out.println("Penulis: " + penulis);
16         System.out.println("ISBN: " + isbn);
17         System.out.println("Status Dipinjam: " + (statusDipinjam ? "Ya" : "Tidak"));
18     }
19     public void setDipinjam() {
20         statusDipinjam=true;
21     }
22     public void setTersedia() {
23         statusDipinjam=false;
24     }
25 }
26
```

Class Main

```
Tugas > J MainJava > ...
Windsurf: Refactor | Explain
1 public class Main {
2     public static void main(String[] args) {
3         // Buat Buku
4         Buku b1 = new Buku(judul:"Pemrograman Java", penulis:"Pak Andi", isbn:"12345");
5         Buku b2 = new Buku(judul:"Database SQL", penulis:"Bu Sari", isbn:"67890");
6
7         // Buat Anggota
8         Anggota a1 = new Anggota(nama:"Budi", nim:"2141720001");
9
10        // Anggota pinjam buku
11        a1.pinjamBuku(b1, tglPinjam:"17-09-2025", tglKembali:"24-09-2025");
12        a1.pinjamBuku(b2, tglPinjam:"17-09-2025", tglKembali:"24-09-2025");
13
14        // Lihat daftar peminjaman anggota
15        a1.lihatDaftar();
16    }
17 }
18
```

Output

```
Daftar peminjaman Budi:
Peminjaman oleh: Budi
Tanggal Pinjam : 17-09-2025
Tanggal Kembali: 24-09-2025
Daftar Buku:
- Pemrograman Java
Peminjaman oleh: Budi
Tanggal Pinjam : 17-09-2025
Tanggal Kembali: 24-09-2025
Daftar Buku:
- Database SQL
PS D:\Semester 3\Pbo\Praktikum\P4\Tugas>
```