**LAPORAN UAS**

**(PENGEMBANGAN BACKEND)**



**Disusun Oleh**

**Abdul Hikam Aprilio/3124101303**

**Novan Ramadani/3124101292**

**Program Studi D3 Manajamen Informatika**

**STIKOM PGRI Banyuwangi Tahun**

**2025**

1. **Pendahuluan**

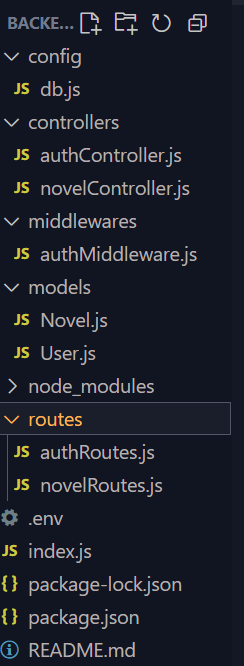
Sistem informasi manajemen novel berbasis web ini dibuat untuk memudahkan pengelolaan data novel secara digital. Pengguna dapat menambahkan, mengedit, menghapus, dan melihat data novel dengan antarmuka berbasis web. Sistem ini memiliki dua peran utama yaitu admin dan user biasa, dengan pembagian hak akses yang berbeda.

1. **Tools dan Teknologi**

Berikut adalah tools dan teknologi yang digunakan dalam pengembangan backend:

|  |  |
| --- | --- |
| **Teknologi** | **Deskripsi** |
| Node.js | Platform runtime JavaScript |
| Express | Framework web untuk Node.js |
| Sequelize | |  | | --- | |  |   ORM untuk Node.js dan database MySQL |
| MySQL | Database relasional |
| JWT | Autentikasi menggunakan JSON Web Token |
| Bcryptjs | Untuk hashing password |
| Dotenv | Mengelola konfigurasi environment |

1. **Struktur Folder (Backend)**

****

1. **Desain Database**

**Tabel User:**

|  |  |  |
| --- | --- | --- |
| **kolom** | **Type Data** | **Keterangan** |
| Id | INTEGER(PK) | Primary Key, auto increment |
| Name | STRING(100) | Nama Pengguna |
| Email | STRING(100) | Unik, wajib |
| Password | STRING(255) | Password terenkripsi |
| role | ENUM | admin atau user |

**Tabel Novel:**

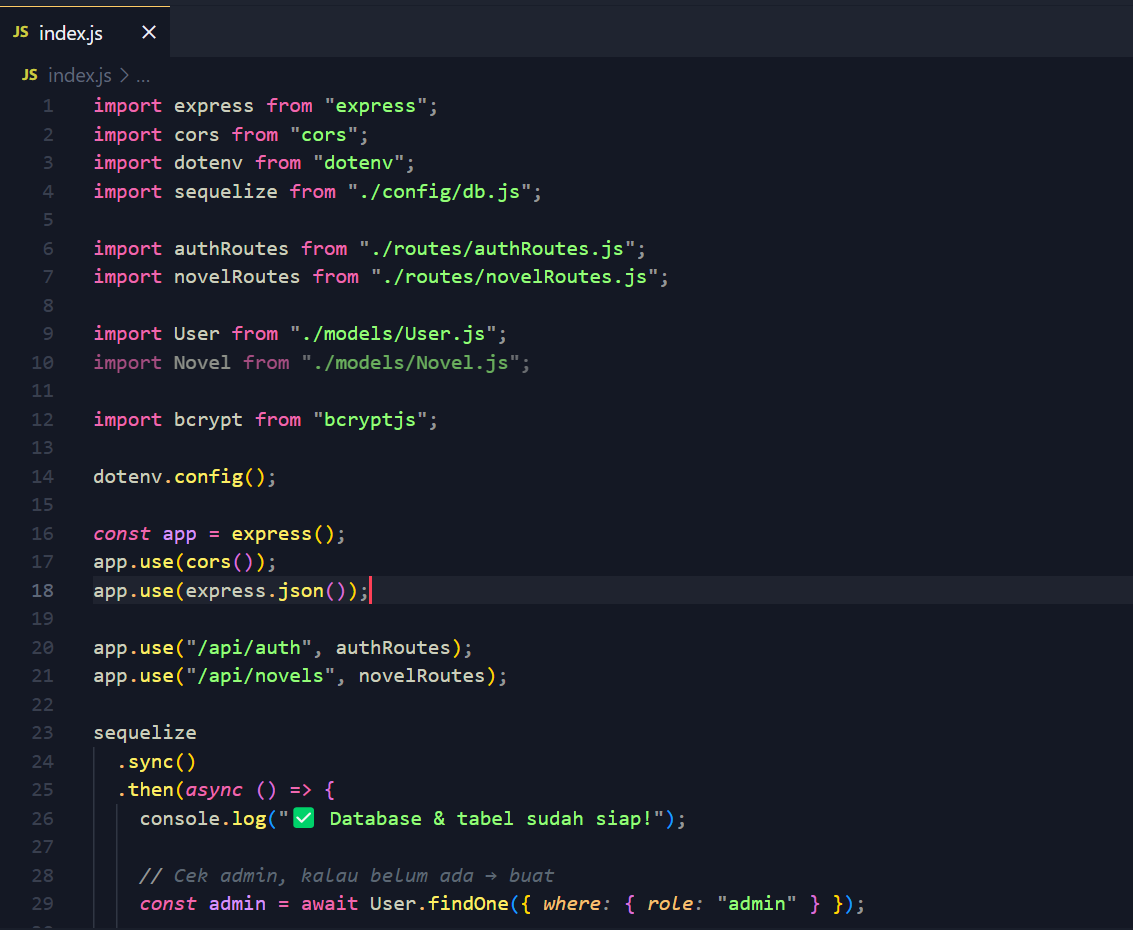
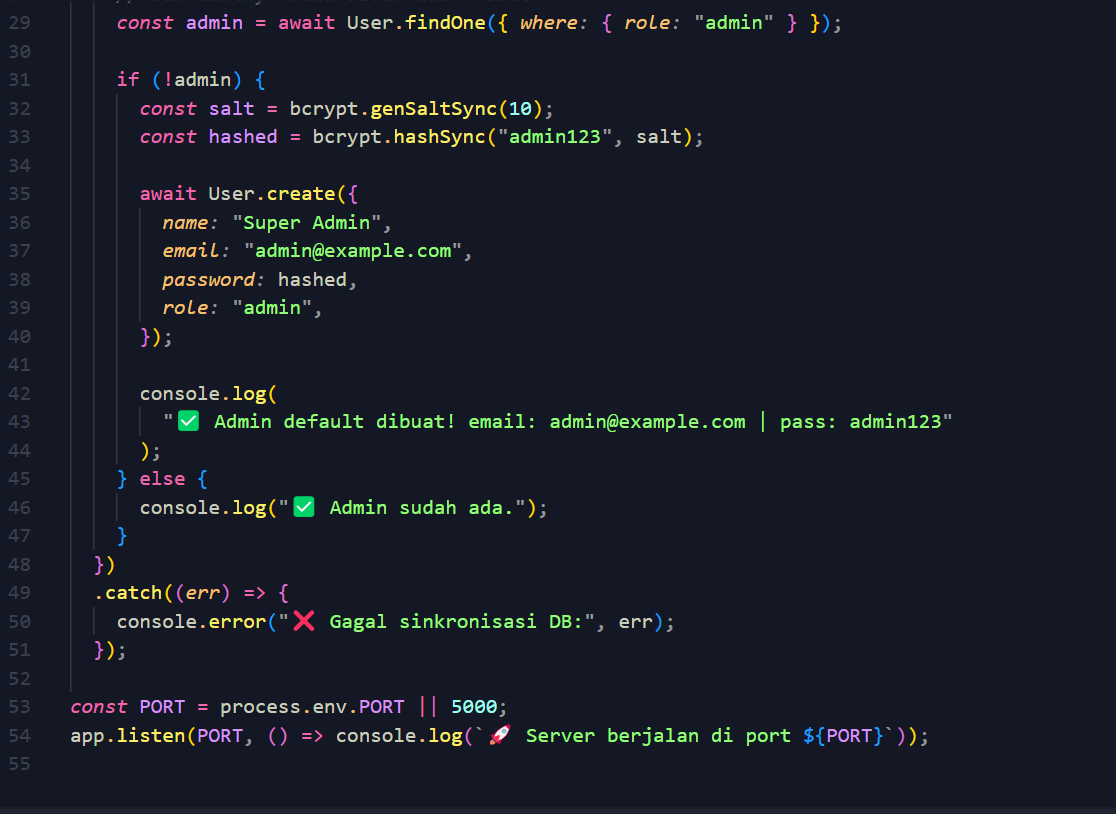
|  |  |  |
| --- | --- | --- |
| **Kolom** | **Tipe Data** | **Keterangan** |
| Id | INTEGER(PK) | Primary Key |
| Title | STRING(200) | Judulnovel |
| Author | STRING(100) | Nama penulis |
| Genre | STRING(100) | Genre novel |
| Decription | TEXT | Deskripsi lengkap |
| Pages | INTEGER | Jumlah halaman |
| Cover\_url | STRING(255) | URL gambar sampul |

1. **Penjelasan File Backend**
   1. **index.js**

File utama yang menjalankan aplikasi backend Express.  
Tugas utama:

1. Menyambungkan ke database
2. Menyiapkan route
3. Menyiapkan user admin default

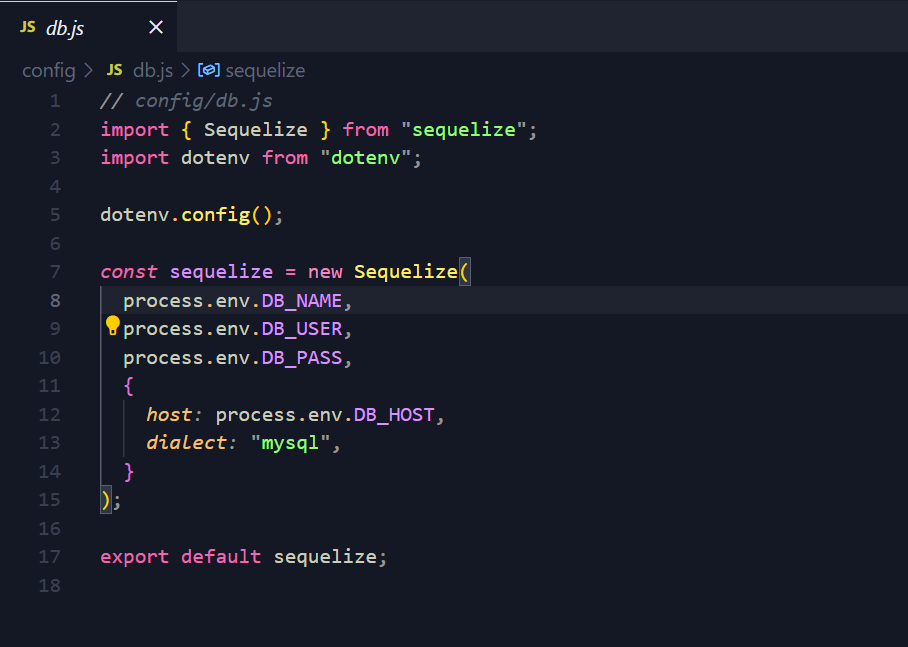
Gambar coding:



**5.2 config/db.js**

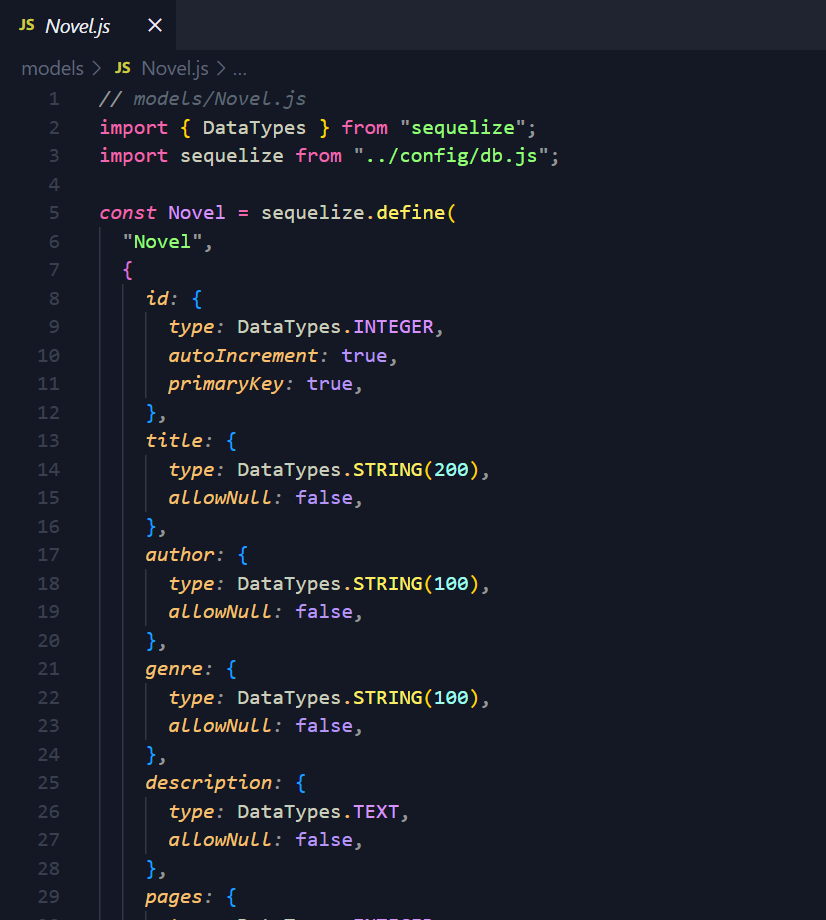
Konfigurasi koneksi ke database menggunakan Sequelize.

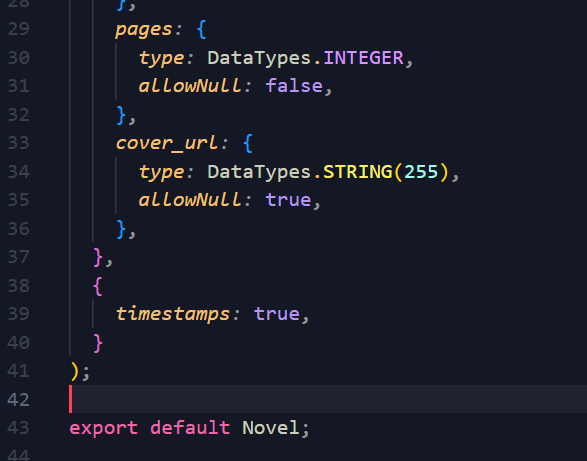
Gambar codingnya:

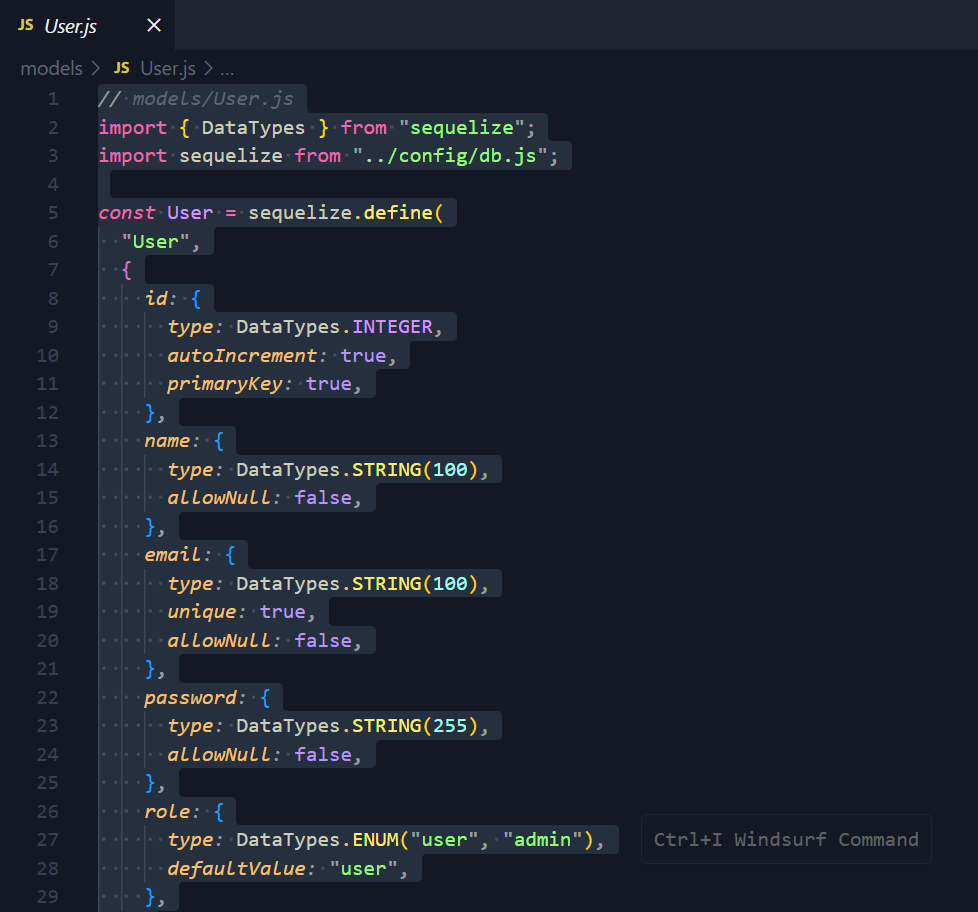


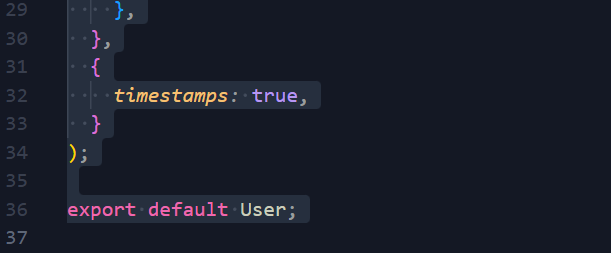
* 1. **models/User.js & models/Novel.js**
  2. User.js: Mendefinisikan model tabel user.
  3. Novel.js: Mendefinisikan model tabel novel.

Gambar codingnya:



****

****

****

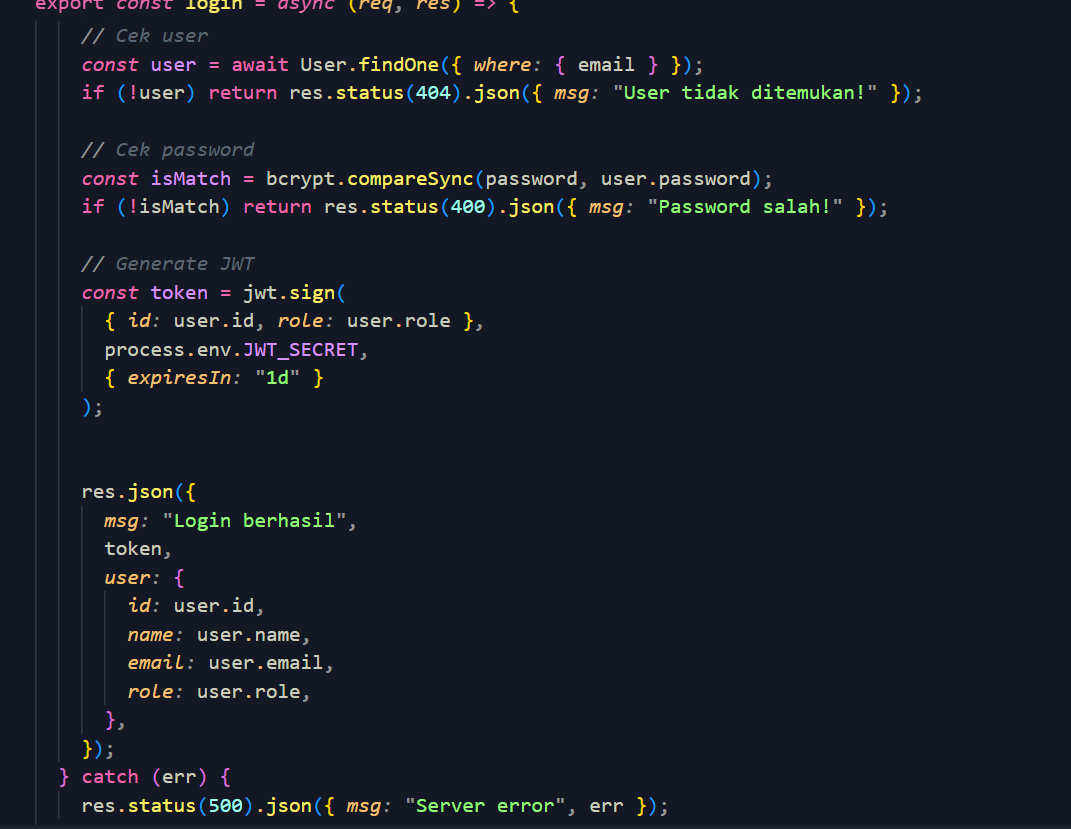
**5.4 controllers/authController.js**

Mengelola proses autentikasi:

1. register: Mendaftarkan user baru
2. login: Login dan menghasilkan token JWT

Gambar codingnya:



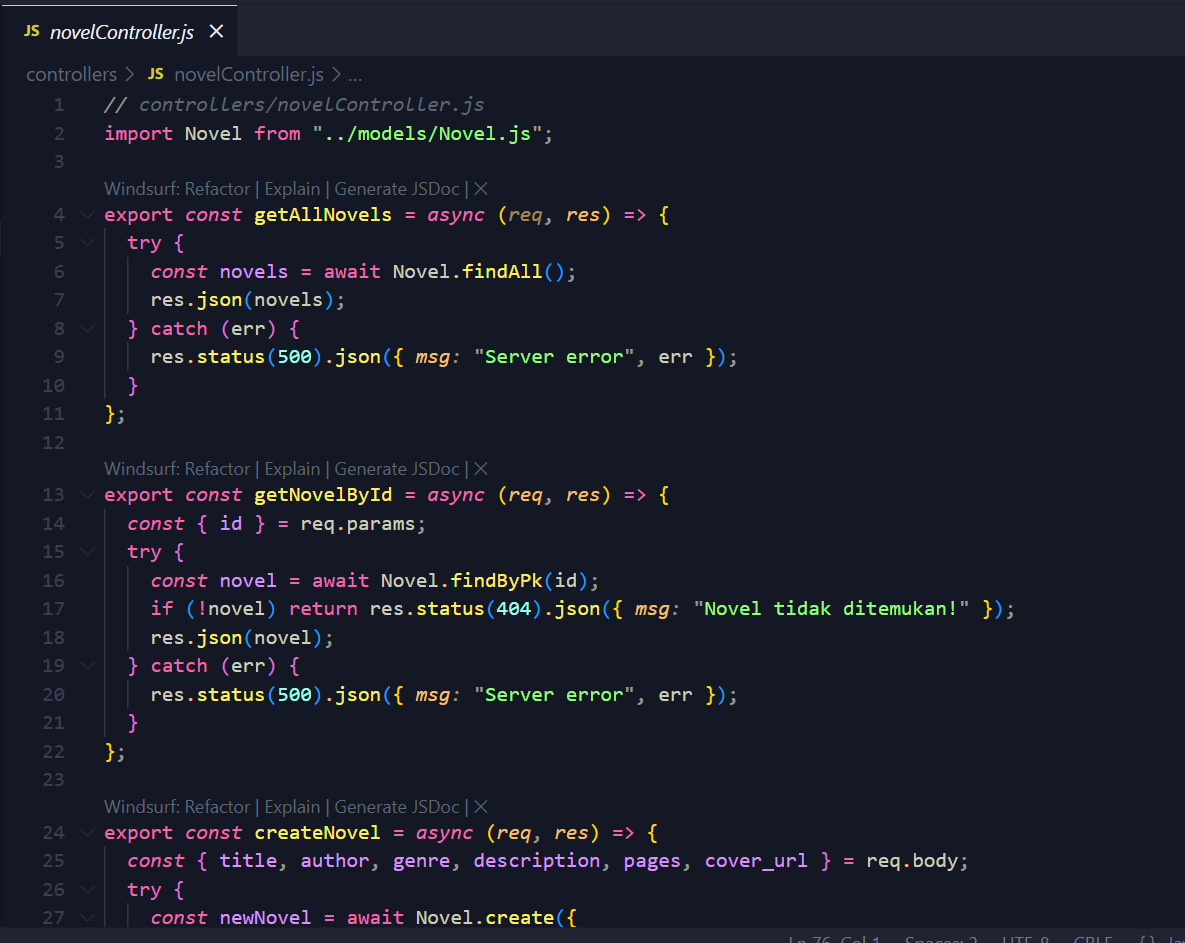


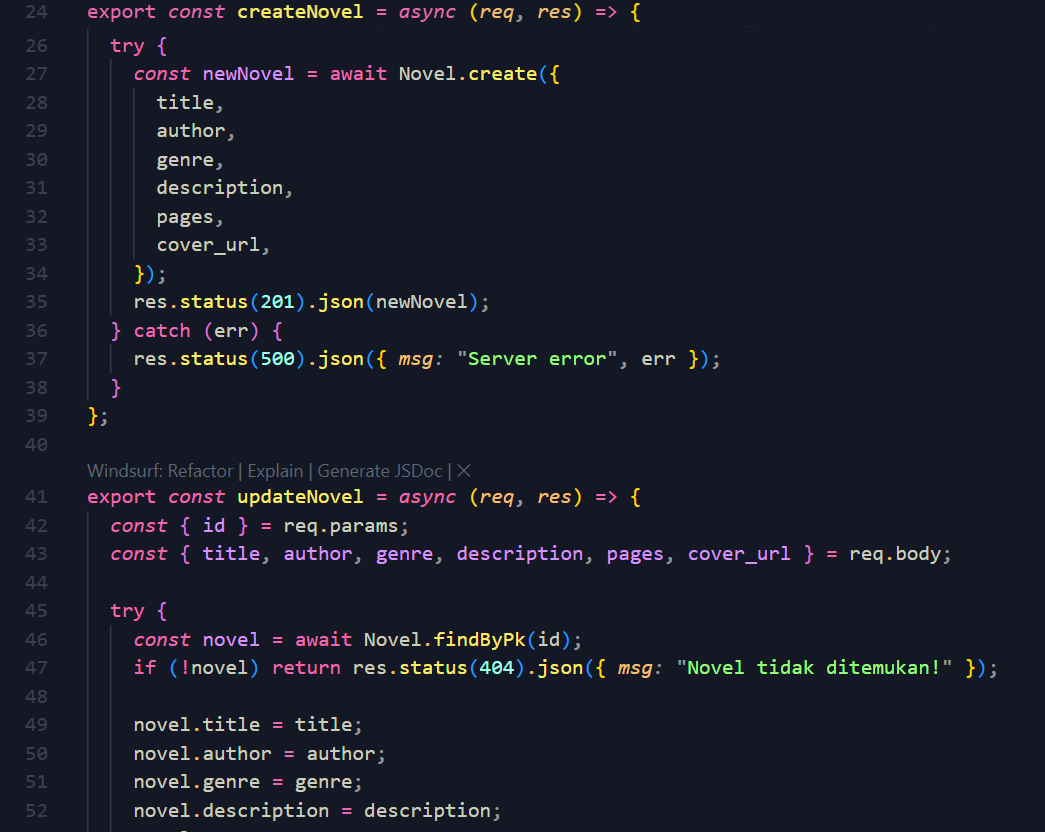
**5.5 controllers/novelController.js**

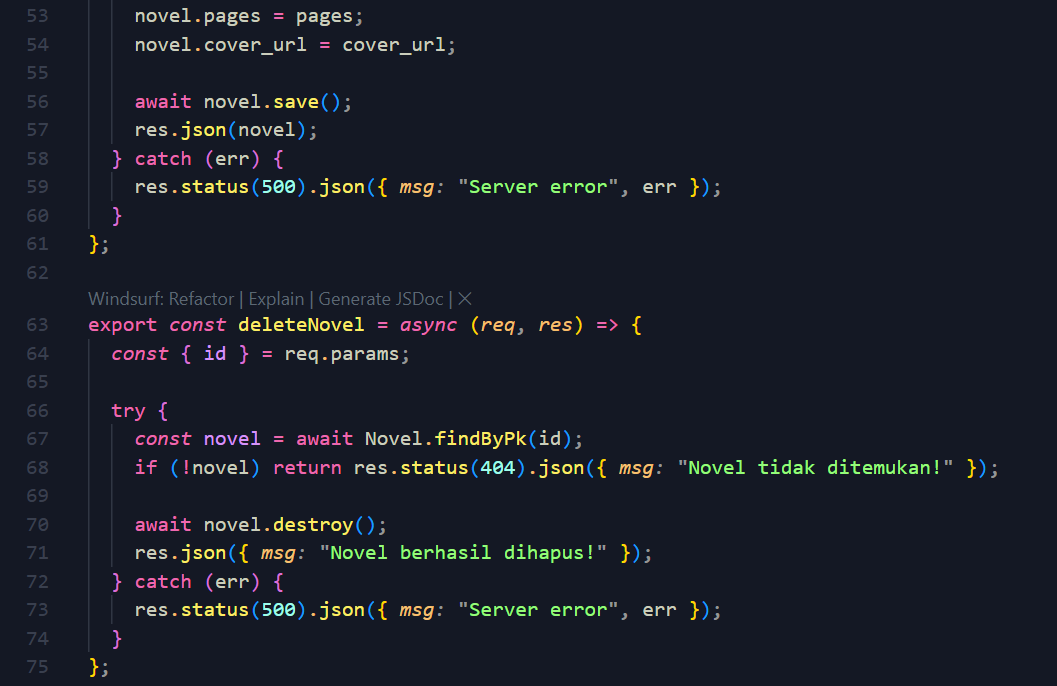
CRUD untuk data novel:

1. getAllNovels
2. getNovelById
3. createNovel
4. updateNovel
5. deleteNovel

Gaambar codingannya:





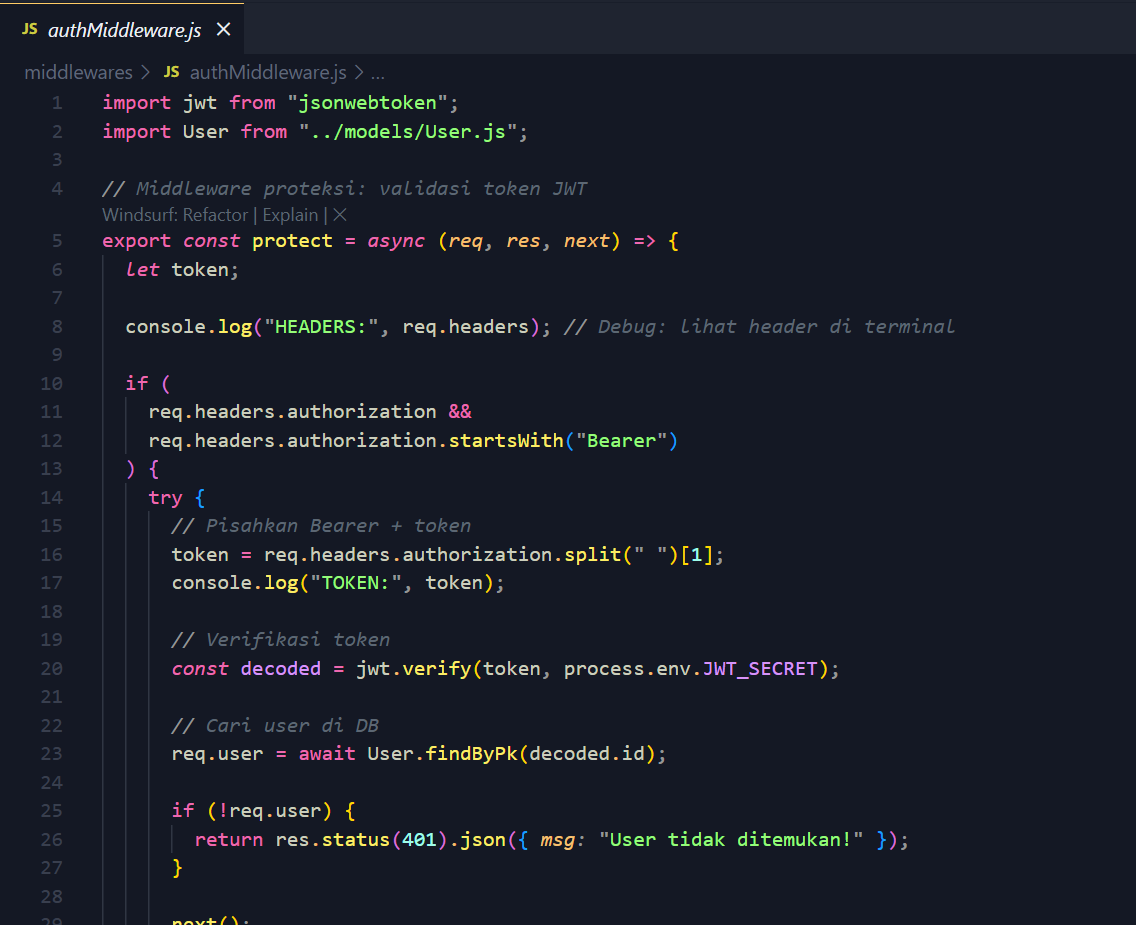


**5.6 middleware/authMiddleware.js**

Melindungi route dengan token JWT:

1. protect: Memastikan pengguna terautentikasi
2. adminOnly: Hanya admin yang bisa akses

Gambar codingannya:





1. **API Endpoint**

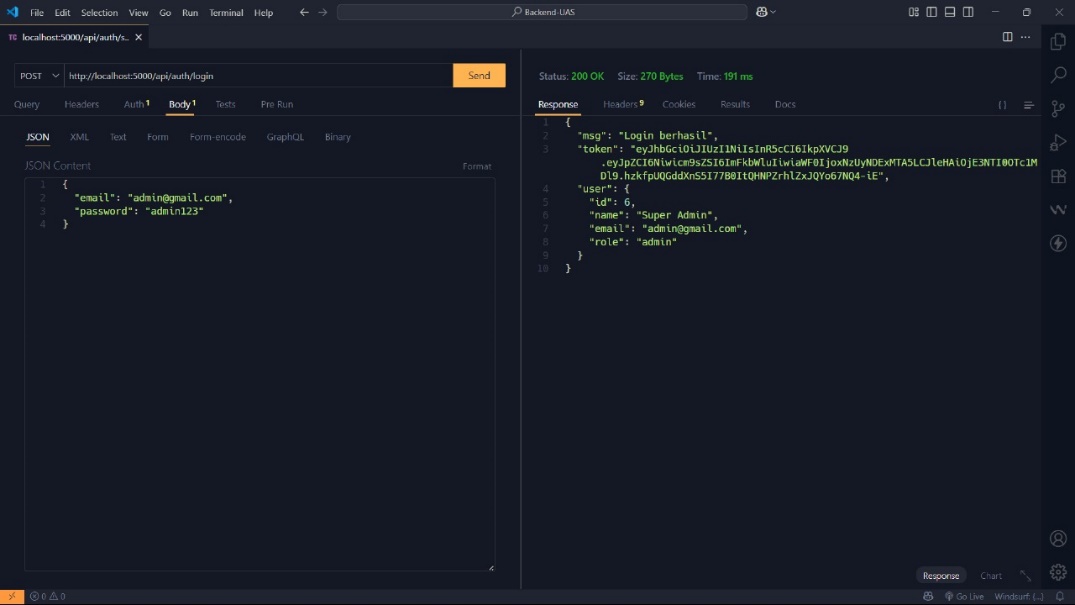
**Auth**

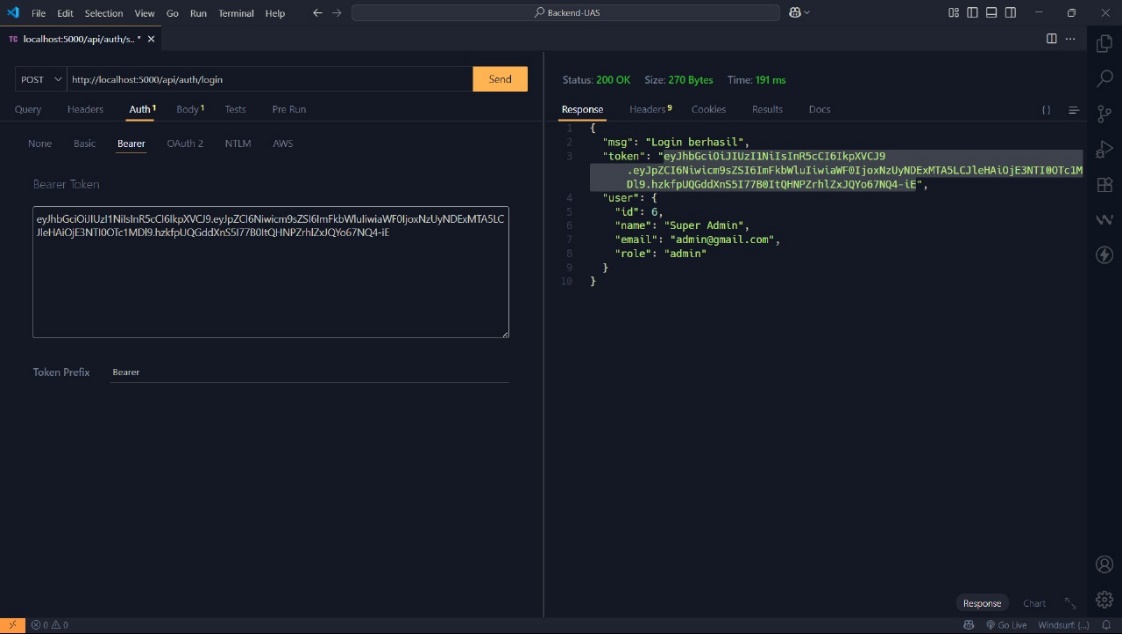
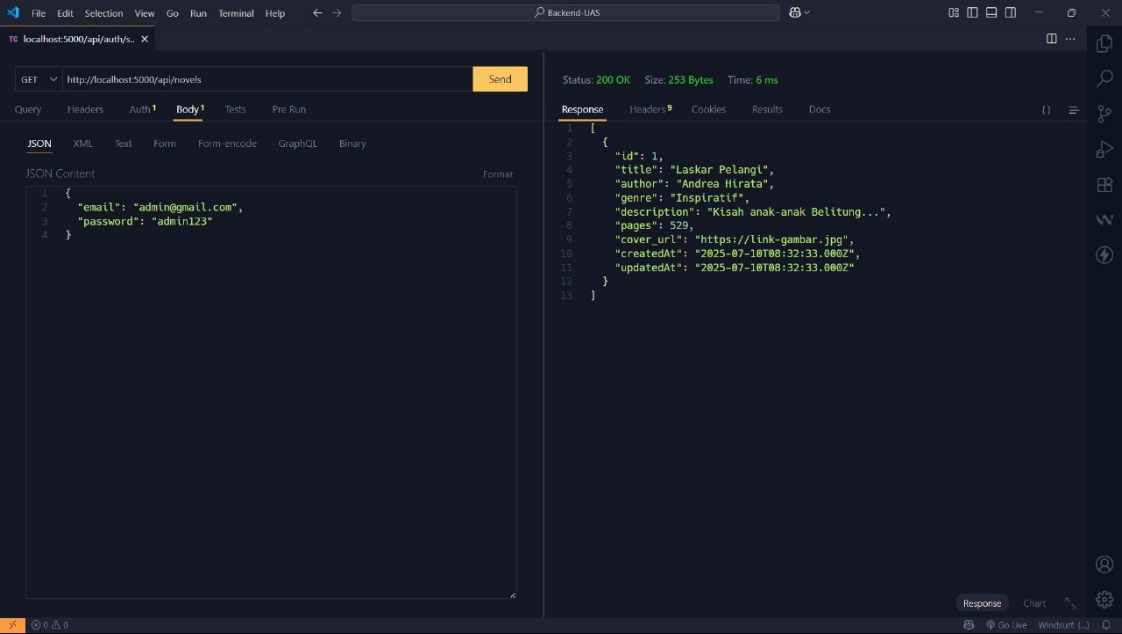
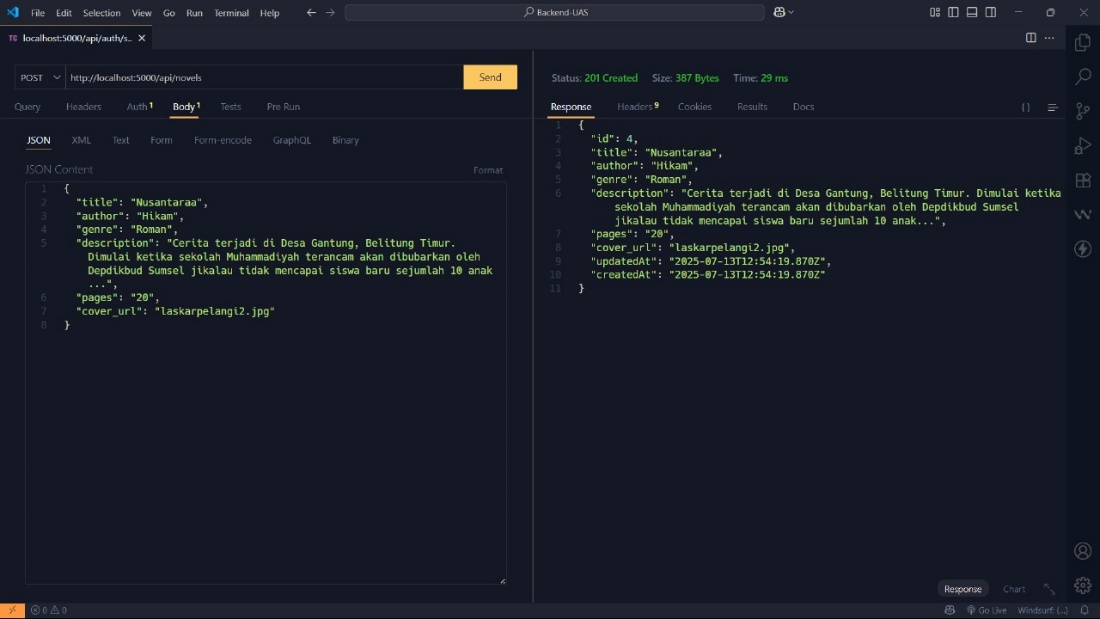
| **Method** | **Endpoint** | **Deskripsi** |
| --- | --- | --- |
| POST | /api/auth/register | Register user baru |
| POST | /api/auth/login | Login user |

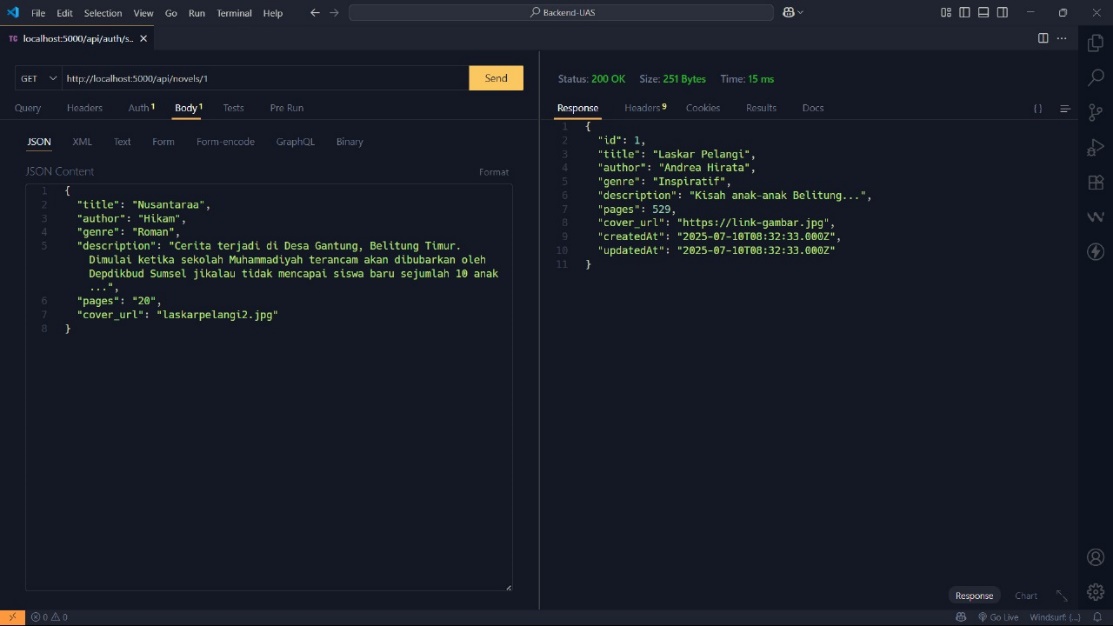
**Novel**

| **Method** | **Endpoint** | **Deskripsi** |
| --- | --- | --- |
| GET | /api/novels | Ambil semua novel |
| GET | /api/novels/:id | Ambil 1 novel berdasarkan ID |
| POST | /api/novels | Tambah novel (hanya admin) |
| PUT | /api/novels/:id | Edit novel (admin) |
| DELETE | /api/novels/:id | Hapus novel (admin) |

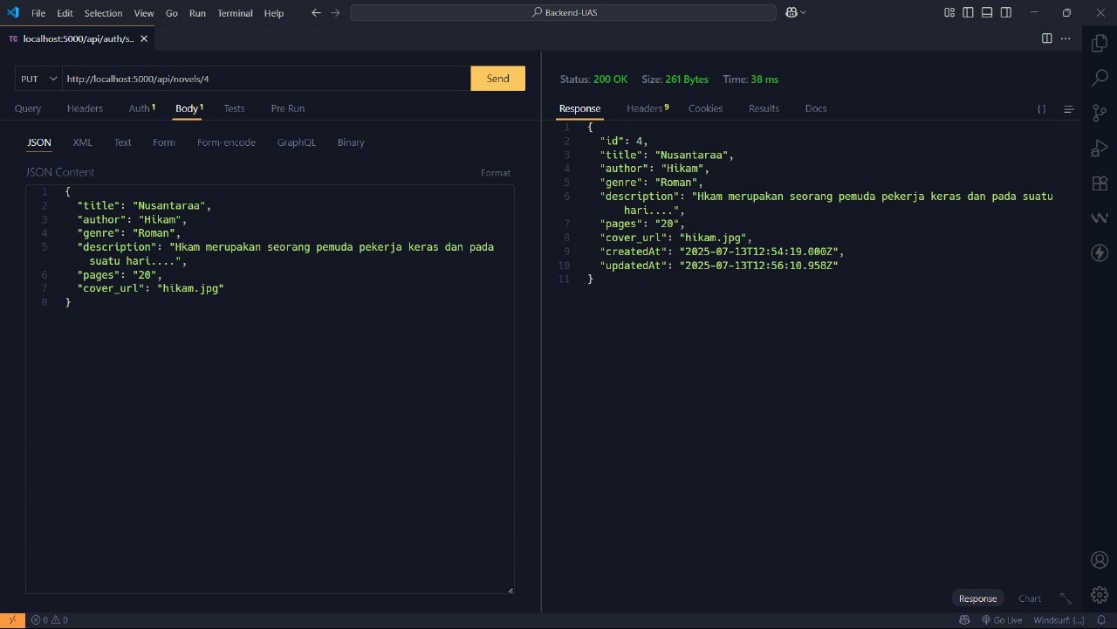
1. **Testing Postman**
2. **Admin**
   * 1. login admin dengan akun yg sudah otomatis ter generate

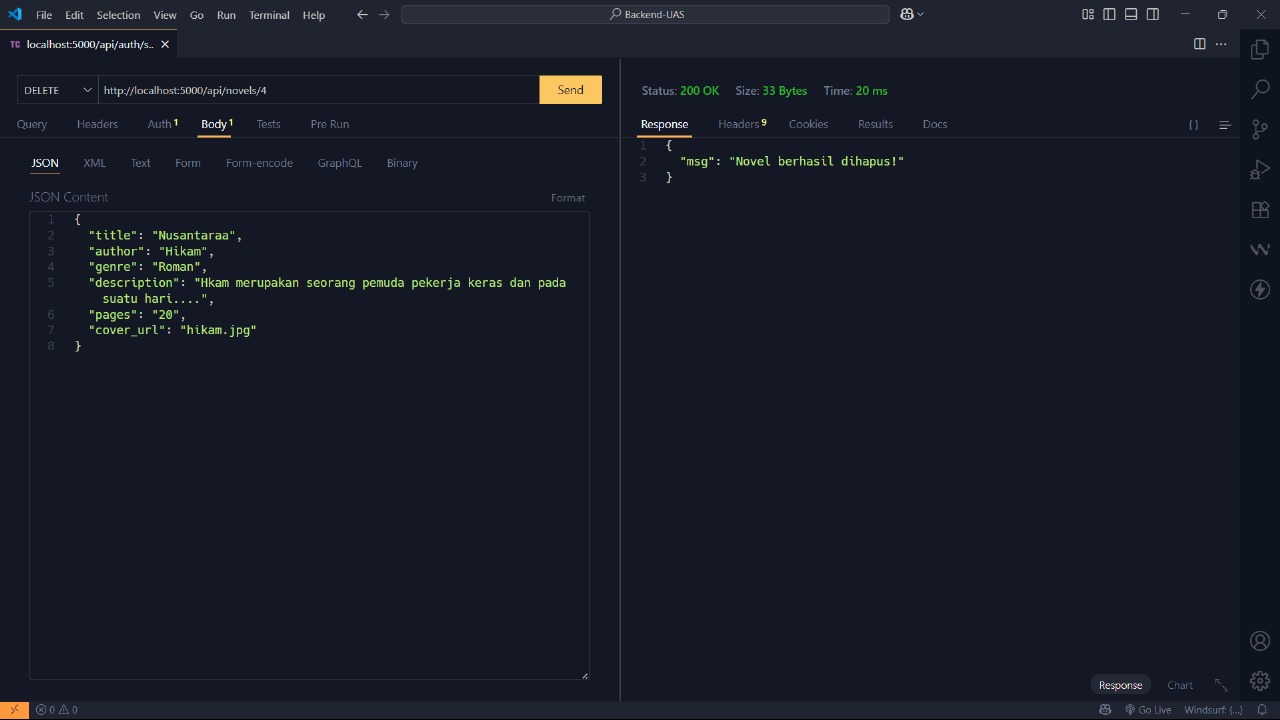


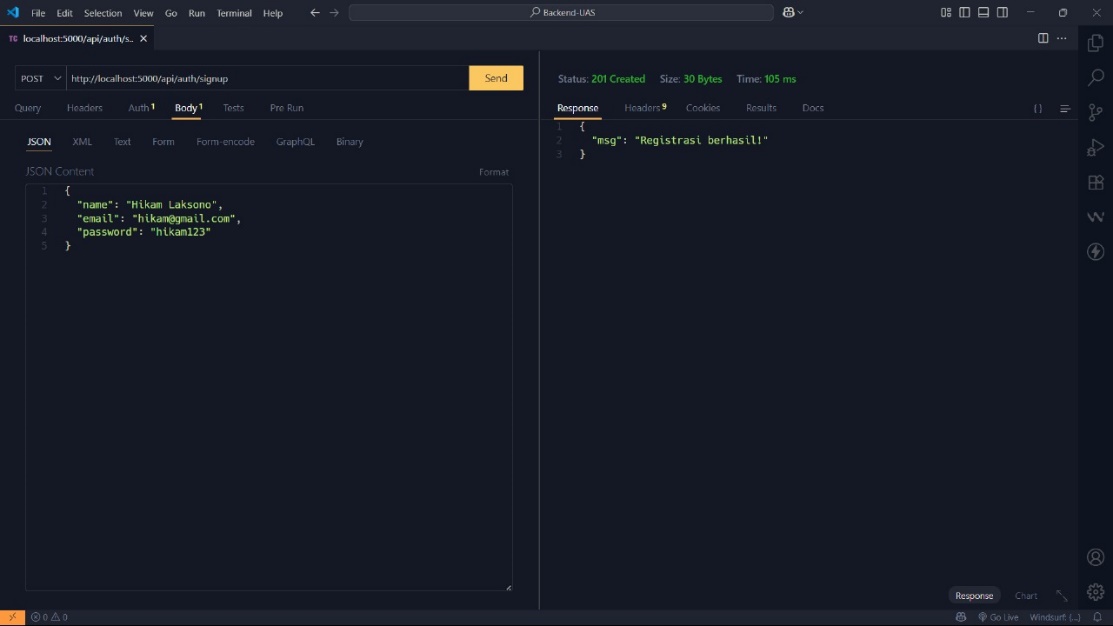
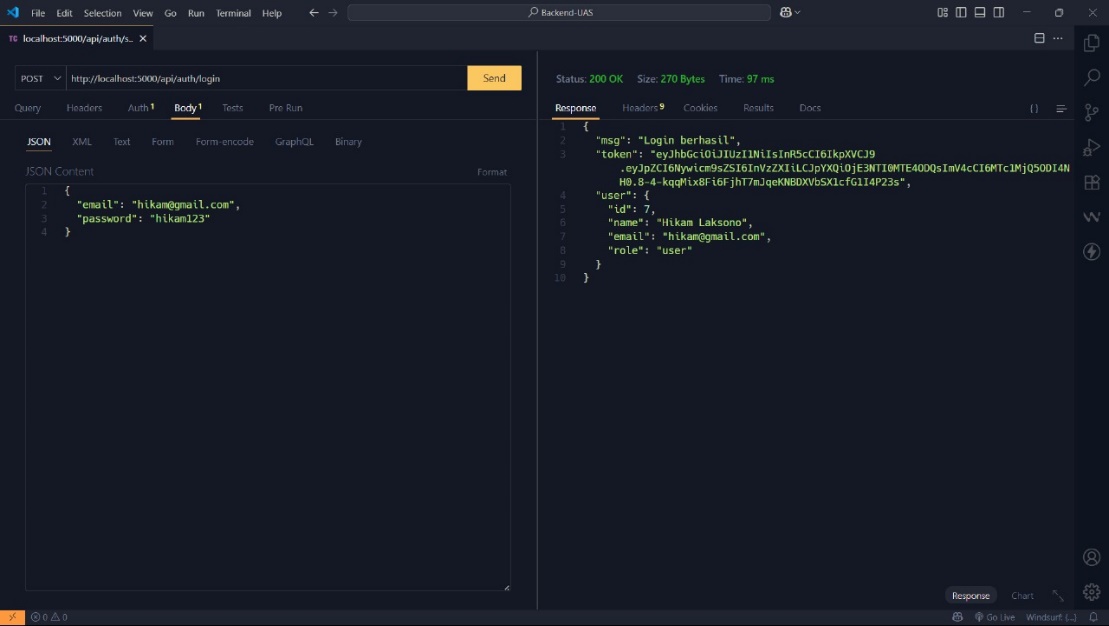
* + 1. **** setelah login berhasil akan ada response yang dimana terdapat sebuah token untuk autentikasi, taruh token yang didapat ke dalam bearer token di Auth agar mendeteksi bahwa akun memiliki token
    2. disini kita mulai untuk Get data setelah login dan Ketika berhasil akan menandakan bahwa token valid
    3. Lanjut untuk Post data baru
    4. Get data berdasarkan id

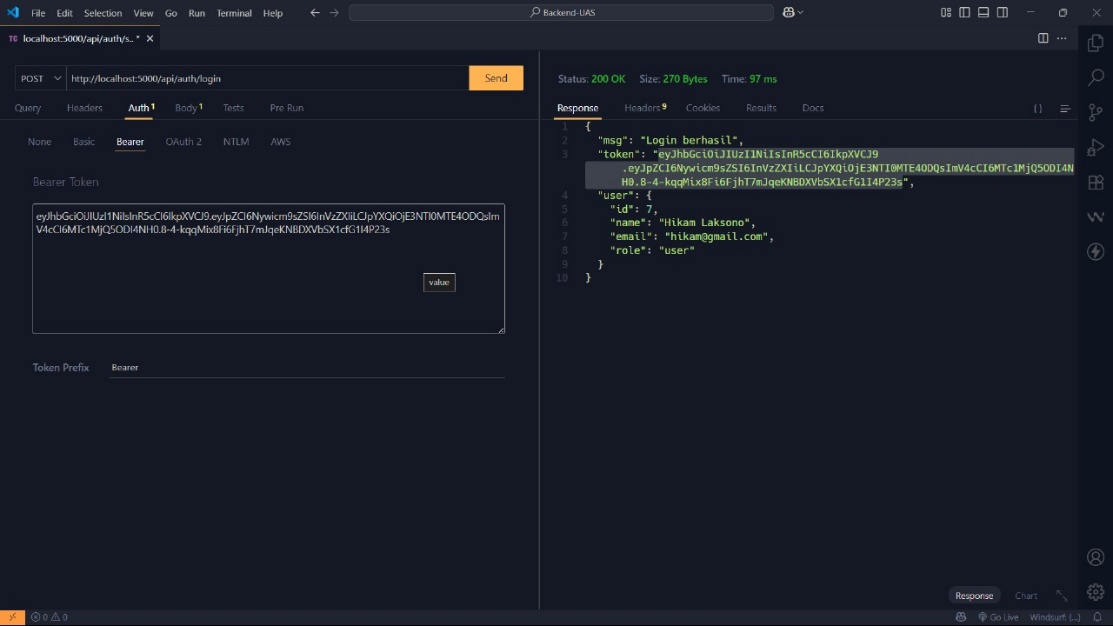


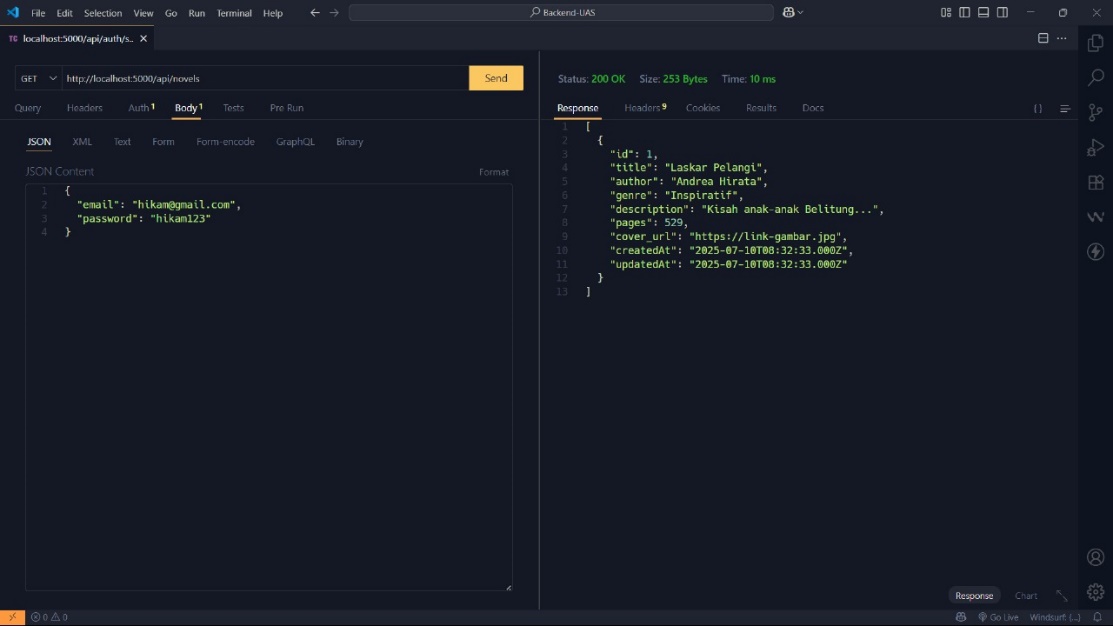
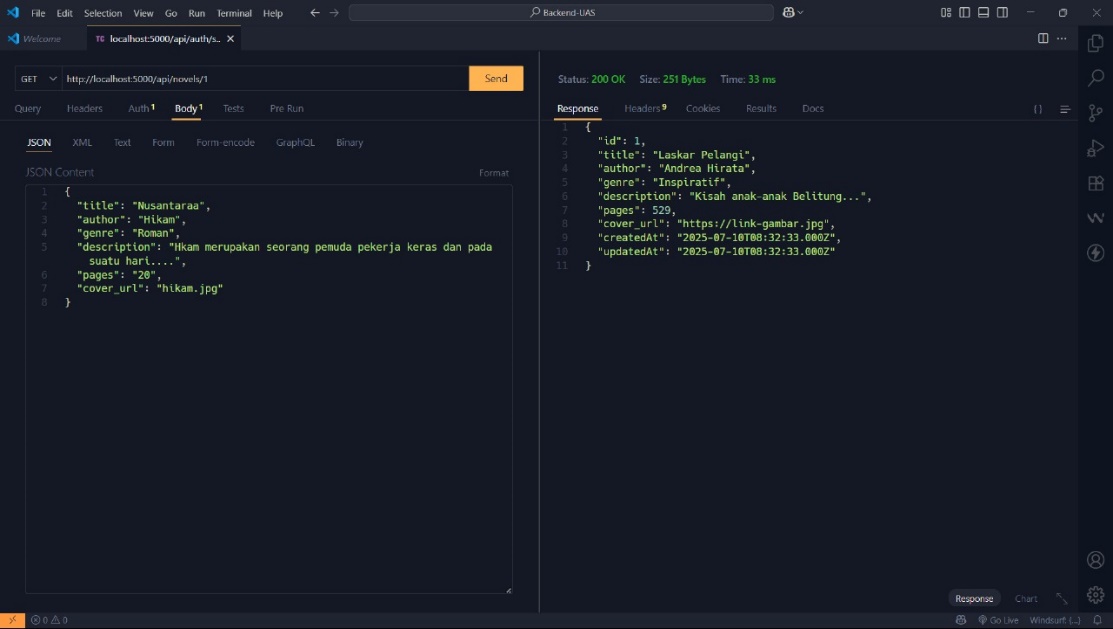
* + 1. edit data pada id 4

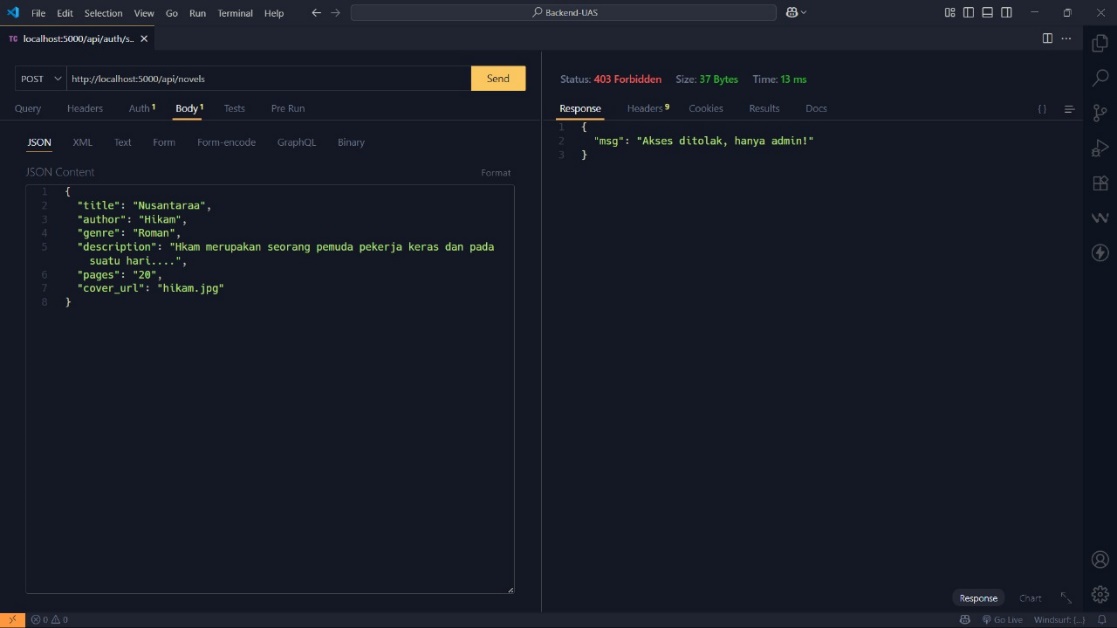


* + 1. Delete data id 4

1. **USER**
   * 1. untuk mengecek Autentikasi dan Autorisasi user pertama-tama kita membuat akun terlebih dahulu
     2. setelah pembuatan akun berhasil kita lanjut untuk masuk sebagai user yang telah terdaftar yg dimana user membawa sebuah token
     3. token user di taruh k dalam Bearer Token di Auth



* + 1. kita coba untuk get data novels dimana user hanya bisa melihat data novel yang ada
    2. dan ini get data berdasarkan id, yg dimana Get data berdasarkan id hanya boleh untuk user dan admin
    3. Bisa dilihat Ketika mencoba post data maka akan ada alert/penolakan akses dikarenakan terdapat autorisasi user hanya boleh Get all data dan Get data berdasarkan id



1. **Fitur Keamanan**
   1. Password dienkripsi menggunakan **bcryptjs**.
   2. Akses API dilindungi dengan **JWT** token.
   3. Role user dibatasi, hanya admin yang dapat mengelola data novel.
2. **Inisialisasi Admin Default**

Ketika aplikasi pertama dijalankan, sistem akan otomatis mengecek apakah ada user dengan role **admin**. Jika belum ada, maka akan dibuat:

1. Email: admin@example.com
2. Password: admin123
3. **Kesimpulan**

Backend dari sistem informasi manajemen novel ini sudah menyediakan fitur login, registrasi, serta pengelolaan data novel dengan kontrol akses berbasis role. Dengan menggunakan Express.js dan Sequelize, proses pengembangan backend menjadi lebih cepat dan terstruktur.