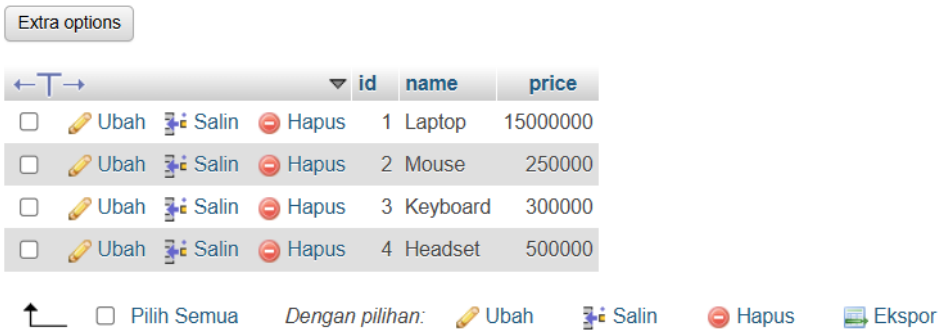
Nama : Pradana Argo Pangestu

Nim : 2311104079

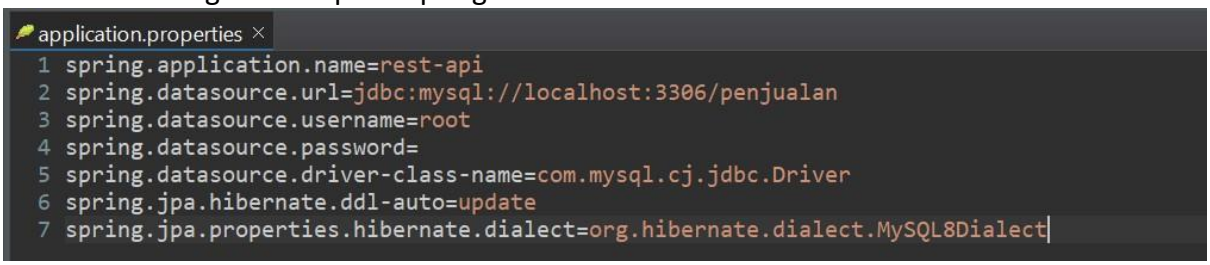
MODUL 10

1. Membuat Database dengan nama Penjualan
   a. Buat tabel dengan nama Product dengan struktur berikut

   b. Isi Tabel Product

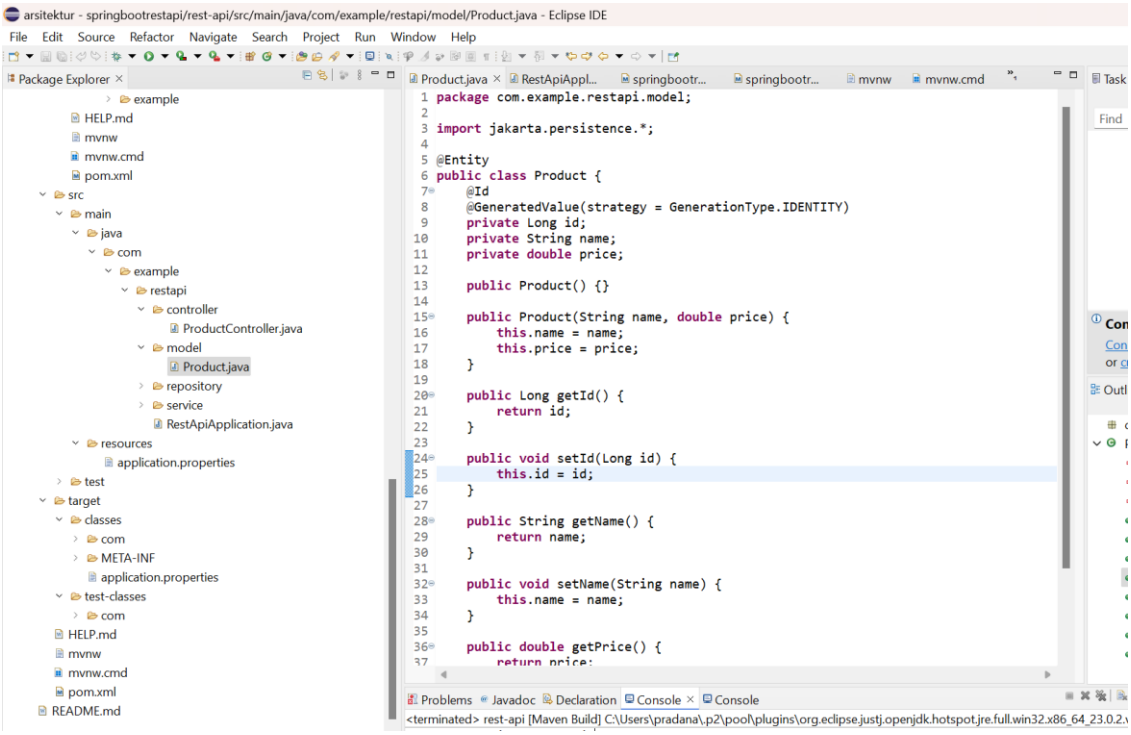


2. Membuat Konfigurasi DB pada Spring Boot



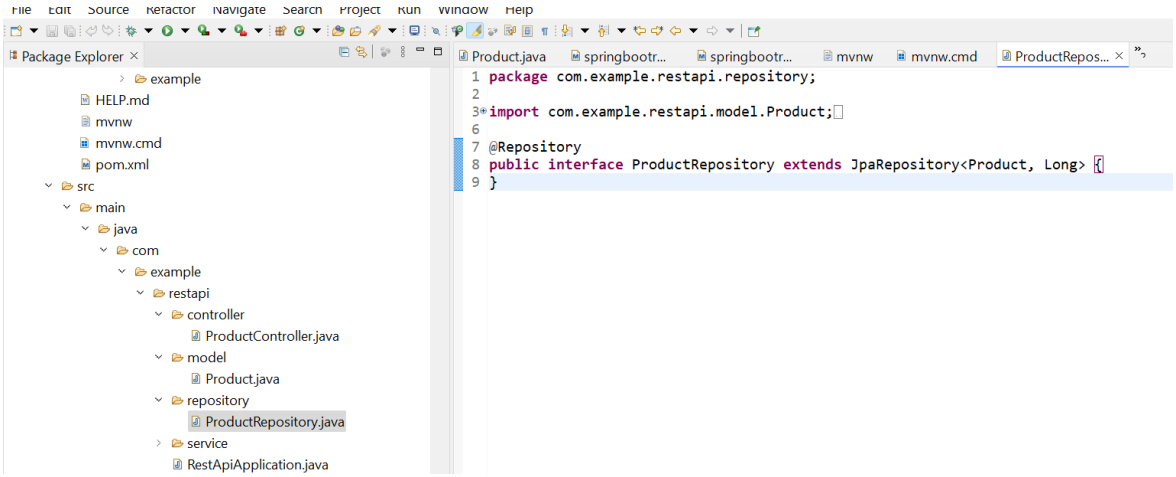
Gambar berikut merupakan pengaturan dari aplikasi Spring Boot dengan nama "rest-api", dengan koneksi ke database MySQL dengan nama"penjualan" yang berjalan di localhost pada port 3306. Aplikasi menggunakan username "root" tanpa password untuk mengakses database, serta memanfaatkan driver JDBC MySQL (com.mysql.cj.jdbc.Driver).Hibernate dikonfigurasi untuk secara otomatis memperbarui skema database sesuai dengan entitas JPA melalui opsi ddl-auto=update. Selain itu, digunakan dialek Hibernate MySQL8Dialect agar query yang dihasilkan sesuai dengan sintaks MySQL versi 8.

3. Membuat Model Entity



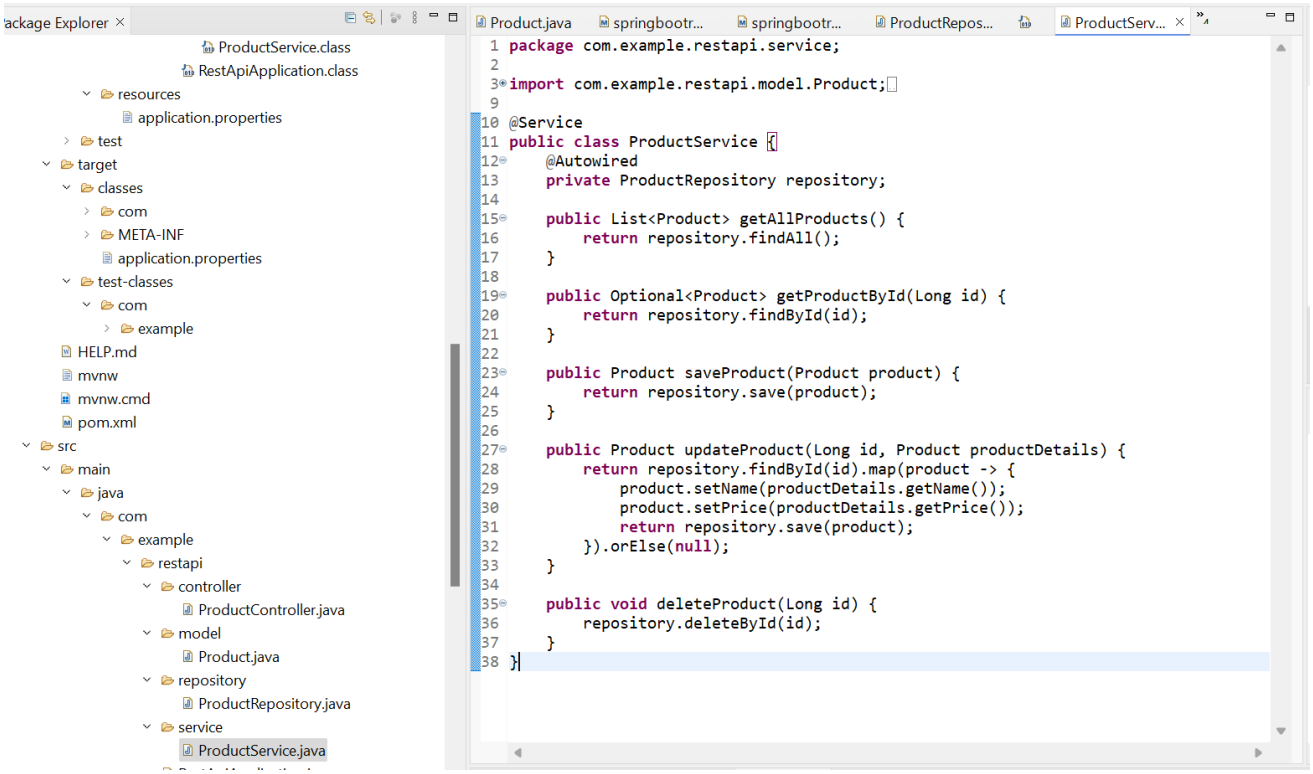
Gambar berikut nama class Product yang merepresentasikan entitas produk dalam aplikasi. Class ini akan disimpan sebagai tabel di database karena diberi anotasi @Entity. Setiap produk memiliki tiga atribut utama: id sebagai identitas unik (yang akan dibuat otomatis oleh database), name sebagai nama produk, dan price sebagai harga produk. Class ini juga menyediakan konstruktor default, konstruktor dengan parameter, serta getter dan setter untuk mengakses dan mengubah nilai dari masing-masing atribut.

4. Membuat Repository



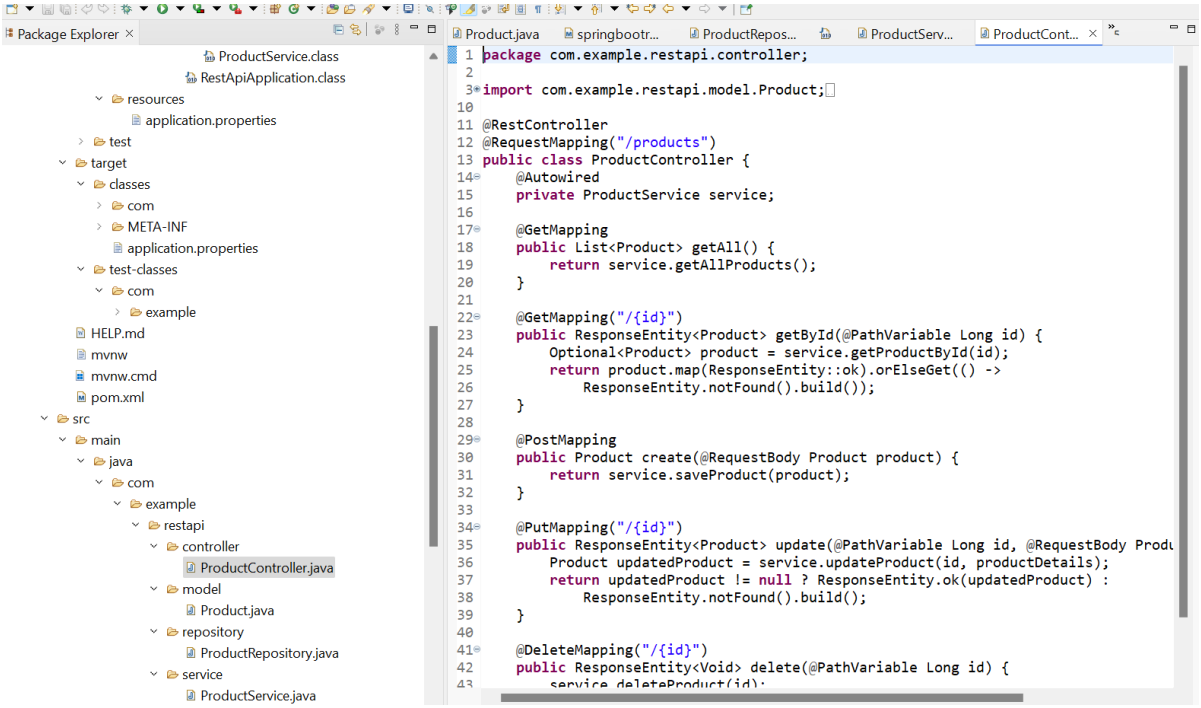
Gambar berikut merupakan interface ProductRepository yang berfungsi sebagai penghubung antara aplikasi dan database untuk entitas Product. Dengan mewarisi JpaRepository, interface ini secara otomatis memiliki fitur bawaan seperti menyimpan, mencari, menghapus, dan memperbarui data produk tanpa perlu menulis kode SQL secara manual. Anotasi @Repository menunjukkan bahwa ini adalah komponen yang berperan dalam pengelolaan data di Spring.

5. Membuat Service Layar



Gambar berikut merupakan class ProductService yang berfungsi sebagai lapisan layanan (service layer) dalam aplikasi untuk mengatur logika bisnis terkait data Product. Class ini menggunakan ProductRepository untuk berinteraksi dengan database. Dengan bantuan anotasi @Service, Spring mengenali class ini sebagai komponen service. Di dalamnya terdapat beberapa method utama, seperti mengambil semua produk (getAllProducts), mengambil produk berdasarkan ID (getProductById), menyimpan produk baru (saveProduct), memperbarui data produk (updateProduct), dan menghapus produk berdasarkan ID (deleteProduct). Semua operasi ini menggunakan method bawaan dari JpaRepository, sehingga memudahkan pengelolaan data tanpa perlu menulis query secara manual.

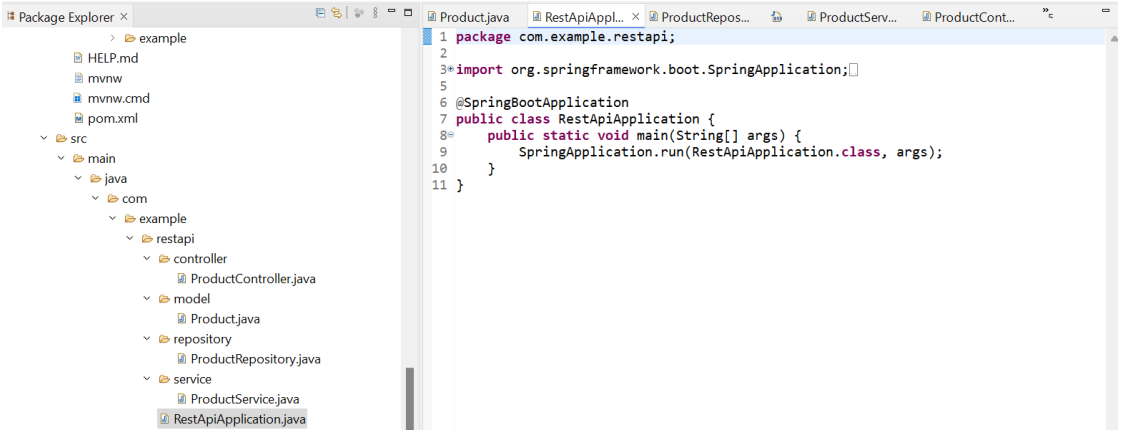
6. Membuat Controller



gambar berikut merupakan class ProductController yang berfungsi sebagai pengatur jalur (controller) dalam aplikasi REST API untuk menangani permintaan HTTP terkait

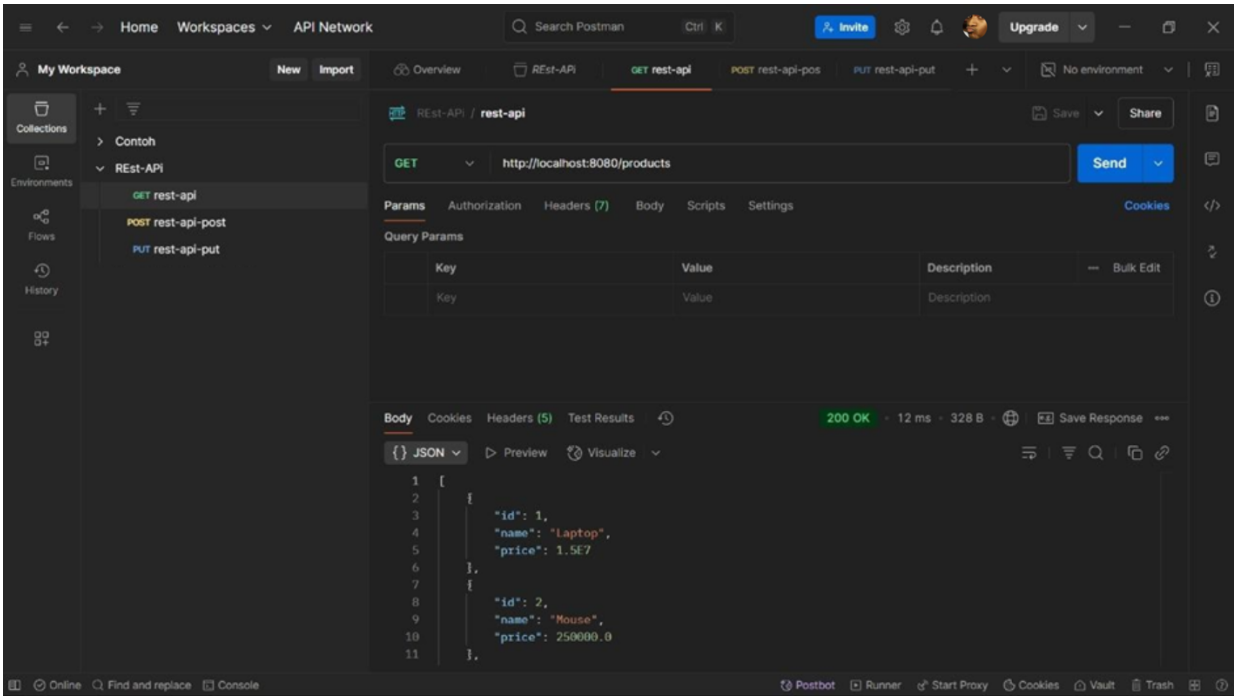
data produk. Dengan anotasi @RestController dan @RequestMapping("/products"), class ini menangani semua permintaan ke endpoint /products. Melalui metode-metode seperti getAll, getById, create, update, dan delete, controller ini memfasilitasi operasi CRUD (Create, Read, Update, Delete) pada data produk. Setiap permintaan akan diteruskan ke ProductService untuk diproses, lalu mengembalikan respons yang sesuai kepada client, baik itu berupa data produk, pesan sukses, atau status "tidak ditemukan".

7. Menjalankan Spring Boot

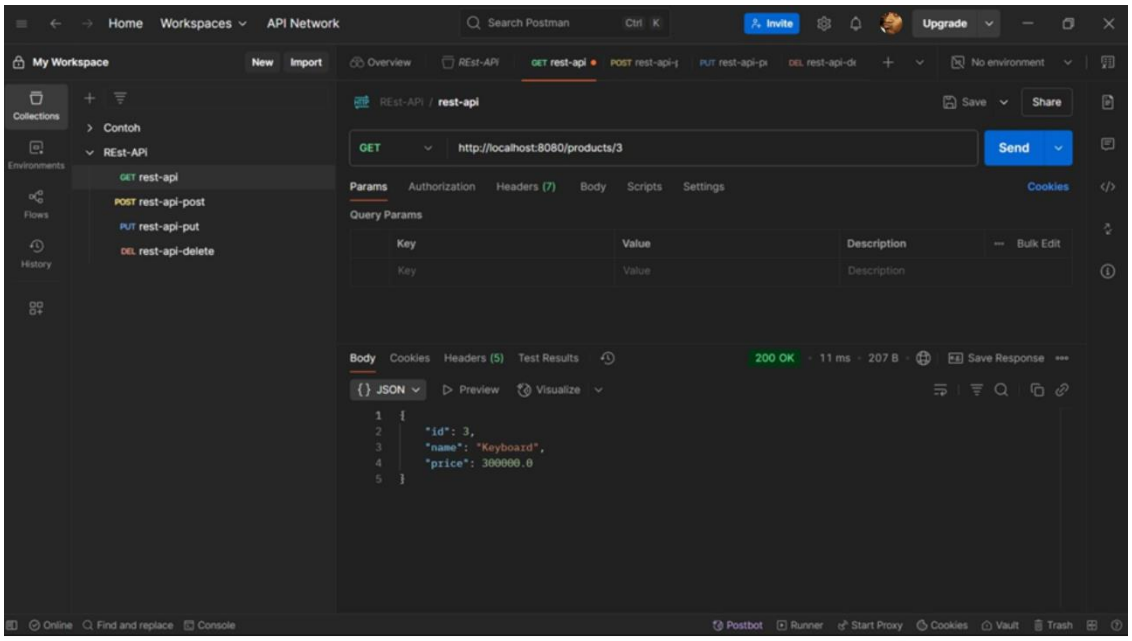


Gambar berikut merupakan class utama RestApiApplication yang menjadi titik awal (entry point) untuk menjalankan aplikasi Spring Boot. Dengan anotasi @SpringBootApplication, Spring akan secara otomatis mengkonfigurasi dan memulai seluruh komponen aplikasi, termasuk pemindaian komponen (component scan), konfigurasi otomatis, dan inisialisasi konteks aplikasi. Method main memanggil SpringApplication.run, yang akan menjalankan aplikasi dan mengaktifkan server web agar siap menerima permintaan (request) dari client.

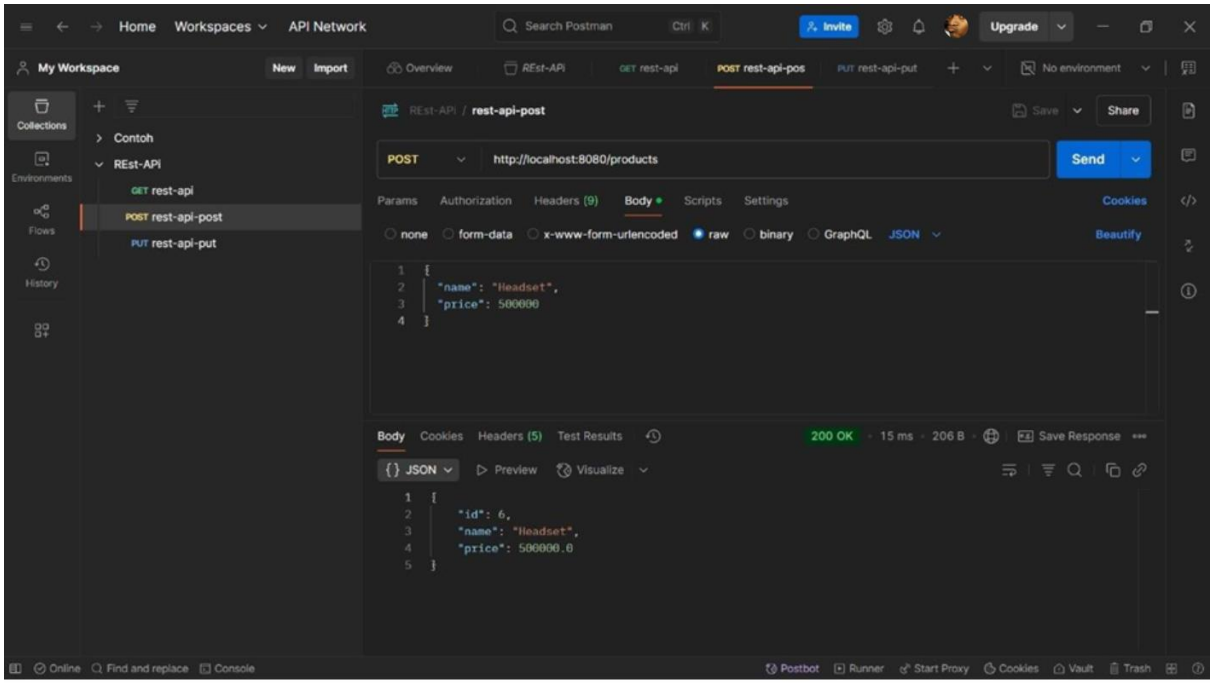
8. GET – Mendapatkan semua product



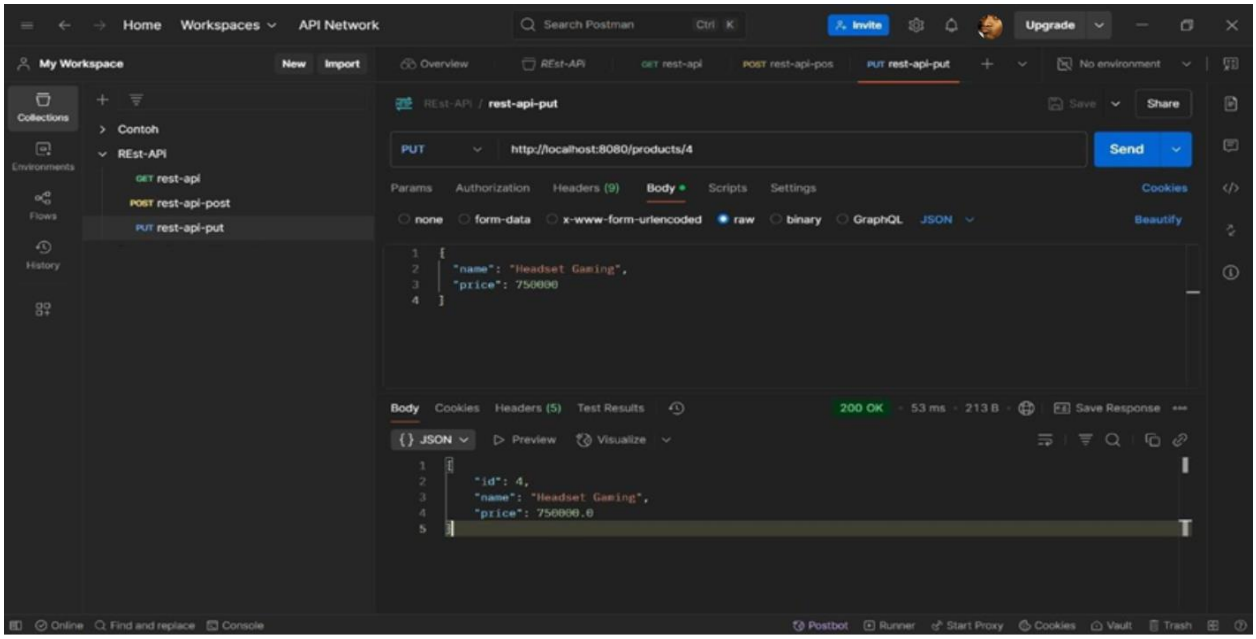
9. GET- Mendapatkan product berdasarkan Id

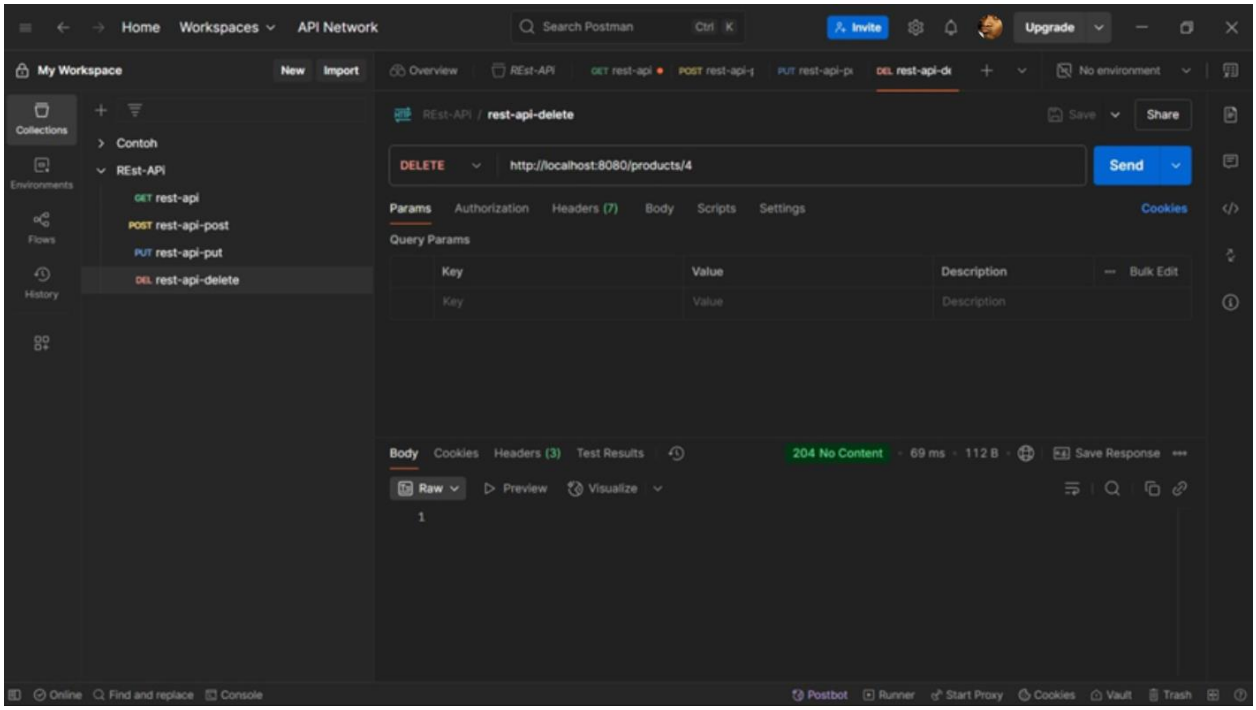


10. POST – menambahkan product baru yang dinamai Headset

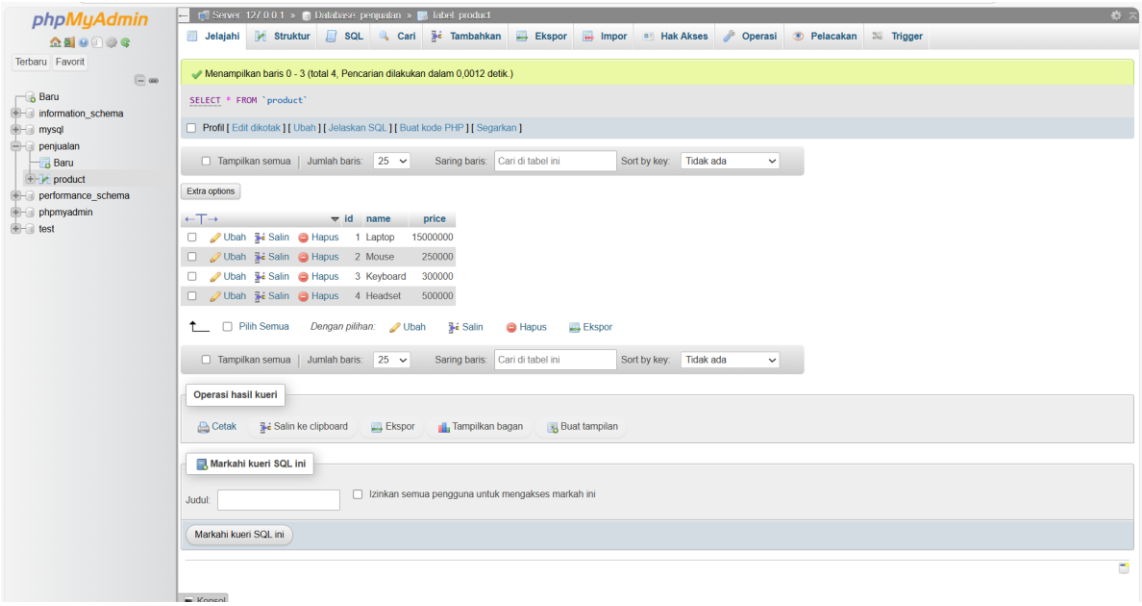


11. PUT – memperbarui nama dan juga harga pada tabel product dengan id 4





12. DEL – Menghapus isi product yang idnya 4



Kesimpulannya adalah untuk mengetahui bagaimana cara mengimplementasikan rest-api menggunakan spring pada eclipse, dan juga mengetahui cara penggunaan mysql dengan eclipse. juga implementasi apa saja yang digunakan untuk menyambungkan dan membuat api.