

LAPORAN PRAKTIKUM

MATA KULIAH PEMROGRAMAN BERBASIS OBJEK

Dosen Pengampu : Vit Zuraida, S.Kom, M.Kom

JOBSHEET - 4



Nama : M. Zidna Billah Faza
NIM : 2341760030
Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

RELASI KELAS

A. Percobaan

- 1) Buatlah folder baru dengan nama RumahSakit
- 2) Buatlah class Pegawai. Tambahkan atribut nip dan nama pada class Pegawai dengan akses modifier private

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

- 3) Buatlah constructor untuk class Pegawai dengan parameter nip dan nama.

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

- 4) Implementasikan setter dan getter untuk class Pegawai.

```
    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;  
    }
```

- 5) Implementasikan method getInfo() sebagai berikut:

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

- 6) 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.

```
public class Pasien {  
  
    private String noRekamMedis;  
    private String nama;  
  
    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;  
    }  
}
```

- 7) Buatlah constructor untuk class Pasien dengan parameter noRekamMedis dan nama

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

- 8) Implementasikan method getInfo() sebagai berikut:

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

- 9) 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.
- 10) 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

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

- 11) 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;  
    }  
}
```

12) Implementasikan method getInfo() sebagai berikut:

```
public String getInfo() {  
    String info = "";  
  
    info += "Tanggal           : " + tanggal + "\n";  
    info += "Dokter           : " + dokter.getInfo() + "\n";  
    info += "Perawat           : " + perawat.getInfo() + "\n\n";  
  
    return info;  
}
```

13) 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;  
    private ArrayList<Konsultasi> riwayatKonsultasi;
```

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

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

15) 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);  
    riwayatKonsultasi.add(konsultasi);  
}
```

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

```
public String getInfo() {
    String info = "";

    info += "\nNo Rekam Medis      : " + this.noRekamMedis + "\n";
    info += "Nama                  : " + this.nama + "\n";

    if (!riwayatKonsultasi.isEmpty()) {
        info += "\nRIWAYAT KONSULTASI\n\n";

        for (Konsultasi konsultasi : riwayatKonsultasi) {
            info += konsultasi.getInfo();
        }
    } else {
        info += "Belum ada riwayat konsultasi";
    }

    info += "\n";

    return info;
}
```

17) Compile kemudian run RumahSakitDemo dan didapatkan hasil seperti berikut:

```
public class RumahSakitDemo {  
    Run | Debug  
    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(x:"=====");  
        System.out.println(pasien1.getInfo());  
        System.out.println(x:"=====");  
  
        Pasien pasien2 = new Pasien(noRekamMedis:"997744", nama:"Yenny Anggraeni");  
        System.out.println(pasien2.getInfo());  
    }  
}
```

```
=====
```

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 Percobaan

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 ?

Setter berfungsi untuk mengisi nilai attribute (write), sedangkan Getter berfungsi untuk mengambil atau membaca nilai attribute (read).

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

Pada class konsultasi terdapat class default namun tidak tertulis secara langsung.

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

Terdapat 3 objek yaitu :

- tanggal (LocalDate (bawaan))
- dokter (Pegawai)
- perawat (Pegawai)

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

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

Pada saat deklarasi atribut menggunakan class Pegawai

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

Pada method tersebut berfungsi menampilkan tanggal, dokter, dan perawat dr class konsultasi.

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

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

Apakah yang dilakukan oleh baris tersebut?

Kode tersebut mengecek arrayList apakah riwayat konsultasi dari pasien tersebut tidak kosong. Jika benar (tidak kosong) maka program didalam if akan dieksekusi.

- 7) Pada constructor class Pasien, terdapat baris kode:

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

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

Apabila baris tersebut dihilangkan maka pasien tidak akan memiliki riwayat konsultasi karena fungsi dari kode tersebut adalah agar setiap pasien memiliki riwayat konsultasi (disiapkan) meskipun belum berkonsultasi.

TUGAS

1) Program pada class Karyawan

```
public class Karyawan {  
  
    private int noKaryawan;  
    private String nama;  
    private String noTelp;  
  
    public Karyawan(int noKaryawan, String nama, String noTelp) {  
        this.noKaryawan = noKaryawan;  
        this.nama = nama;  
        this.noTelp = noTelp;  
    }  
  
    public int getNoKaryawan() {  
        return noKaryawan;  
    }  
  
    public void setNoKaryawan(int noKaryawan) {  
        this.noKaryawan = noKaryawan;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
  
    public String getNoTelp() {  
        return noTelp;  
    }  
  
    public void setNoTelp(String noTelp) {  
        this.noTelp = noTelp;  
    }  
  
    public String getInfo() {  
        String info = "";  
  
        info += nama + " - " + noKaryawan;  
  
        return info;  
    }  
}
```

2) Program pada class Pelanggan

```
package Jobsheet04.srcTugas;

import java.util.ArrayList;

public class Pelanggan {

    private int noPelanggan;
    private String nama;
    private String alamat;
    private ArrayList<Transaksi> riwayatTransaksi;

    public Pelanggan(int noPelanggan, String nama, String alamat) {
        this.noPelanggan = noPelanggan;
        this.nama = nama;
        this.alamat = alamat;
        this.riwayatTransaksi = new ArrayList<Transaksi>();
    }

    public int getNoPelanggan() {
        return noPelanggan;
    }

    public void setNoPelanggan(int noPelanggan) {
        this.noPelanggan = noPelanggan;
    }

    public String getName() {
        return nama;
    }

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

    public String getAddress() {
        return alamat;
    }

    public void setAddress(String alamat) {
        this.alamat = alamat;
    }

    public void tambahTransaksi(double berat, Karyawan kasir, Karyawan petugasCuci) {
        Transaksi transaksi = new Transaksi();
        transaksi.setBerat(berat);
        transaksi.setKasir(kasir);
        transaksi.setPetugasCuci(petugasCuci);
        riwayatTransaksi.add(transaksi);
    }

    public String getInfo() {
        String info = "";

        info += "No Pelanggan      : " + this.noPelanggan;
        info += "\nNama Pelanggan      : " + this.nama;

        if (!riwayatTransaksi.isEmpty()) {

            info += "\n=====\\n";
            info += "                RIWAYAT TRANSAKSI";
            info += "\n=====\\n";

            for (Transaksi transaksi : riwayatTransaksi) {
                info += transaksi.getInfo();
                info += "\n-----\\n";
            }

        } else {

            info += "\n\\nMaaf kamu belum melakukan transaksi";

        }

        return info;
    }
}
```

3) Program pada class Transaksi

```
public class Transaksi {  
  
    private Karyawan kasir;  
    private Karyawan petugasCuci;  
    private double berat;  
  
    public Karyawan getKasir() {  
        return kasir;  
    }  
  
    public void setKasir(Karyawan kasir) {  
        this.kasir = kasir;  
    }  
  
    public Karyawan getPetugasCuci() {  
        return petugasCuci;  
    }  
  
    public void setPetugasCuci(Karyawan petugasCuci) {  
        this.petugasCuci = petugasCuci;  
    }  
  
    public double getBerat() {  
        return berat;  
    }  
  
    public void setBerat(double berat) {  
        this.berat = berat;  
    }  
  
    public String getInfo() {  
        String info = "";  
  
        info += "\nBerat          : " + berat;  
        info += "\nKasir          : " + kasir.getInfo();  
        info += "\nPetugas Cuci   : " + petugasCuci.getInfo();  
  
        return info;  
    }  
}
```

4) Program pada class LaundryDemo

```
public class LaundryDemo {  
    Run | Debug  
    public static void main(String[] args) {  
  
        Karyawan ola = new Karyawan(noKaryawan:31, nama:"Ola", noTelp:"08563243824");  
        Karyawan umar = new Karyawan(noKaryawan:32, nama:"Umar", noTelp:"08573829742");  
  
        Karyawan imam = new Karyawan(noKaryawan:41, nama:"Imam", noTelp:"08562414342");  
        Karyawan ajeng = new Karyawan(noKaryawan:42, nama:"Ajeng", noTelp:"08574324325");  
  
        Pelanggan p1 = new Pelanggan(noPelanggan:4231, nama:"Willy", alamat:"Jl. Senggani");  
        p1.tambahTransaksi(berat:3.5, imam, umar);  
        p1.tambahTransaksi(berat:2., ajeng, ola);  
        System.out.println(x:"=====");  
        System.out.println(p1.getInfo());  
        System.out.println(x:"=====");  
  
        Pelanggan p2 = new Pelanggan(noPelanggan:4231, nama:"Willy", alamat:"Jl. Senggani");  
        p2.tambahTransaksi(berat:1.8, imam, ola);  
        System.out.println(p2.getInfo());  
        System.out.println(x:"=====");  
  
        Pelanggan p3 = new Pelanggan(noPelanggan:4231, nama:"Willy", alamat:"Jl. Senggani");  
        System.out.println(p3.getInfo());  
        System.out.println(x:"=====");  
    }  
}
```

5) Output atau hasil

```
=====  
No Pelanggan      : 4231  
Nama Pelanggan    : Willy  
=====
```

RIWAYAT TRANSAKSI

```
=====  
Berat             : 3.5  
Kasir             : Imam - 41  
Petugas Cuci      : Umar - 32  
-----
```

```
Berat             : 2.0  
Kasir             : Ajeng - 42  
Petugas Cuci      : Ola - 31  
-----
```

```
=====  
No Pelanggan      : 4231  
Nama Pelanggan    : Willy  
=====
```

RIWAYAT TRANSAKSI

```
=====  
Berat             : 1.8  
Kasir             : Imam - 41  
Petugas Cuci      : Ola - 31  
-----
```

```
=====  
No Pelanggan      : 4231  
Nama Pelanggan    : Willy  
  
Maaf kamu belum melakukan transaksi  
=====
```