Job Sheet 04 - Relation Class

I. Competence

After to go through main discussion This, student capable:

1. Understand draft relation class;
2. Implementing association relationship to in program.

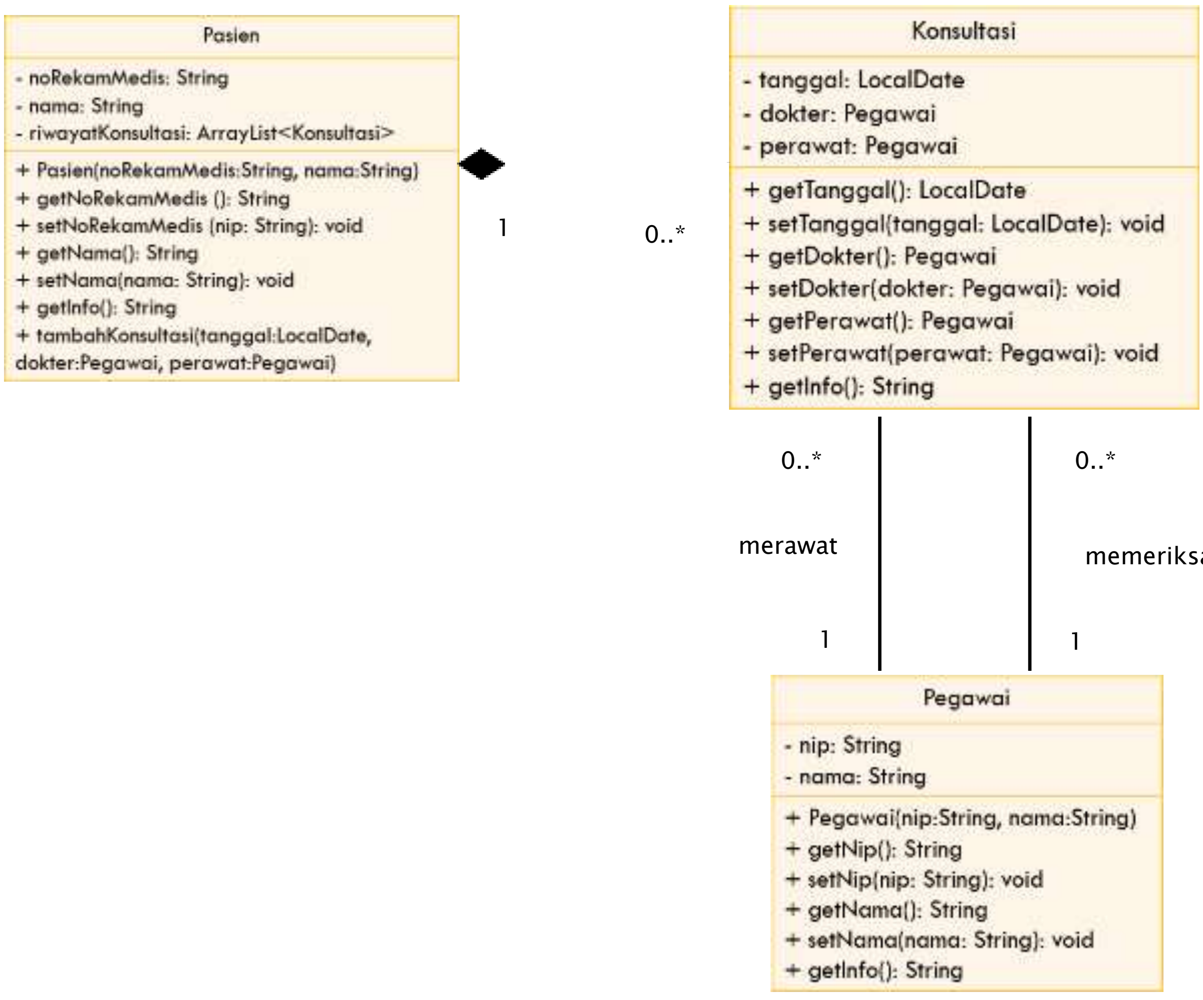
II. Introduction

In more complex cases, more than one will be found in a system. *class* Which each other own relatedness between *class* One with Which other. On previous experiments, the majority of cases that have been worked on only focus on One *class* just. On job sheet This will done test Which involving a number of *class* Which each other related.

III. Practicum

On practical work This will developed a system information House Sick Whichsave data history patient consultation.

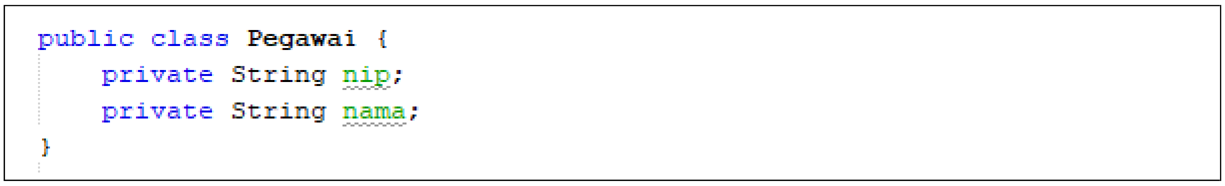
Take note *class* diagram following:



Pasien
- noRekamMedis: String
- nama: String
- riwayatKonsultasi: ArrayList<Konsultasi>
+ Pasien(noRekamMedis:String, nama:String)
+ getNoRekamMedis (): String
+ setNoRekamMedis (nip: String): void
+ getNama(): String
+ setNama(nama: String): void
+ getInfo(): String
+ tambahKonsultasi(tanggal:LocalDate, dokter:Pegawai, perawat:Pegawai)
1
0..*
Konsultasi
- tanggal: LocalDate
- dokter: Pegawai
- perawat: Pegawai
+ getTanggal(): LocalDate
+ setTanggal(tanggal: LocalDate): void
+ getDokter(): Pegawai
+ setDokter(dokter: Pegawai): void
+ getPerawat(): Pegawai
+ setPerawat(perawat: Pegawai): void
+ getInfo(): String
0..*
merawat
1
0..*
memeriksa
1
Pegawai
- nip: String
- nama: String
+ Pegawai(nip:String, nama:String)
+ getNip(): String
+ setNip(nip: String): void
+ getNama(): String
+ setNama(nama: String): void
+ getInfo(): String

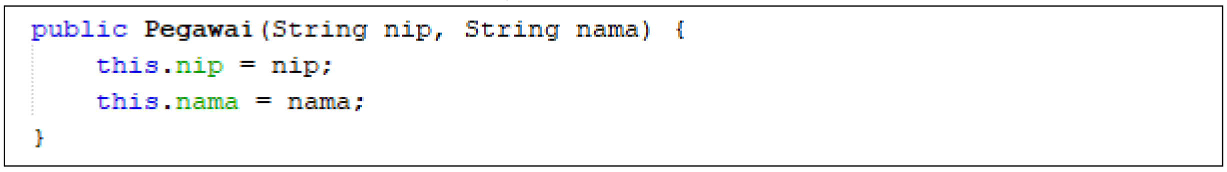
a. Make it folder new with Name Hospital

b. Make it class Employee. Add attribute nip And Name on class Employee withaccess modifier private



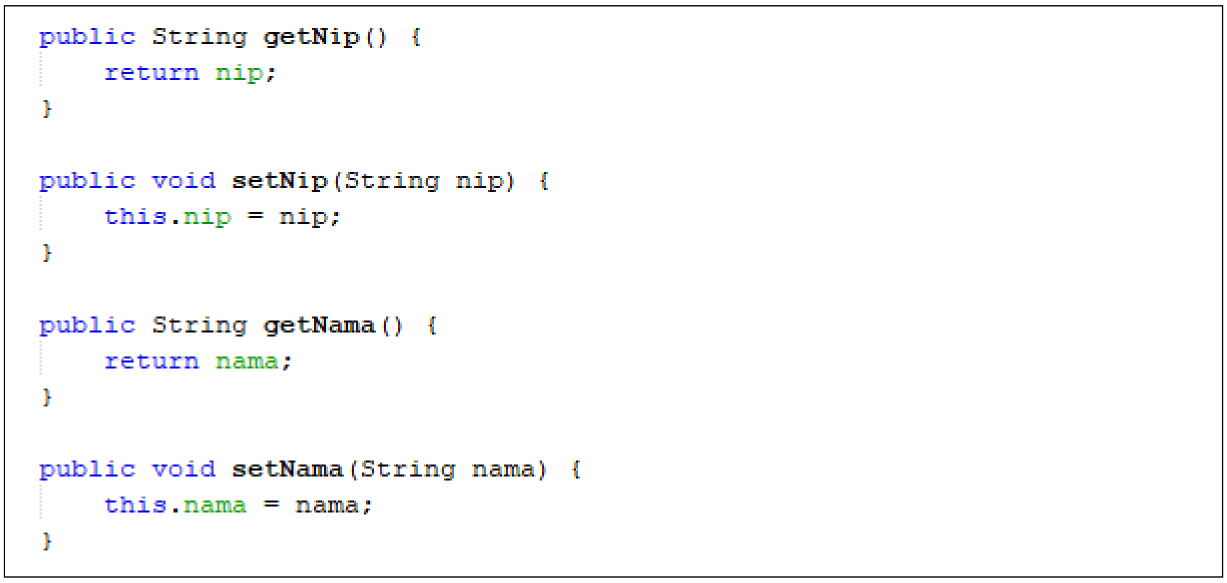
```
public class Pegawai {
    private String nip;
    private String nama;
}
```

c. Make it *constructor* For class Employee with nip parameters And Name.



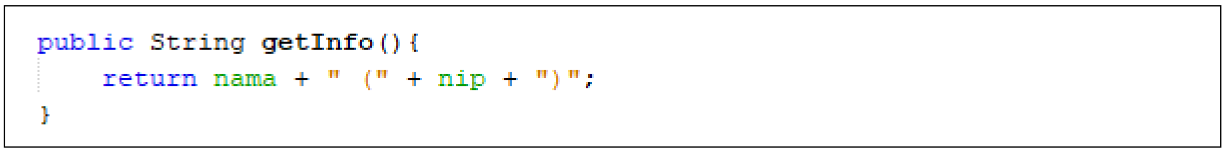
```
public Pegawai(String nip, String nama) {
    this.nip = nip;
    this.nama = nama;
}
```

d. Implement **setter** And **getter** For class Employee.

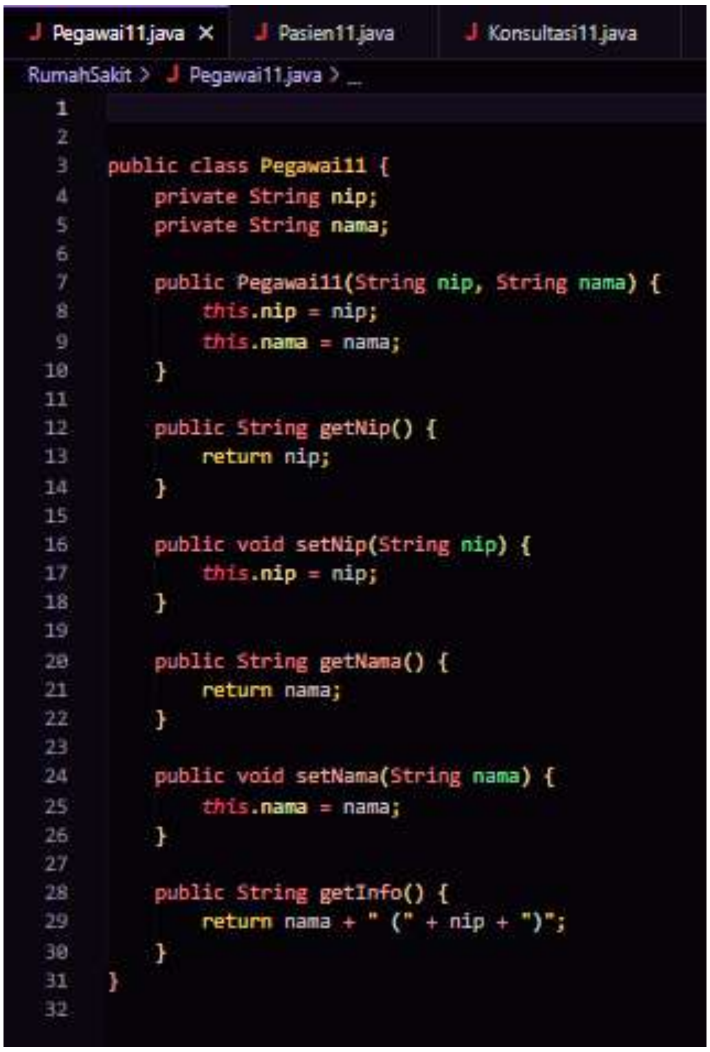


```
public String getNip() {
    return nip;
}

public void setNip(String nip) {
    this.nip = nip;
}

public String getNama() {
    return nama;
}

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

e. Implement *method* getInfo()as following:



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



```java


public class Pegawai11 {
    private String nip;
    private String nama;

    public Pegawai11(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 + ")";
    }
}
```

f. Furthermore make it class Patient Then add attribute noMedicalRecord Andname in the Patient class with the access level modifier private. Also provide setters and getter for second attribute the.

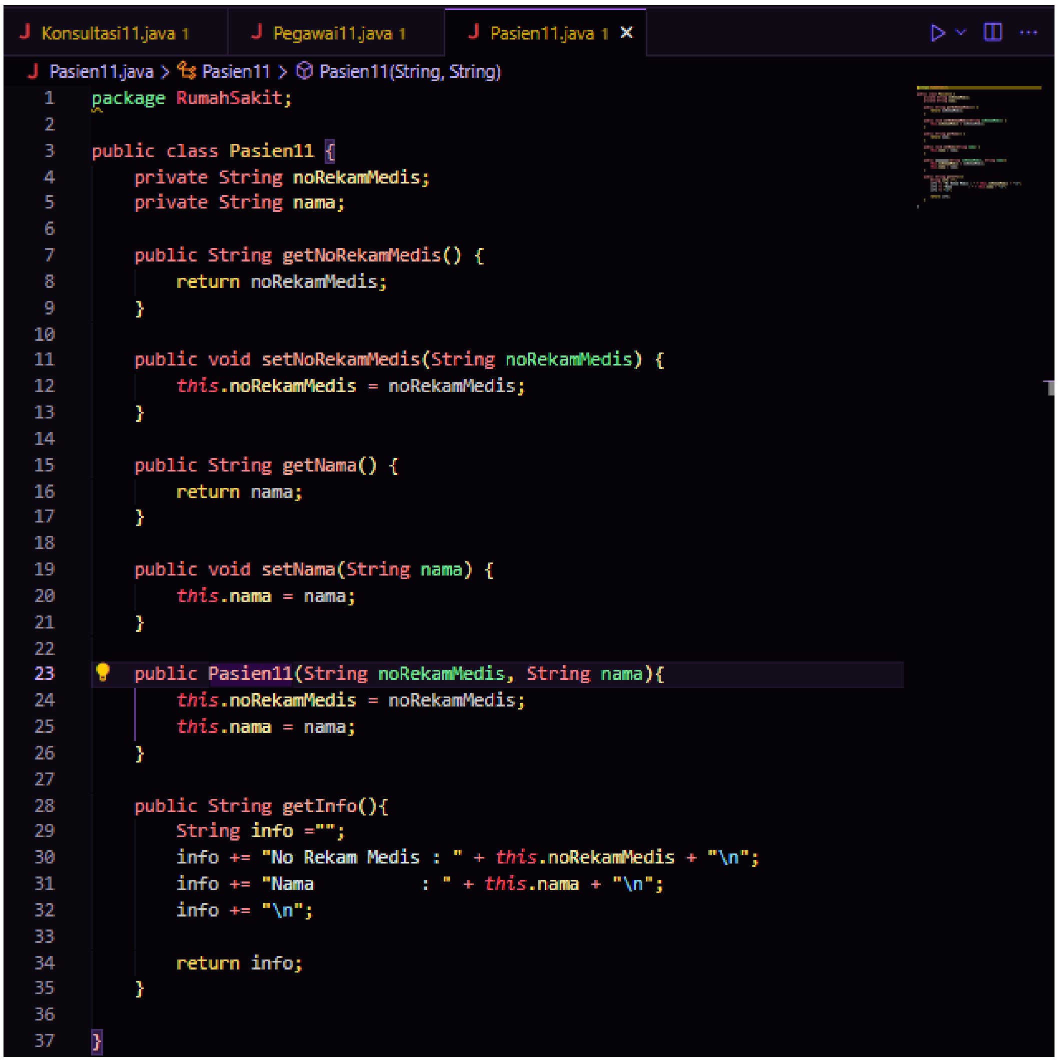
```java
public class Pasien {
    private String noRekamMedis;
    private String nama;

    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;
    }
}
```

g. Make it constructor For class Patient with noRekorMedic parameter And Name

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

h. Implement *method* getInfo()as following:

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

    return info;
}
```



```java
package RumahSakit;

public class Pasien11 {
    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;
    }

    public Pasien11(String noRekamMedis, String nama){
        this.noRekamMedis = noRekamMedis;
        this.nama = nama;
    }

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

        return info;
    }

}
```

i. This system will store data on every consultation that the patient has. Patients can conduct consultations more than once. Therefore, the consultation data will be stored in ArrayList form from objects Which type Consultation.

j. Make it class with Name Consultation with attribute date type LocalDate, doctor is of type Employee, and nurse is of type Employee. Set access level modifier privateFor all over attribute. Do import `java.time.LocalDate` so that can to declare attribute date type LocalDate.

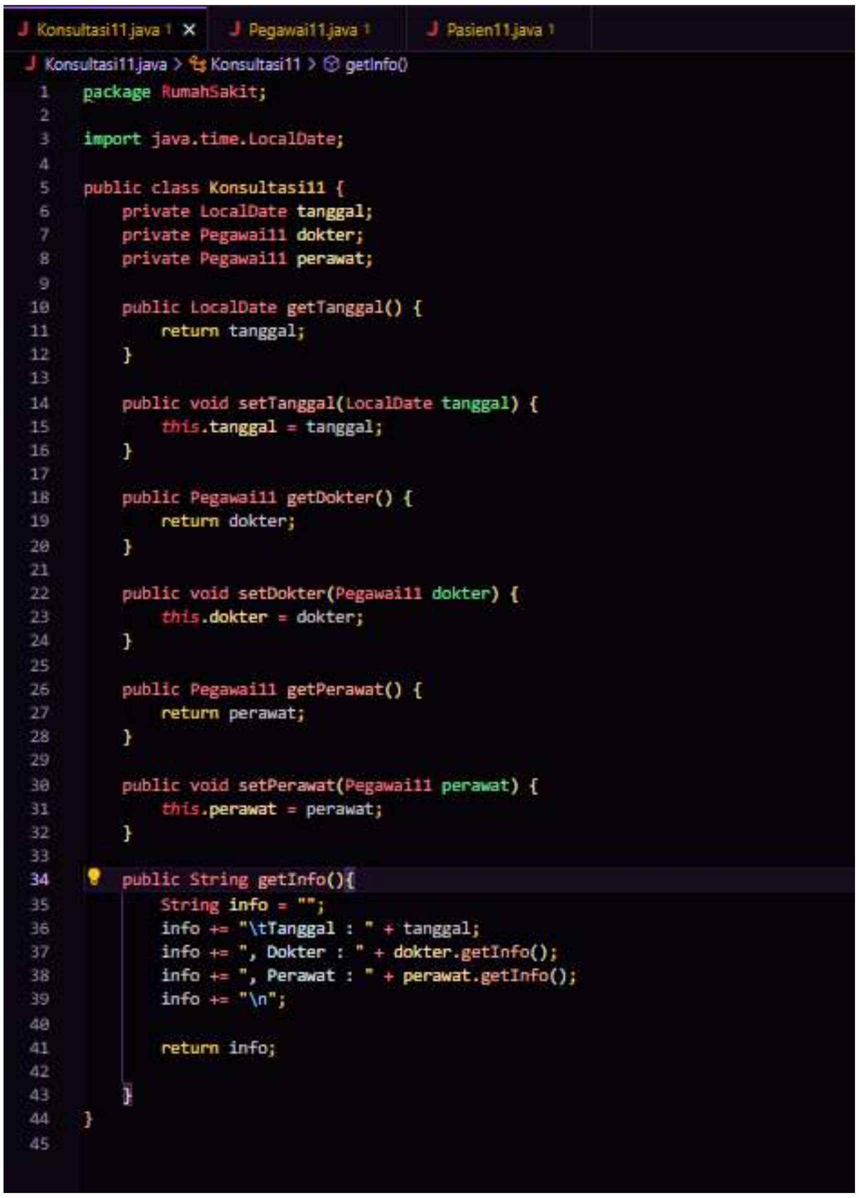
```java
import java.time.LocalDate;

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

k. Provide setter And getter For each attribute on class Consultation

```java
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;
}
```

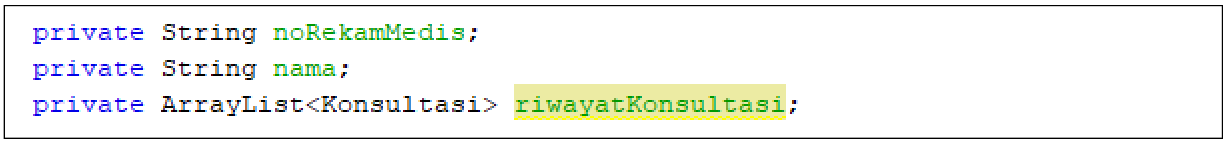
l. Implement method getInfo() as following:

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

    return info;
}
```



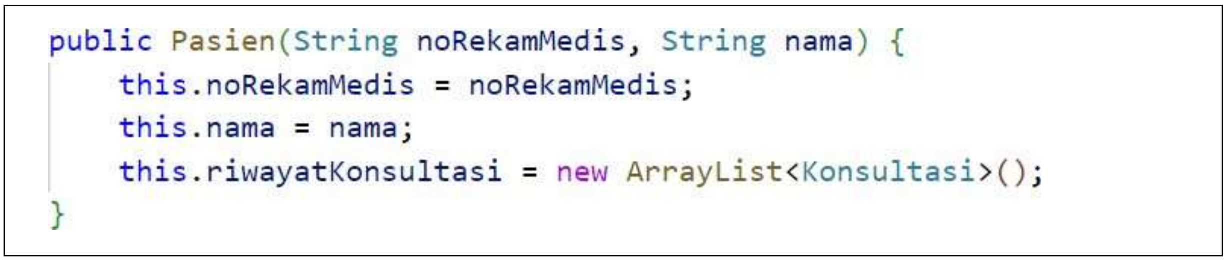
```java
package RumahSakit;

import java.time.LocalDate;

public class Konsultasi11 {
    private LocalDate tanggal;
    private Pegawai11 dokter;
    private Pegawai11 perawat;

    public LocalDate getTanggal() {
        return tanggal;
    }

    public void setTanggal(LocalDate tanggal) {
        this.tanggal = tanggal;
    }

    public Pegawai11 getDokter() {
        return dokter;
    }

    public void setDokter(Pegawai11 dokter) {
        this.dokter = dokter;
    }

    public Pegawai11 getPerawat() {
        return perawat;
    }

    public void setPerawat(Pegawai11 perawat) {
        this.perawat = perawat;
    }

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

        return info;

    }
}
```

m. For keep data history consultation patient, so add attribute historyConsultation on class Patient with type arrayList<Consultation>. Attribute This will store a series of objects of type Consultation. Import java.util.ArrayList so that can to declare attribute type ArrayList of object.



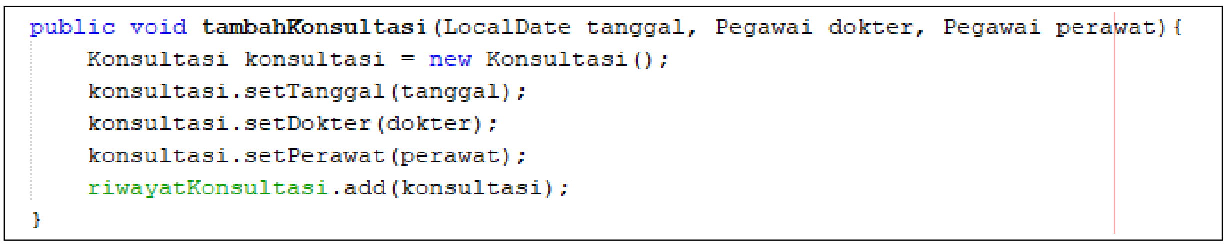
```
private String noRekamMedis;
private String nama;
private ArrayList<Konsultasi> riwayatKonsultasi;
```

n. Make it constructor parameterized For class Patient. Initiation mark attribute noRekamMedis and name based on name attribute. Instantiate/create new ArrayList For attribute historyConsultation;



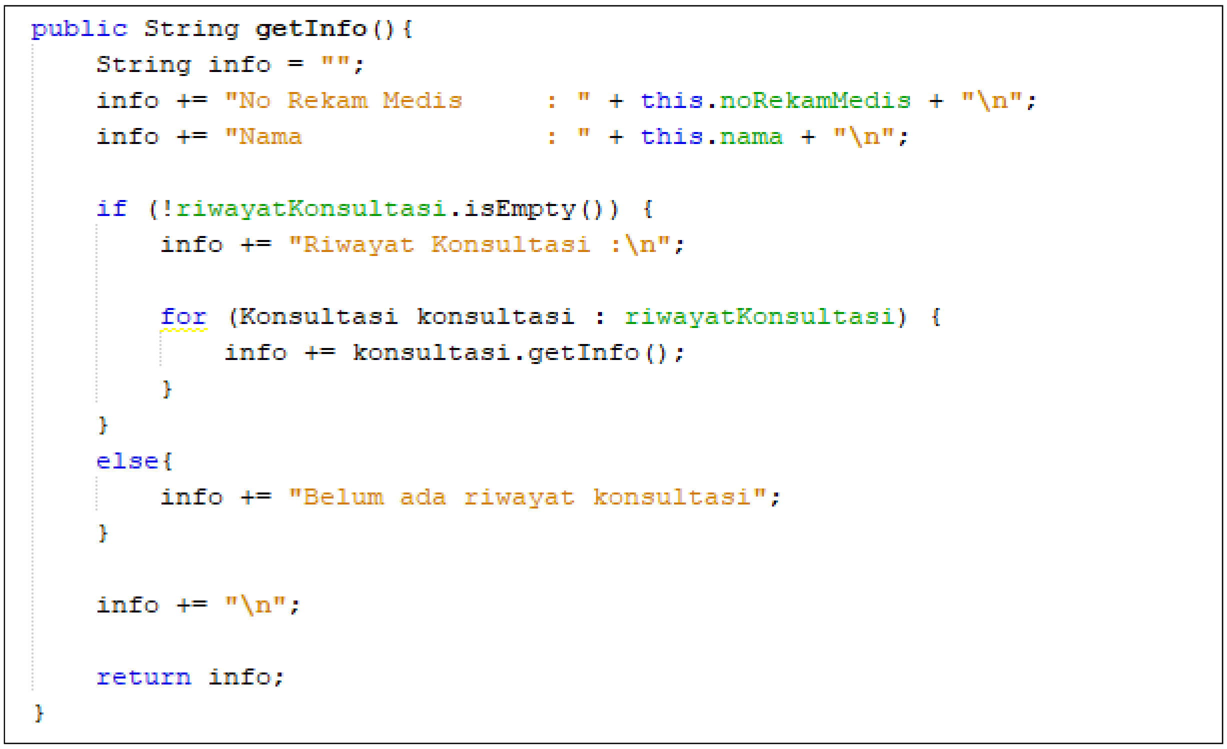
```
public Pasien(String noRekamMedis, String nama) {
    this.noRekamMedis = noRekamMedis;
    this.nama = nama;
    this.riwayatKonsultasi = new ArrayList<Konsultasi>();
}
```

o. Do import java.time.LocalDate so that can to declare attribute date type LocalDate on class Patient. Furthermore, implement method addConsultation() as follows:

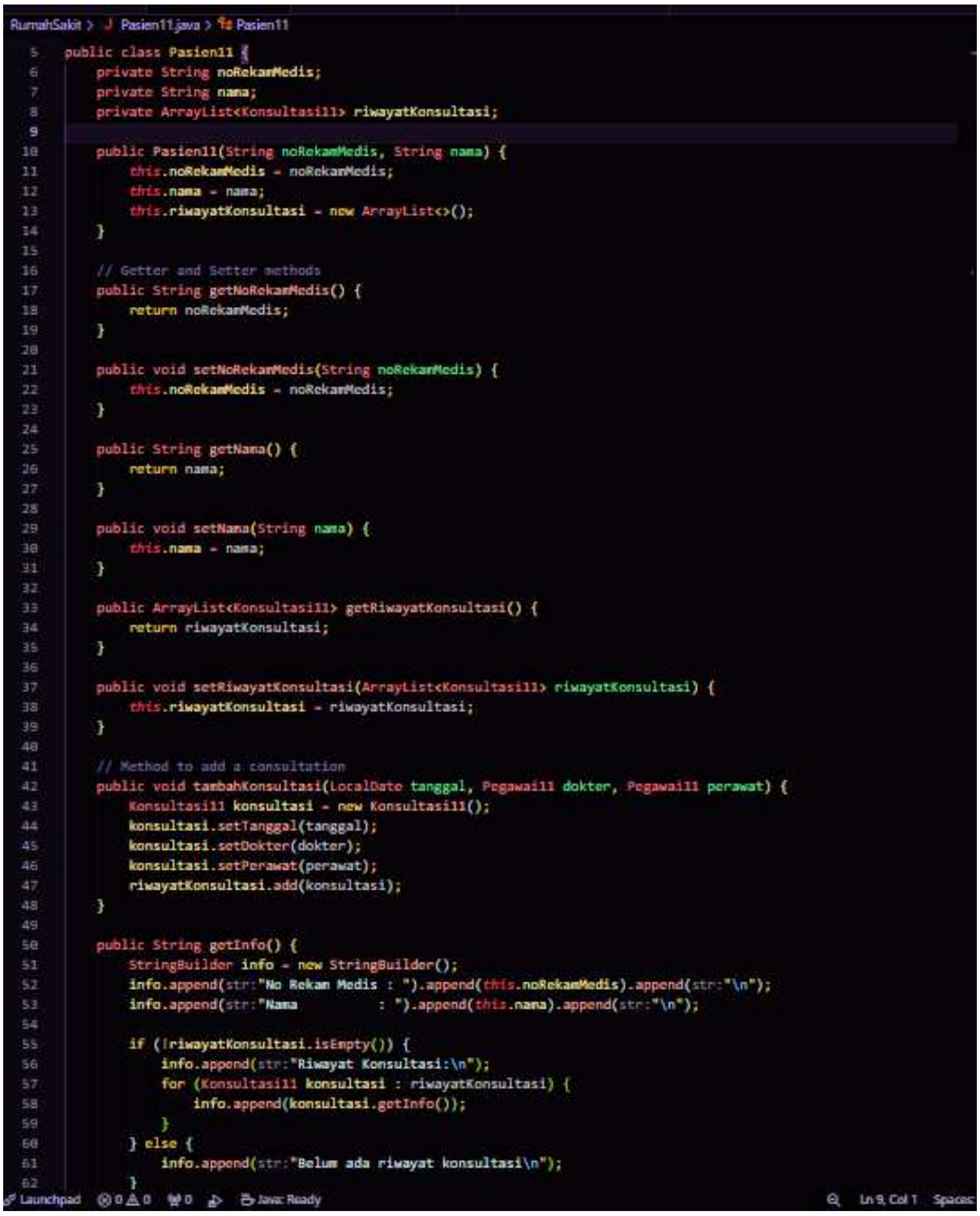


```
public void tambahKonsultasi(LocalDate tanggal, Pegawai dokter, Pegawai perawat){
    Konsultasi konsultasi = new Konsultasi();
    konsultasi.setTanggal(tanggal);
    konsultasi.setDokter(dokter);
    konsultasi.setPerawat(perawat);
    riwayatKonsultasi.add(konsultasi);
}
```

p. Modification method getInfo() For return info patient And list consultationever done



```
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;
}
```

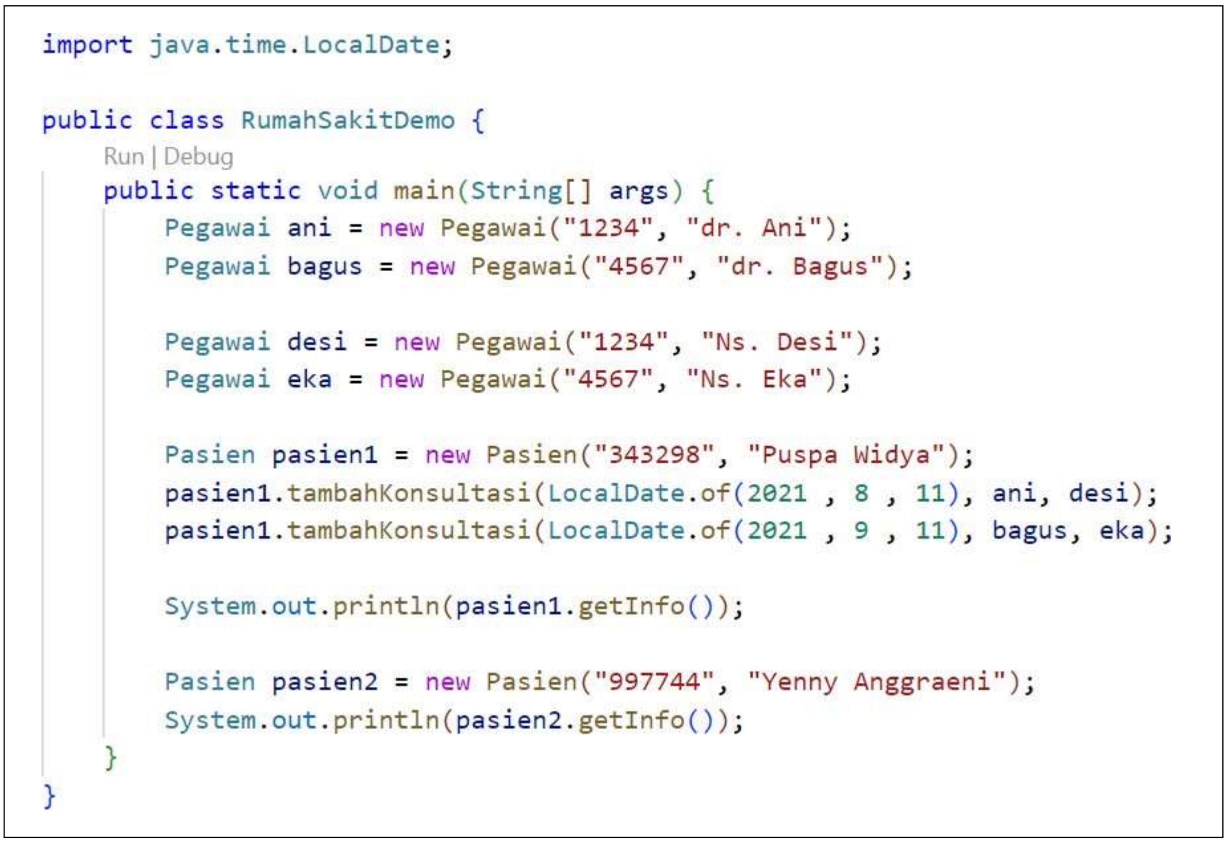


RumahSakit > Pasien11.java > Pasien11

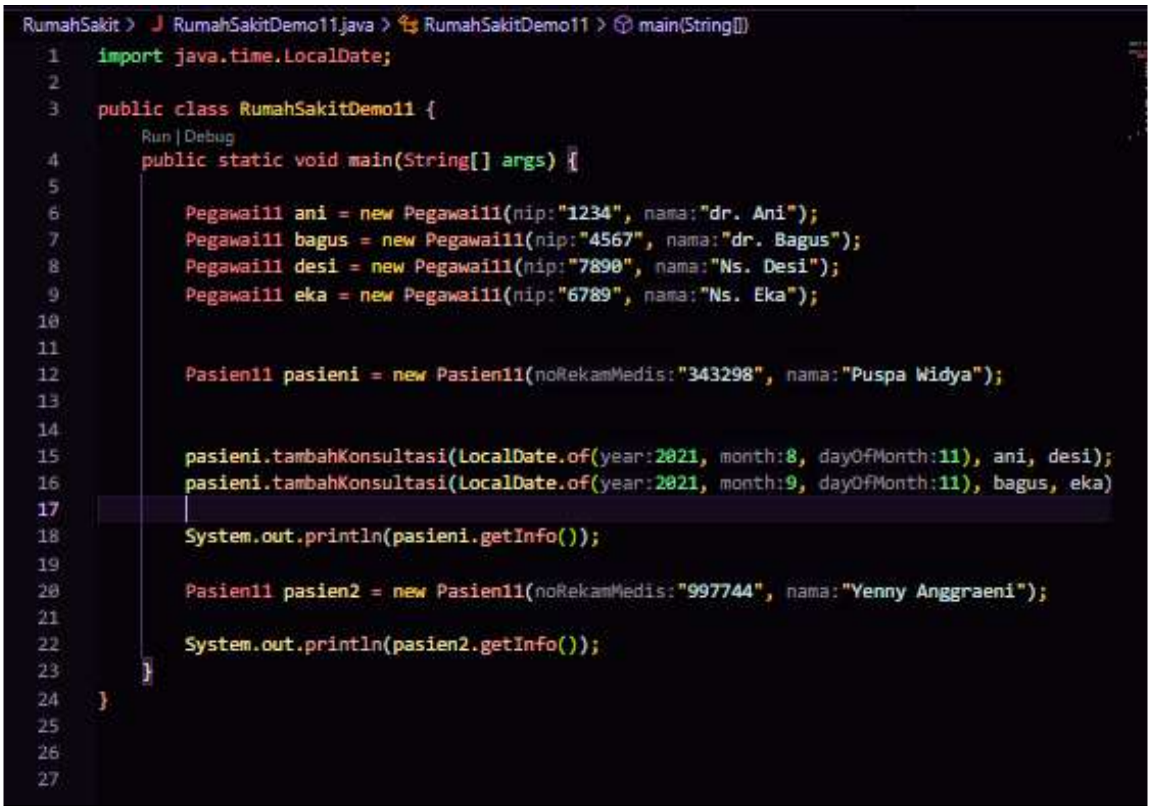
```java
public class Pasien11 {
    private String noRekamMedis;
    private String nama;
    private ArrayList<Konsultasi11> riwayatKonsultasi;

    public Pasien11(String noRekamMedis, String nama) {
        this.noRekamMedis = noRekamMedis;
        this.nama = nama;
        this.riwayatKonsultasi = new ArrayList<>();
    }

    // Getter and Setter methods
    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 ArrayList<Konsultasi11> getRiwayatKonsultasi() {
        return riwayatKonsultasi;
    }

    public void setRiwayatKonsultasi(ArrayList<Konsultasi11> riwayatKonsultasi) {
        this.riwayatKonsultasi = riwayatKonsultasi;
    }

    // Method to add a consultation
    public void tambahKonsultasi(LocalDate tanggal, Pegawai11 dokter, Pegawai11 perawat) {
        Konsultasi11 konsultasi = new Konsultasi11();
        konsultasi.setTanggal(tanggal);
        konsultasi.setDokter(dokter);
        konsultasi.setPerawat(perawat);
        riwayatKonsultasi.add(konsultasi);
    }

    public String getInfo() {
        StringBuilder info = new StringBuilder();
        info.append(str:"No Rekam Medis : ").append(this.noRekamMedis).append(str:"\n");
        info.append(str:"Nama           : ").append(this.nama).append(str:"\n");

        if (!riwayatKonsultasi.isEmpty()) {
            info.append(str:"Riwayat Konsultasi:\n");
            for (Konsultasi11 konsultasi : riwayatKonsultasi) {
                info.append(konsultasi.getInfo());
            }
        } else {
            info.append(str:"Belum ada riwayat konsultasi\n");
        }
```

Launchpad ⊗ 0 ⚠ 0 Java: Ready Ln 9, Col 1 Spaces

q. Do import java.time.LocalDate so that can to declare attribute date type LocalDate on class HospitalDemo. Test program Which Already made with make objects on class HospitalDemo. Instantiation object new type Employee with Name ani use constructor Employee(String nip, String Name) with mark argument nip “1234” And Name “dr. Ani”. Continue instantiation object as follows:



```java
import java.time.LocalDate;

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

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

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

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

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

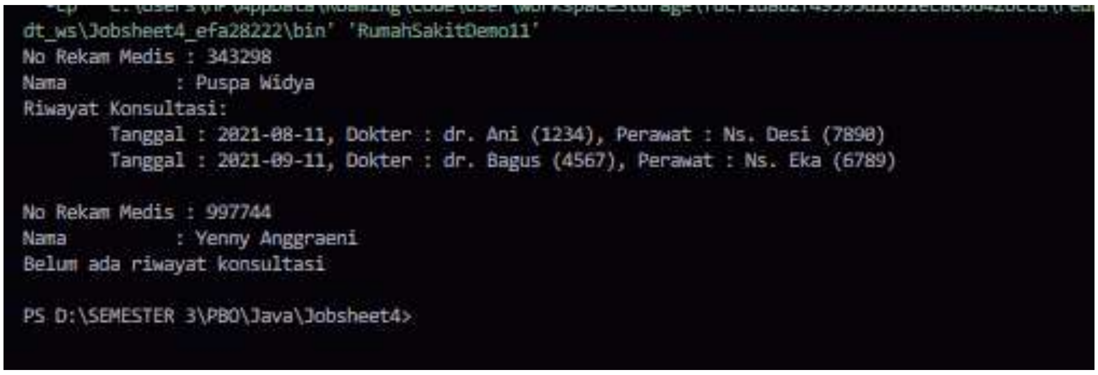


RumahSakit > RumahSakitDemo11.java > RumahSakitDemo11 > main(String[])

```java
import java.time.LocalDate;

public class RumahSakitDemo11 {
    Run | Debug
    public static void main(String[] args) {

        Pegawai11 ani = new Pegawai11(nip:"1234", nama:"dr. Ani");
        Pegawai11 bagus = new Pegawai11(nip:"4567", nama:"dr. Bagus");
        Pegawai11 desi = new Pegawai11(nip:"7890", nama:"Ns. Desi");
        Pegawai11 eka = new Pegawai11(nip:"6789", nama:"Ns. Eka");


        Pasien11 pasieni = new Pasien11(noRekamMedis:"343298", nama:"Puspa Widya");


        pasieni.tambahKonsultasi(LocalDate.of(year:2021, month:8, dayOfMonth:11), ani, desi);
        pasieni.tambahKonsultasi(LocalDate.of(year:2021, month:9, dayOfMonth:11), bagus, eka)

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

        Pasien11 pasien2 = new Pasien11(noRekamMedis:"997744", nama:"Yenny Anggraeni");

        System.out.println(pasien2.getInfo());
    }
}
```

r. *Compile* Then *run* HospitalDemo and obtained results like following:

```
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
```



```
dt_ws\Jobsheet4_efa28222\bin' 'RumahSakitDemo11'
No Rekam Medis : 343298
Nama           : Puspa Widya
Riwayat Konsultasi:
        Tanggal : 2021-08-11, Dokter : dr. Ani (1234), Perawat : Ns. Desi (7890)
        Tanggal : 2021-09-11, Dokter : dr. Bagus (4567), Perawat : Ns. Eka (6789)

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

PS D:\SEMESTER 3\PBO\Java\Jobsheet4>
```

Question

Based on test 1, answer it questions Which related:

1. In in *class* Employee, Patient, And Consultation, there is method *setter* And *getter* For each its attributes. Whether the use *of method setter* And *getter* the ?

   - **Employee Class:**
     Getter: To retrieve the nip value and employee name.
     Setter: To change the employee's NIP and name values, and can add validity checks.
   - **Patient Class:**
     Getter: To retrieve the Medical Record number and patient name values.
     Setter: To change the Medical Record number and patient name values, and can check the validity of the data.
   - **Consultation Class:**
     Getter: To retrieve the values of date, doctor, and nurse consultation.
     Setter: To change the values of the date, doctor, and nurse consultation, and can adjust the data if necessary.

2. In in *class* Consultation No in a way explicit there is constructor withparameter. Is this means class Consultation No own constructor?

   Yes, the `Consulting class` still has a constructor, even though it is not explicitly defined. Java automatically provides a default constructor with no parameters.

3. Take note *class* Consultation, attribute where just Which type *object* ?
   A attribute of type object :

    - **doctor**
    - **nurse**

4. Take note *class* Consultation, on line which one Which show that *class* Consultation own relation with *class* Employee?

   In the Consultation class, the relationship with the Employee class is seen in the line :

   - **private Doctor's staff;**
   - **private Nursing staff;**
   - **This line shows that Consultation uses the Employee class for the doctor and nurse attributes, so there is a relationship between Consultation and Employee.**

5. Take note on *class* Patient, What Which done by code `consultation.getInfo()` ?

   In the `Patient class` , the code `consult.getInfo()` is used to returns detailed information about the `consultation object` , such as date, doctor, and nurse, in string form.

6. On method `getInfo()` in class Patient, there is line code:

   `if (!consultationhistory.isEmpty())`

   Whether Which done by line the?

   The code line if (!rihiwatKonsultasi.isEmpty()) checks whether the list of sejarahKonsultasi contains data.

   If There Is Data : The code inside the `if block` will be executed , displaying the consultation history information.

   If Empty : The code inside the `else block` will be executed , stating that there is no consultation history yet .

7. In the Patient class constructor, there is a line of code:this.historyConsultation = new ArrayList<>();

   Whether Which done by line the? Whether Which happen If line theremoved?

   The code line `this.rihwatKonsultasi = new ArrayList<>();` in the `Patient class constructor` does :

   Initialize List : Create and assign an empty list for `ConsultationHistory` .

   If removed:

   `UninitializedConsultationHistory` : `ConsultationHistory` will be `null , causing` an error when trying to access it.

IV. Task

Implement studies the case that has made on PBO duties Theory to in program