PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

JOBSHEET 4 - RELASI KELAS

ADHE WIDYA GALIH KARTIKA

SIB 2C_03

244107060067

Jurusan Teknologi Informasi

POLITEKNIK NEGERI MALANG

2025

class Pegawai

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

    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;
    }

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

class Pasien

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

public class Pasien {
    private String noRekamMedis;
    private String nama;
    private ArrayList<Konsultasi> riwayatKonsultasi;

    public String getNoRekamMedis() {
        return noRekamMedis;
    }

    public void setNoRekamMedis(String noRekamMedis) {
        this.noRekamMedis = noRekamMedis;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

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

    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;
    }

    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);
    }
}
```

class Konsultasi

```
import java.time.LocalDate;

public class Konsultasi {
    private LocalDate tanggal;
    private Pegawai dokter;
    private Pegawai perawat;

    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;
    }

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

        return info;
    }
}
```

class RumahSakitDemo

```java
import java.time.LocalDate;

public class RumahSakitdemo {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Pegawai ani = new Pegawai(nip:"1234", nama:"dr. Ani");
        Pegawai bagus = new Pegawai(nip:"4567", nama:"dr. Bagus");

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

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

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

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

Output

```
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
```

PERTANYAAN

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 *getter* digunakan untuk mengambil nilai dari atribut yang bersifat *private.* Karena atribut tersebut dideklarasikan dengan akses *private*, maka atribut tersebut tidak bisa diakses langsung dari luar class.
   - method *setter* digunakan untuk mengubah/mengisi nilai dari atribut yang bersifat *private.* Dengan *setter*, kita bisa mengontrol bagaimana nilai atribut diubah.

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, yaitu constructor default (tanpa parameter / kosong) yang otomatis dibuat oleh Java karena tidak mendefinisikan constructor secara eksplisit.

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

   Jawab :

   Semua atribut (tanggal, dokter, perawat) di *class* Konsultasi adalah objek

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

   Jawab :

```
5       private Pegawai dokter;
6       private Pegawai perawat;
```

   Pada baris tersebut *class* Konsultasi memiliki relasi dengan *class* Pegawai. Karena dokter dan perawat didefinisikan sebagai objek dari *class* Pegawai.

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

   Jawab :

   Kode tersebut mengembalikan sebuah String berisi detail informasi konsultasi. Jadi fungsi dari kode tersebut adalah untuk mengambil informasi detail dari setiap objek Konsultasi dalam daftar riwayatKonsultasi, lalu menambahkannya ke variabel info milik pasien.

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

   if (!riwayatKonsultasi.isEmpty())

   Apakah yang dilakukan oleh baris tersebut?

   Jawab :

   Kode tersebut memiliki arti jika riwayatKonsultasi tidak kosong / ada pasien, maka kode di dalamnya akan dijalankan dan menampilkan daftar konsultasi pasien. Sedangkan saat kosong maka akan masuk ke **else** dan menampilkan pesan “Belum ada riwayat konsultasi”.

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

   this.riwayatKonsultasi = new ArrayList<>();

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

   Jawab :

   Yang dilakukan baris kode tersebut adalah menginisialisasi atribut riwayatKonsultasi dengan ArrayList kosong setiap objek Pasien dibuat. Dengan begitu, objek Pasien sudah siap menyimpan daftar riwayat konsultasi tanpa error.

   Jika baris tersebut dihilangkan, atribut riwayatKonsultasi bernilai null, sehingga program akan error saat mencoba menambah atau membaca riwayat konsultasi.

TUGAS

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

class Anggota

```java
import java.util.ArrayList;

public class Anggota {
    private String nama;
    private String idAnggota;
    private ArrayList<Buku> bukuDipinjam;

    public Anggota(String nama, String idAnggota) {
        this.nama = nama;
        this.idAnggota = idAnggota;
        this.bukuDipinjam = new ArrayList<>();
    }

    public void pinjamBuku(Buku buku) {
        if (buku.isTersedia()) {
            buku.pinjam();
            bukuDipinjam.add(buku);
        } else {
            System.out.println("Buku " + buku.getJudul() + " tidak tersedia untuk dipinjam.");
        }
    }

    public void kembalikanBuku(Buku buku) {
        if (bukuDipinjam.contains(buku)) {
            buku.kembalikan();
            bukuDipinjam.remove(buku);
        } else {
            System.out.println("Buku " + buku.getJudul() + " tidak ditemukan dalam daftar pinjaman.");
        }
    }
```

```java
    public void tampilkanBukuDipinjam() {
        System.out.println("Buku yang dipinjam oleh " + nama + ":");
        if (bukuDipinjam.isEmpty()) {
            System.out.println(x:"Tidak ada buku yang dipinjam.");
        } else {
            for (Buku buku : bukuDipinjam) {
                System.out.println("- " + buku.tampilkanInfo());
            }
        }
    }

    public String toString() {
        return "Nama : " + nama + ", ID Anggota : " + idAnggota;
    }
}
```

class Buku

```java
public class Buku {
    private String judul;
    private String penulis;
    private boolean status;

    public Buku(String judul, String penulis) {
        this.judul = judul;
        this.penulis = penulis;
        this.status = true;
    }

    public void pinjam() {
        if (status) {
            status = false;
            System.out.println("Buku " + judul + " berhasil dipinjam.");
        } else {
            System.out.println("Buku " + judul + " sedang tidak tersedia untuk dipinjam.");
        }
    }

    public void kembalikan() {
        if (!status) {
            status = true;
            System.out.println("Buku " + judul + " berhasil dikembalikan.");
        } else {
            System.out.println("Buku " + judul + " tidak sedang dipinjam.");
        }
    }

    public String tampilkanInfo() {
        return "Judul : " + judul + ", Penulis : " + penulis + ", Status : " + (status ? "Tersedia" : "Dipinjam");
    }

    public boolean isTersedia() {
        return status;
    }

    public String getJudul() {
        return judul;
    }
}
```

class Perpustakaan

```java
import java.util.ArrayList;

public class Perpustakaan {
    private ArrayList<Anggota> daftarAnggota;
    private ArrayList<Buku> daftarBuku;

    public Perpustakaan() {
        daftarAnggota = new ArrayList<>();
        daftarBuku = new ArrayList<>();
    }

    public void tambahAnggota(Anggota anggota) {
        daftarAnggota.add(anggota);
    }

    public void tambahBuku(Buku buku) {
        daftarBuku.add(buku);
    }

    public void tampilkanData() {
        System.out.println(x:"=== Daftar Anggota : ===");
        for (Anggota anggota : daftarAnggota) {
            System.out.println(anggota);
            anggota.tampilkanBukuDipinjam();
        }

        System.out.println(x:"\n=== Daftar Buku : ===");
        for (Buku buku : daftarBuku) {
            System.out.println(buku.tampilkanInfo());
        }
    }
}
```

class MainPerpustakaan

```java
public class MainPerpustakaan {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Perpustakaan perpustakaan = new Perpustakaan();

        Anggota anggota1 = new Anggota(nama:"Alice", idAnggota:"A001");
        Anggota anggota2 = new Anggota(nama:"Bob", idAnggota:"B002");

        Buku buku1 = new Buku(judul:"Pemrograman Java", penulis:"John Doe");
        Buku buku2 = new Buku(judul:"Struktur Data", penulis:"Jane Smith");
        Buku buku3 = new Buku(judul:"Basis Data", penulis:"Mike Johnson");

        perpustakaan.tambahAnggota(anggota1);
        perpustakaan.tambahAnggota(anggota2);

        perpustakaan.tambahBuku(buku1);
        perpustakaan.tambahBuku(buku2);
        perpustakaan.tambahBuku(buku3);

        anggota1.pinjamBuku(buku1);
        anggota1.pinjamBuku(buku2);
        anggota2.pinjamBuku(buku3);

        perpustakaan.tampilkanData();

        anggota1.kembalikanBuku(buku1);
        anggota2.kembalikanBuku(buku3);

        System.out.println(x:"\nSetelah pengembalian buku :\n");
        perpustakaan.tampilkanData();
    }
}
```

Output

```
Buku Pemrograman Java berhasil dipinjam.
Buku Struktur Data berhasil dipinjam.
Buku Basis Data berhasil dipinjam.
=== Daftar Anggota : ===
Nama : Alice, ID Anggota : A001
Buku yang dipinjam oleh Alice:
- Judul : Pemrograman Java, Penulis : John Doe, Status : Dipinjam
- Judul : Struktur Data, Penulis : Jane Smith, Status : Dipinjam
Nama : Bob, ID Anggota : B002
Buku yang dipinjam oleh Bob:
- Judul : Basis Data, Penulis : Mike Johnson, Status : Dipinjam

=== Daftar Buku : ===
Judul : Pemrograman Java, Penulis : John Doe, Status : Dipinjam
Judul : Struktur Data, Penulis : Jane Smith, Status : Dipinjam
Judul : Basis Data, Penulis : Mike Johnson, Status : Dipinjam
Buku Pemrograman Java berhasil dikembalikan.
Buku Basis Data berhasil dikembalikan.

Setelah pengembalian buku :

=== Daftar Anggota : ===
Nama : Alice, ID Anggota : A001
Buku yang dipinjam oleh Alice:
- Judul : Struktur Data, Penulis : Jane Smith, Status : Dipinjam
Nama : Bob, ID Anggota : B002
Buku yang dipinjam oleh Bob:
Tidak ada buku yang dipinjam.

=== Daftar Buku : ===
Judul : Pemrograman Java, Penulis : John Doe, Status : Tersedia
Judul : Struktur Data, Penulis : Jane Smith, Status : Dipinjam
Judul : Basis Data, Penulis : Mike Johnson, Status : Tersedia
PS C:\PBO\Praktikum PBO 4>
```