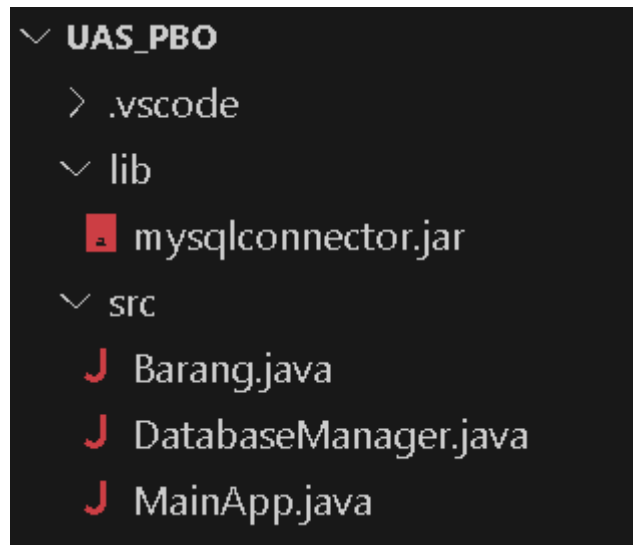


Nama : Regal Nugraha  
NIM : 20230040175  
Kelas : TI23C

**Jelaskan struktur foldernya sehingga MySQL Connector bisa digunakan didalam program**

Struktur Folder Proyek Java dan Penggunaan MySQL Connector



➤ **Penjelasan Setiap Bagian:**

1. UAS\_PBO/ (Folder Utama Proyek):

- Ini adalah folder teratas yang menampung seluruh aplikasi Java . Semua yang berkaitan dengan proyek ini, termasuk kode program, library tambahan, dan konfigurasi lainnya, akan berada di dalam folder ini. Ini berfungsi sebagai titik awal untuk Java dalam memahami struktur proyek.

2. src/ (Source Folder):

- Singkatan dari "source", folder ini secara standar digunakan untuk menyimpan semua file kode sumber Java (.java). Ini adalah tempat menulis logika program, seperti kelas Barang, DatabaseManager, dan MainApp.
- Saat program Java dikompilasi, file .java di sini akan diubah menjadi file .class (bytecode) yang dapat dieksekusi oleh Java Virtual Machine (JVM).

3. lib/ (Libraries Folder):

- Singkatan dari "libraries", folder ini adalah tempat yang direkomendasikan untuk meletakkan semua library atau pustaka pihak ketiga yang tidak termasuk dalam Java Development Kit (JDK) standar, tetapi dibutuhkan oleh program.
- mysqlconnector.jar library yang digunakan pada proyek ini. File JAR ini berisi semua kelas dan fungsi yang diperlukan agar program Java dapat "berkomunikasi" dengan database MySQL (yaitu, membuka koneksi, mengirim perintah SQL, menerima hasil, dll.). Tanpa library ini, Java tidak akan tahu bagaimana caranya berbicara dengan MySQL.

### ➤ Bagaimana MySQL Connector Digunakan dalam Program

Dalam proyek ini, mysql-connector-java.jar berperan sebagai penghubung antara program Java dan database MySQL. Karena sudah dikonfigurasi melalui fitur Java: Configure Classpath di VS Code, library ini menyediakan semua kelas JDBC yang dibutuhkan untuk menjalankan koneksi database, memanggil *stored procedure*, serta menjalankan *trigger* secara otomatis.

#### 1. Membuka Koneksi ke Database

Kelas DatabaseManager bertanggung jawab menginisialisasi koneksi ke database MySQL:

```
connection = DriverManager.getConnection(JDBC_URL, USER, PASSWORD);
```

Perintah ini hanya berhasil karena mysqlconnector.jar menyediakan driver com.mysql.cj.jdbc.Driver yang mengenali format URL JDBC MySQL (jdbc:mysql://...).

#### 2. Menjalankan Stored Procedure

Aplikasi menggunakan stored procedure bernama InsertBarang untuk menambahkan data ke tabel barang. Pemanggilan ini dilakukan menggunakan CallableStatement:

```
String callSQL = "{CALL InsertBarang(?, ?, ?, ?)}";
```

```
callableStatement = connection.prepareCall(callSQL);
```

Library mysqlconnector.jar memungkinkan pemanggilan stored procedure tersebut dari Java agar dapat dijalankan di sisi MySQL.

#### 3. Pemicu (Trigger) Otomatis Dijalankan dari MySQL

Setiap kali prosedur InsertBarang dijalankan, trigger AfterInsertBarang di sisi server MySQL akan aktif secara otomatis. Trigger ini bertugas mencatat log penyisipan data barang ke dalam tabel log\_insert\_barang.

Struktur tabel log tersebut adalah:

- id\_log (Primary Key, auto-increment)
- kode (Kode barang yang ditambahkan)

- waktu (Timestamp saat data dimasukkan)

Karena trigger dijalankan di sisi MySQL, aplikasi Java tidak perlu memanggil trigger secara eksplisit. Prosesnya berjalan otomatis setelah prosedur InsertBarang selesai dijalankan.

Contoh alur:

- Pengguna memasukkan data barang via Java (MainApp.java)
- Java memanggil InsertBarang via CallableStatement
- InsertBarang menambahkan data ke tabel barang
- Trigger AfterInsertBarang langsung aktif, dan mencatat data ke tabel log\_insert\_barang

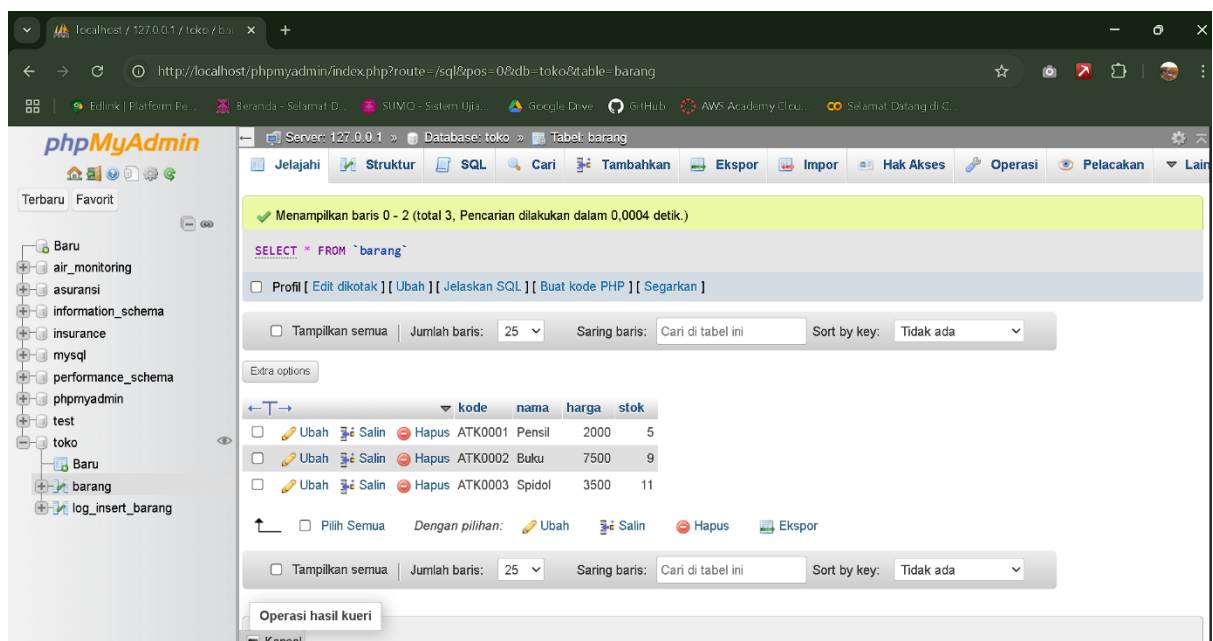
#### 4. Menampilkan Data Barang

Data barang ditampilkan menggunakan perintah SQL SELECT, melalui PreparedStatement:

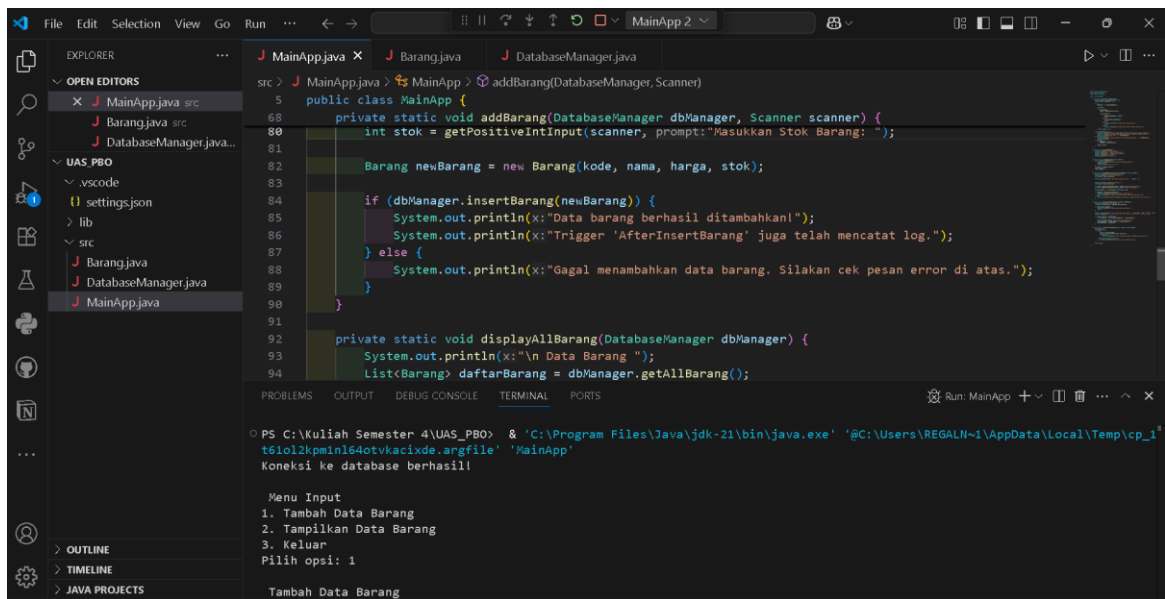
*SELECT kode, nama, harga, stok FROM barang*

Hasilnya diproses menggunakan ResultSet, dan ditampilkan dalam format tabel di konsol.

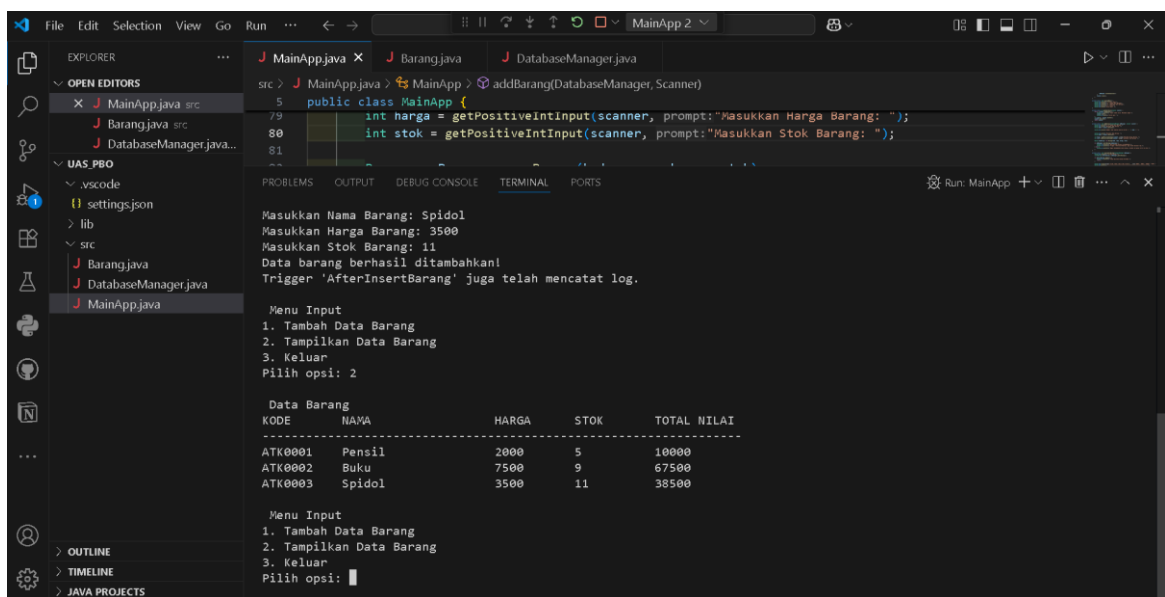
Konfigurasi mysqlconnector.jar sebagai library melalui Java: Configure Classpath memastikan bahwa semua fitur JDBC mulai dari pemanggilan prosedur tersimpan hingga aktivasi trigger otomatis dapat digunakan tanpa perlu penambahan library secara manual di runtime. Seluruh eksekusi SQL berjalan stabil karena dependensi sudah dikenali oleh proyek Java.



Gambar Screenshoot Database, data berhasil diinsert



Gambar Screenshoot Program Insert



Gambar Screenshoot Input Data dan Tampilkan Data