

SOAL RPL-BO JAVA I/O DAN SERIALISASI

Waktu Pengerjaan: **2 jam 30 menit**

Petunjuk Pengumpulan Jawaban :

Buatlah folder bernama UG14_NIM sebagai tempat untuk menyimpan jawaban kamu.

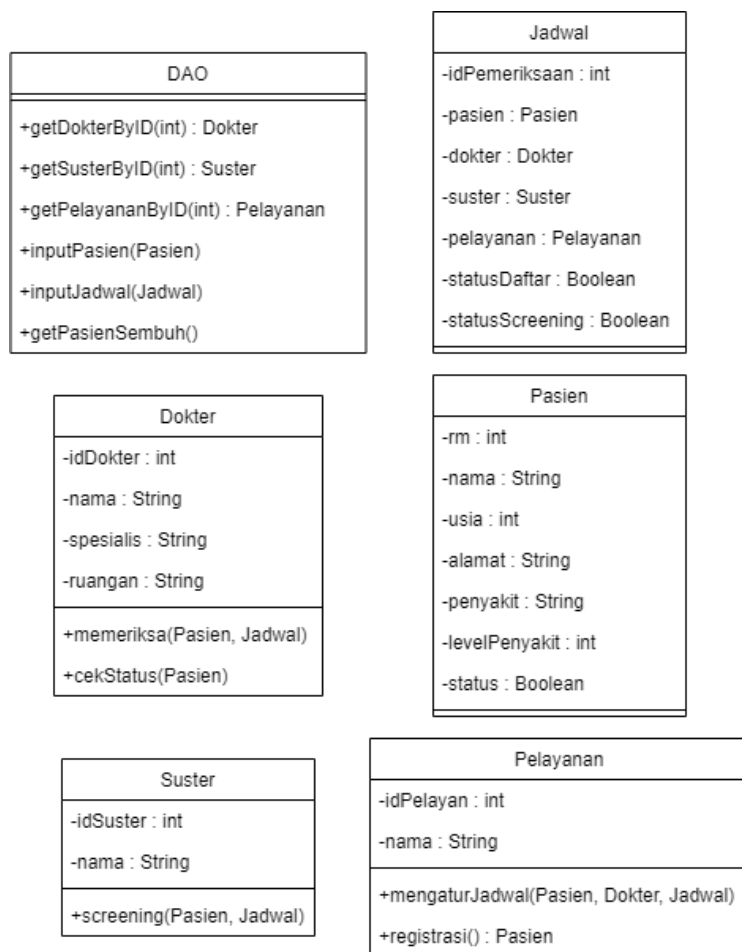
Selanjutnya, buatlah project sesuai dengan ketentuan soal. Simpan project tersebut ke dalam folder yang baru saja kamu buat. Upload/push folder UG14_NIM berisi project jawaban kamu ke repository UG_RPLBO_NIM_GRUP yang sudah ada di Github kamu.

Soal (100 Poin)

Buatlah sebuah project Maven dengan ketentuan:

- GroupId: com.ug14.rumahsakit
- Nama Project/ArtifactId: UG14-NIM-SakitRumah --> Pastikan ini menjadi folder yang berisi jawaban project kamu!
- Archetype: maven-archetype-quickstart

Anda merupakan seorang backend developer rumah sakit yang bertugas untuk membuat sistem alur kedatangan pasien. Berikut adalah task yang harus anda kerjakan.



Class Pasien :

Attribut	Penjelasan
rm	Digunakan sebagai penanda unik dari pasien
nama	Digunakan untuk menyimpan nama pasien
usia	Digunakan untuk menyimpan usia pasien
alamat	Digunakan untuk menyimpan alamat pasien
penyakit	Digunakan untuk menyimpan penyakit pasien
levelPenyakit	Digunakan untuk mendefinisikan masih berapa kali harus diperiksa dokter agar bisa sembuh (Ditentukan sendiri saja)
status	Digunakan untuk menyimpan status kesehatan pasien (sembuh atau belum sembuh) DEFAULT : False (sakit)

Class Dokter :

Attribut	Penjelasan
idDokter	Digunakan sebagai penanda unik dari dokter
nama	Digunakan untuk menyimpan nama dokter
spesialis	Digunakan untuk menyimpan spesialisasi Dokter
ruangan	Digunakan untuk menyimpan alamat pasien

Method	Penjelasan
memeriksa(Pasien, Jadwal)	Method ini berfungsi untuk memeriksa SEKALIGUS menyembuhkan pasien. Nantinya setiap kali method ini dijalankan, maka levelPenyakit akan terus berkurang 1 poin. Dan jika sudah 0, maka status pasien akan sembuh. (Harus screening dulu di suster baru bisa diperiksa dan yang memeriksa harus dokter yang terjadwal)
cekStatus(Pasien)	Method ini digunakan untuk mengecek status pasien apakah sudah sembuh atau belum

Class Suster :

Attribut	Penjelasan
idSuster	Digunakan sebagai penanda unik dari suster
nama	Digunakan untuk menyimpan nama suster
Method	Penjelasan
screening(Pasien, Jadwal)	Method ini berfungsi untuk melakukan screening dulu terhadap pasien sebagai persyaratan untuk diperiksa oleh dokter (Harus melalui proses pelayanan dulu). (Apabila yang melakukan screening bukan suster terdaftar maka akan error)

Class Pelayanan :

Attribut	Penjelasan
idPelayanan	Digunakan sebagai penanda unik pelayanan
nama	Digunakan untuk menyimpan nama pelayan
Method	Penjelasan
mengaturJadwal(Pasien, Dokter, Jadwal)	Method ini untuk mengatur jadwal dokteran antara Pasien dan Dokter terkait sekaligus entry jadwal pemeriksaan ke database. (Pasien harus sakit dulu)
registrasi()	Method ini berfungsi untuk melakukan registrasi Pasien dan menginisialisasi object pasien sekaligus melakukan entry data pasien ke database melalui class DAO.

Class Jadwal :

Attribut	Penjelasan
idPemeriksaan	Digunakan sebagai penanda unik jadwal pemeriksaan
pasien	Digunakan untuk menyimpan data pasien
dokter	Digunakan untuk menyimpan data dokter

suster	Digunakan untuk menyimpan data suster
pelayanan	Digunakan untuk menyimpan data pelayanan
statusDaftar	Digunakan untuk menyimpan statusDaftar untuk lanjut ke tahap berikutnya (DEFAULT = false)
statusScreening	Digunakan untuk menyimpan statusScreening untuk lanjut ke tahap berikutnya (DEFAULT = false)

Class DAO:

Method	Penjelasan
getDokterByID	Method ini berfungsi untuk mendapatkan data dokter dari database dengan parameter idDokter
getSusterByID	Method ini berfungsi untuk mendapatkan data suster dari database dengan parameter idSuster
getPelayananByID	Method ini berfungsi untuk mendapatkan data pelayanan dari database dengan parameter idPelayan
inputPasien	Method ini berfungsi untuk menambahkan data pasien ke database pada table pasien
inputJadwal	Method ini berfungsi untuk menambahkan data jadwal ke database pada table jadwal
getPasienSembuh	Method ini berfungsi untuk mendapatkan daftar pasien yang sudah sembuh atau belum berdasarkan jadwal yang sudah dibuat dalam bentuk daftar nama di output console (Di diagram tidak ada karena storage drive saya penuh :D)
updateStatusPasien	Method ini berfungsi untuk mengubah status pasien yang semula sakit menjadi sembuh (Di database tipe datanya int berarti antara 0 dan 1. 0 = sakit, 1 = sembuh)

Alur :

Alur dari programnya adalah pasien harus melakukan proses pelayanan di pelayanan terlebih dahulu. Nanti akan ditanyakan nama, usia, alamat, dan penyakit. Untuk levelnya nanti ditentukan manualan (disini baru pasien dideklarasikan setelah diinputkan). Setelah itu, pasien akan menuju proses screening oleh suster. Setelah proses screening oleh suster, barulah pasien boleh diperiksa oleh dokter .

Contoh App.java dan Output yang diharapkan :

```
package com.ug14.rumahsakit;

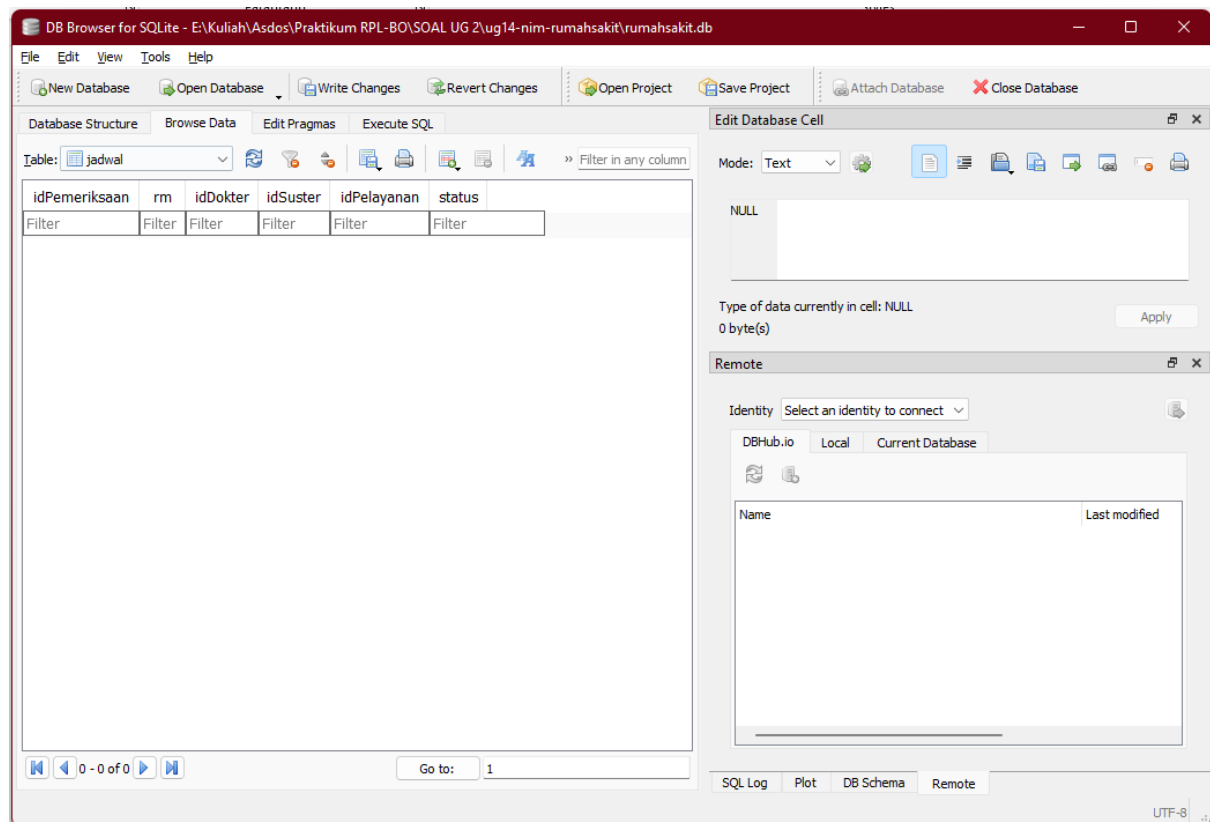
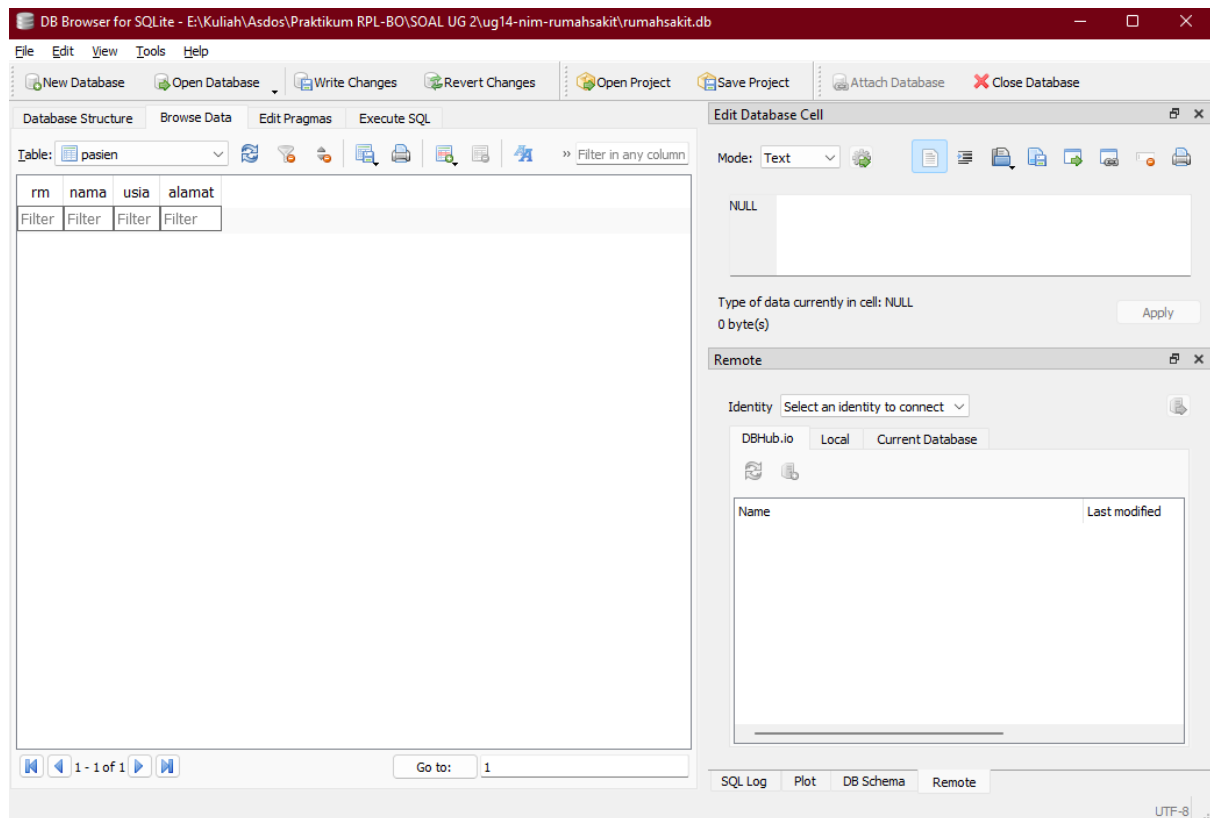
public class App {
    public static void main(String[] args) {
        System.out.println("=====PROSES
INISIALISASI=====");
        System.out.println();
        Jadwal jadwal = new Jadwal();
        Pasien illegal = new Pasien("test",21,"di mana saja");
        Pelayanan pelayanan = new Pelayanan("Andi");
        Suster suster = new Suster("Siti");
```

```

        Dokter dokter = new Dokter("Budi", "Organ Dalam", "A123");
        Pasien pasien = pelayanan.registrasi();
        pelayanan.mengaturJadwal(pasien, dokter, suster, jadwal);
        System.out.println("=====PROSES UJI
COBA PASIEN DAN JADWAL ILLEGAL=====");
        System.out.println();
        suster.screening(illegal, jadwal);
        dokter.memeriksa(illegal, jadwal);
        System.out.println("=====PROSES UJI
COBA PASIEN DAN JADWAL ILLEGAL
BERHASIL=====");
        System.out.println();
        System.out.println("=====PROSES
SCREENING SUSTER=====");
        System.out.println();
        suster.screening(pasien, jadwal);
        System.out.println("=====PROSES
SCREENING SUSTER BERHASIL=====");
        System.out.println();
        System.out.println("=====PROSES
PEMERIKSAAN DOKTER=====");
        System.out.println();
        int x = pasien.getLevelPenyakit();
        for(int i = 0; i <= x; i++) {
            dokter.memeriksa(pasien, jadwal);
            dokter.cekStatus(pasien, jadwal);
        }
        System.out.println("=====PROSES
PEMERIKSAAN DOKTER BERHASIL=====");
        System.out.println();
        DAO.getPasienSembuh();
    }
}

```

Isi Database sebelum Program Dijalankan :



Output Console :

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS E:\Kuliah\Asdos\Praktikum RPL-BO\SOAL UG 2\ug14-nim-rumahsakit> & 'C:\Program Files\Java\jdk-17.0.2\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '@C:\Users\salah\AppData\Local\Temp\cp_8bxxuswjsao4xj591qh8wy7kz_argfile' 'com.ug14_rumahsakit.App'
-----PROSES INISIALISASI-----

Masukkan rm baru : 12345
Masukkan nama anda : Gideon
Masukkan usia anda : 21
Masukkan alamat anda : Wonogiri
Masukkan penyakit anda : Pilek

Data Pasien Berhasil Disimpan!
-----Proses Registrasi Berhasil-----
Data Jadwal Berhasil Disimpan!
-----Proses Pengaturan Jadwal Berhasil-----
-----PROSES UJI COBA PASIEN DAN JADWAL ILLEGAL-----
-----ANDA HARUS MELAKUKAN PROSES PENDAFTARAN TERLEBIH DAHULU DI BAGIAN PELAYANAN-----
-----ANDA HARUS MELAKUKAN PROSES PENDAFTARAN TERLEBIH DAHULU DI BAGIAN PELAYANAN-----
-----PROSES UJI COBA PASIEN DAN JADWAL ILLEGAL BERHASIL-----
-----PROSES SCREENING SUSTER-----
-----PROSES SCREENING SUTER BERHASIL-----
-----PROSES PEMERIKSAAN DOKTER-----
-----PASIEN ANDA MASIH SAKIT-----
-----PASIEN ANDA MASIH SAKIT-----
-----PASIEN ANDA MASIH SAKIT-----
-----SELAMAT PASIEN ANDA SUDAH SEMBUH DAN SEHAT-----
Status Pasien berhasil diubah!
-----PROSES PEMERIKSAAN DOKTER BERHASIL-----

12345
PS E:\Kuliah\Asdos\Praktikum RPL-BO\SOAL UG 2\ug14-nim-rumahsakit>

```

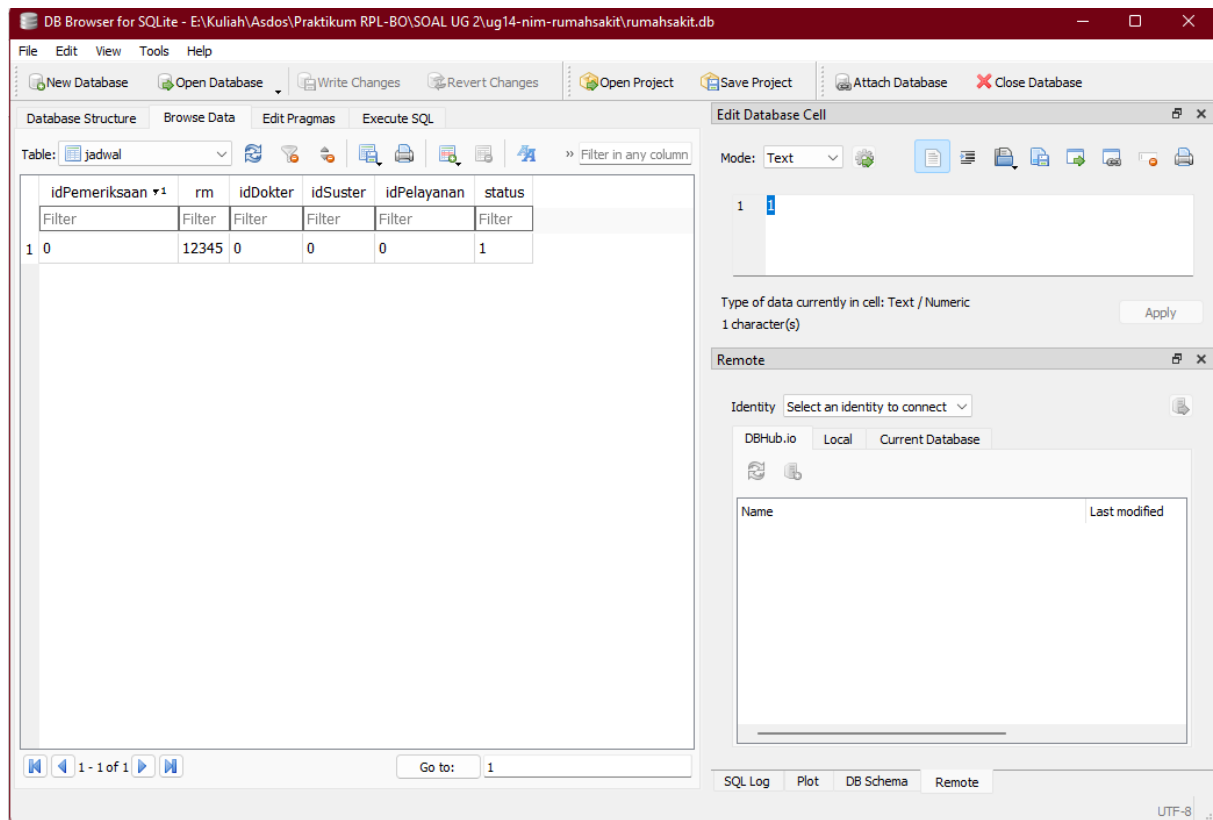
Isi Database Pasien :

The screenshot shows the DB Browser for SQLite application. The main window displays the 'pasien' table with the following data:

rm	nama	usia	alamat
12345	Gideon	21	Wonogiri

The interface includes a menu bar (File, Edit, View, Tools, Help), a toolbar with various database actions, and a sidebar with tabs for Database Structure, Browse Data, Edit Pragma, and Execute SQL. The 'Browse Data' tab is active, showing the table data. The 'Edit Database Cell' panel on the right allows for editing the selected cell. The 'Remote' panel at the bottom right shows connection options for DBHub.io, Local, and Current Database.

Isi Database Jadwal :



Poin Penilaian (100) :

- Alur program berjalan sebagaimana mestinya (10)
- Semua method berjalan sesuai ketentuan (40)
- Semua class dideklarasikan sesuai ketentuan (40)
- Output sesuai (10)