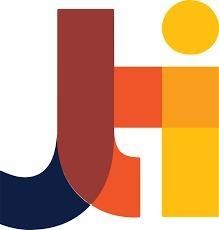
**Laporan Praktikum Pemrograman Berorientasi Objek**

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



Nama : Syaqira Nazaretna

NIM : 2341760123

Prodi : D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

# Jobsheet 04 ‑ Relasi Kelas

## I. Kompetensi

Setelah menempuh pokok bahasan ini, mahasiswa mampu:

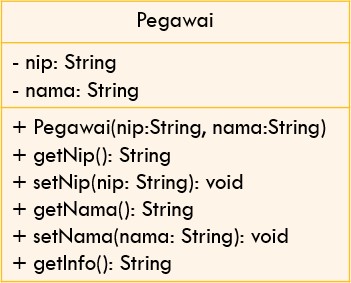
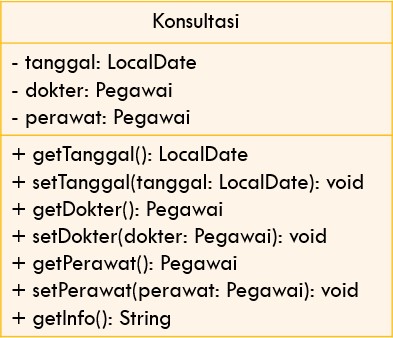
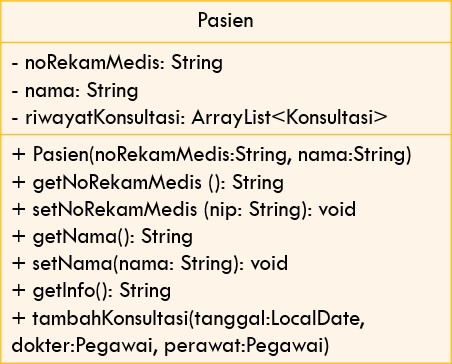
1. Memahami konsep relasi kelas; 2. Mengimplementasikan relasi asosiasi ke dalam program.

## II. Pendahuluan

Pada kasus yang lebih kompleks, dalam suatu sistem akan ditemukan lebih dari satu *class* yang saling memiliki keterkaitan antara *class* satu dengan yang lain. Pada percobaan‑percobaan sebelumnya, mayoritas kasus yang sudah dikerjakan hanya fokus pada satu *class* saja. Pada jobsheet ini akan dilakukan percobaan yang melibatkan beberapa *class* yang saling berkaitan.

## III. Praktikum

Pada praktikum ini akan dikembangkan suatu sistem informasi rumah sakit yang menyimpan data riwayat konsultasi pasien. Perhatikan diagram *class* berikut:



1



0..\*



0..\*



0..\*



merawat



memeriksa



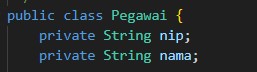
1



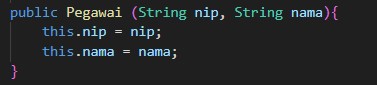
1

1. Buatlah folder baru dengan nama RumahSakit

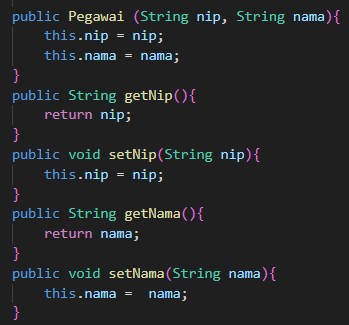
1. Buatlah class Pegawai. Tambahkan atribut nip dan nama pada class Pegawai dengan akses modifier private



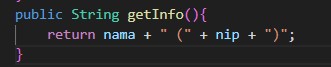
1. Buatlah *constructor* untuk class Pegawai dengan parameter nip dan nama.



1. Implementasikan **setter** dan **getter** untuk class Pegawai.



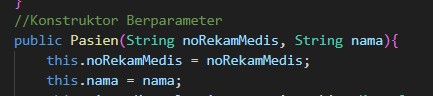
1. Implementasikan *method* getInfo()sebagai berikut:



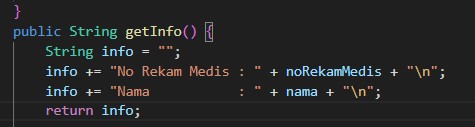
1. Selanjutnya buatlah class Pasien kemudian tambahkan atribut noRekamMedis dan nama pada class Pasien dengan access level modifier private. Sediakan pula setter dan getter untuk kedua atribut tersebut



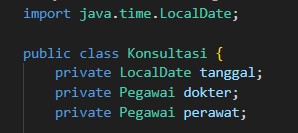
1. Buatlah constructor untuk class Pasien dengan parameter noRekamMedis dan nama



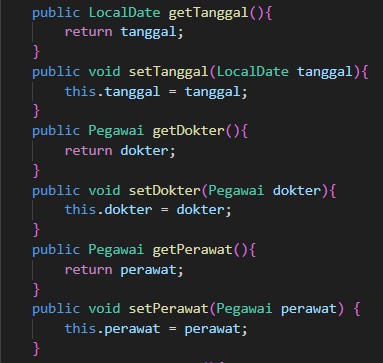
1. Implementasikan *method* getInfo()sebagai berikut:



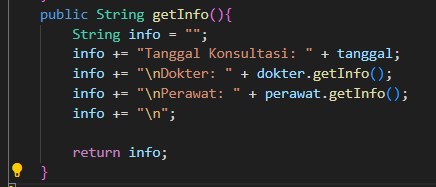
1. Sistem ini akan menyimpan data setiap konsultasi yang dilakukan pasien. Pasien bisa melakukan konsultasi lebih dari sekali. Oleh karena itu, data konsultasi akan disimpan dalam bentuk ArrayList dari objek-objek yang bertipe Konsultasi.
2. Buatlah class dengan nama Konsultasi dengan atribut tanggal 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.



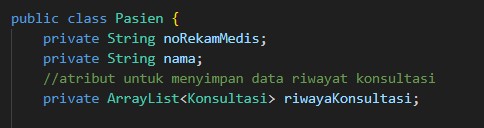
1. Sediakan setter dan getter untuk masing-masing atribut pada class Konsultasi



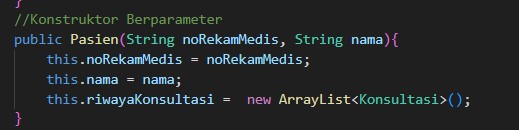
1. Implementasikan method getInfo() sebagai berikut:



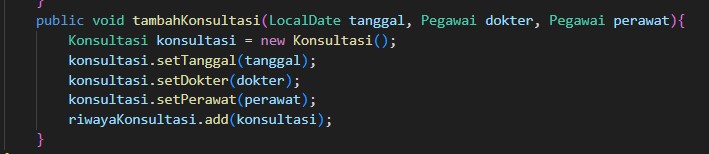
1. Untuk menyimpan data riwayat konsultasi pasien, maka tambahkan atribut riwayatKonsultasi pada class Pasien dengan tipe arrayList<Konsultasi>. Atribut ini akan menyimpan sekumpulan objek bertipe Konsultasi. **Import java.util.ArrayList** agar dapat mendeklarasikan atribut bertipe ArrayList of object.



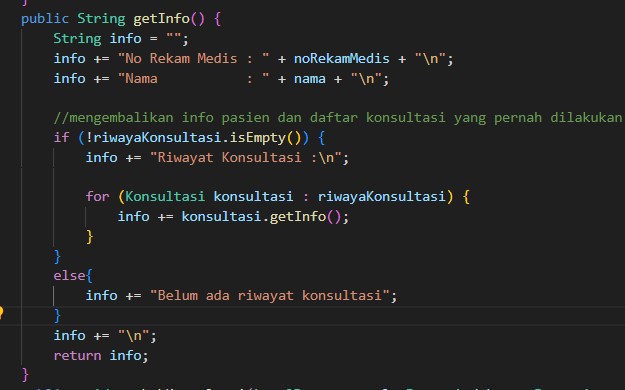
1. Buatlah constructor berparameter untuk class Pasien. Inisialisasi nilai atribut noRekamMedis dan nama berdasarkan parameter nama. Instansiasi/buat ArrayList baru untuk atribut riwayatKonsultasi;



1. Lakukan **import java.time.LocalDate** agar dapat mendeklarasikan atribut tanggal bertipe LocalDate pada class Pasien. Selanjutnya, implementasikan method tambahKonsultasi() sebagai berikut:

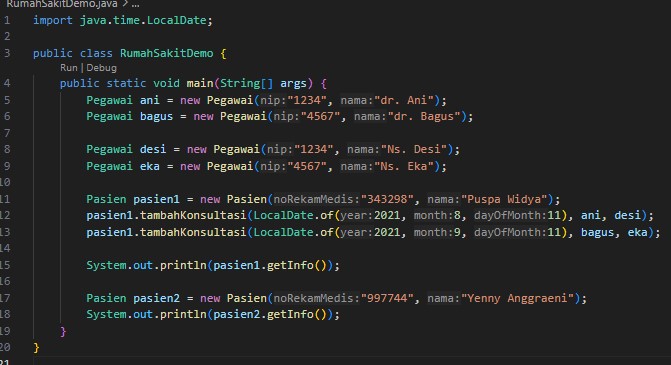


1. Modifikasi method getInfo() untuk mengembalikan info pasien dan daftar konsultasi yang pernah dilakukan

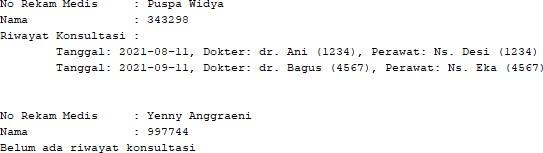


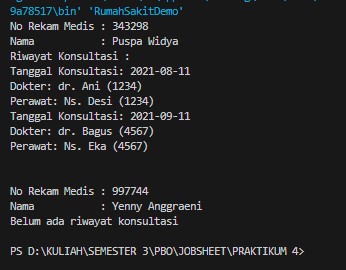
1. Test program yang sudah dibuat dengan membuat objek-objek pada class

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:



1. *Compile* kemudian *run* RumahSakitDemo dan didapatkan hasil seperti berikut:





## 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 : pada class tersebut setter gunanya untuk mengubah atau menetapkan nilai dari atribut privat, sedangkan getter digunakan untuk mengambil nilai dari atribut privat.

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

Jawab : Jika di dalam kelas *Konsultasi* tidak terdapat konstruktor yang didefinisikan secara eksplisit, maka Java secara otomatis akan menyediakan konstruktor default. Konstruktor default ini adalah konstruktor tanpa parameter yang menginisialisasi objek tanpa melakukan tindakan khusus.

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

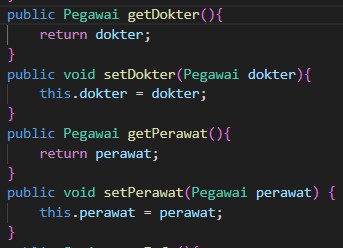
Jawab : dalam class konsultasi atribut yang bertipe object adalah tanggal, dokter, perawat

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

Jawab : pada atribute dan method pada kode ini Atribut



Method



1. Perhatikan pada *class* Pasien, apa yang dilakukan oleh kode konsultasi.getInfo()? Jawab : akan melakukan pemanggilan method konsultasi.getInfo() yang dilakukan adalah mengakses informasi yang ada dalam method tersebut
2. Pada method getInfo() dalam class Pasien, terdapat baris kode:

if (!riwayatKonsultasi.isEmpty())

Apakah yang dilakukan oleh baris tersebut?

Jawab : mengecek apakah riwayat konsultasi kosong atau tidak, jika tidak maka akan mencetak riwayat konsultasi pasien. Jika iya maka akan mencetak info “Belum ada riwayat konsultasi”

1. 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 : melakukan inisialisasi atribut riwayatKonsultasi, yang kemungkinan bertipe ArrayList, dengan objek ArrayList baru. Artinya, saat objek Pasien dibuat, atribut riwayatKonsultasi akan langsung memiliki nilai berupa daftar kosong yang siap digunakan untuk menampung riwayat konsultasi.

## IV. Tugas

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









Outputnya

