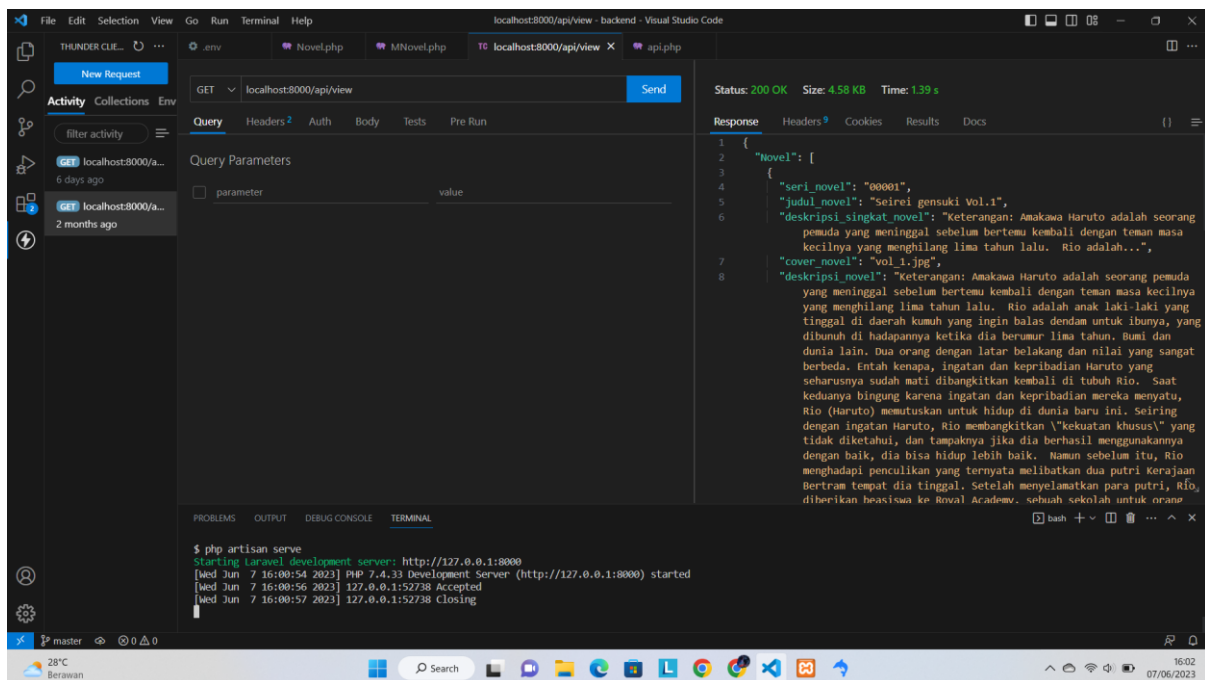
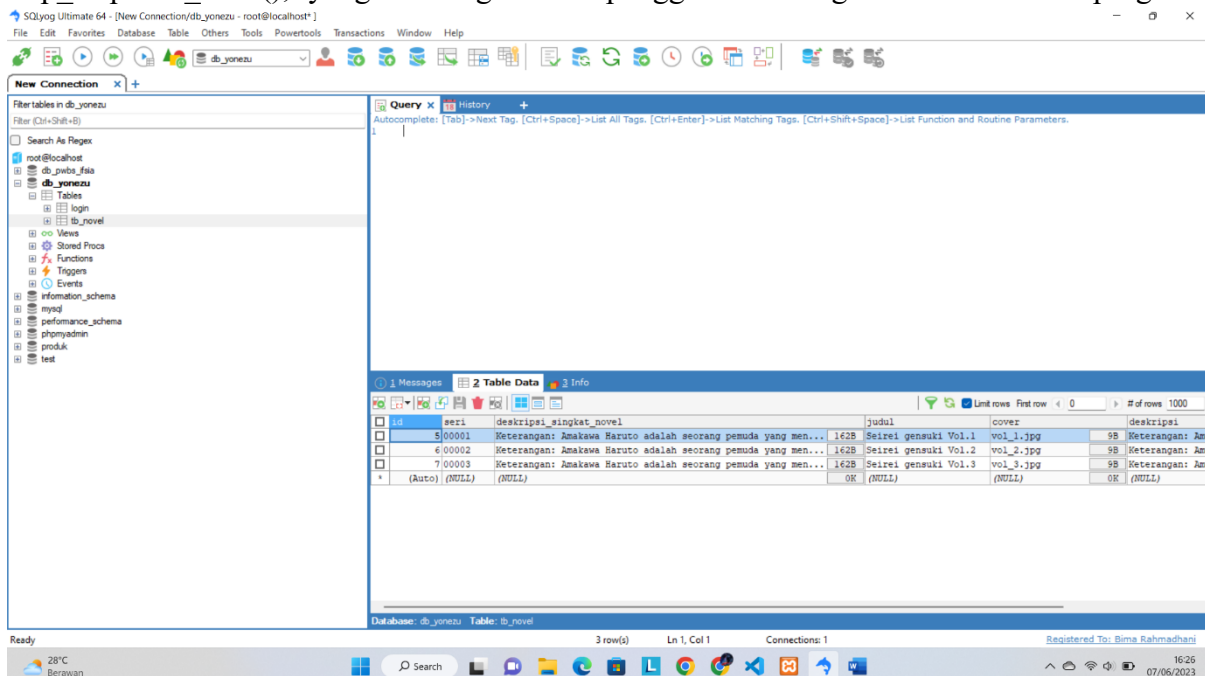


VIEW

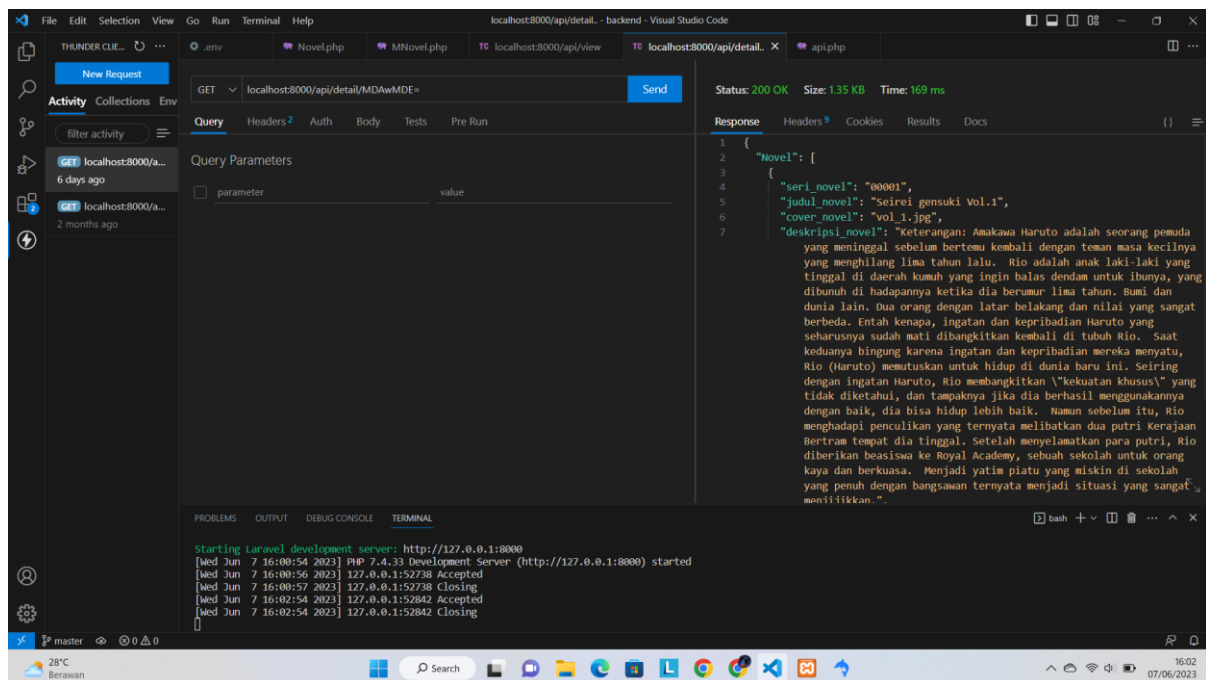


Program view ini digunakan untuk menampilkan data dari tabel "tb_novel". Pertama, method ini memanggil method `viewData()` dari objek `$this->model` (objek class MNovel) untuk mengambil data. Data yang diperoleh kemudian disimpan dalam variabel `$data`. Selanjutnya, method ini mengembalikan sebuah response dalam bentuk JSON yang berisi data novel yang telah diambil. Status code HTTP yang digunakan dalam response ditentukan oleh `http_response_code()`, yang kemungkinan dipanggil dari fungsi lain di dalam program.



Selanjutnya, fungsi ini menggunakan query builder dari Laravel untuk membuat query yang mengambil kolom-kolom tertentu dari tabel. Hasil query tersebut kemudian diurutkan berdasarkan kolom "seri_novel" dan dikembalikan.

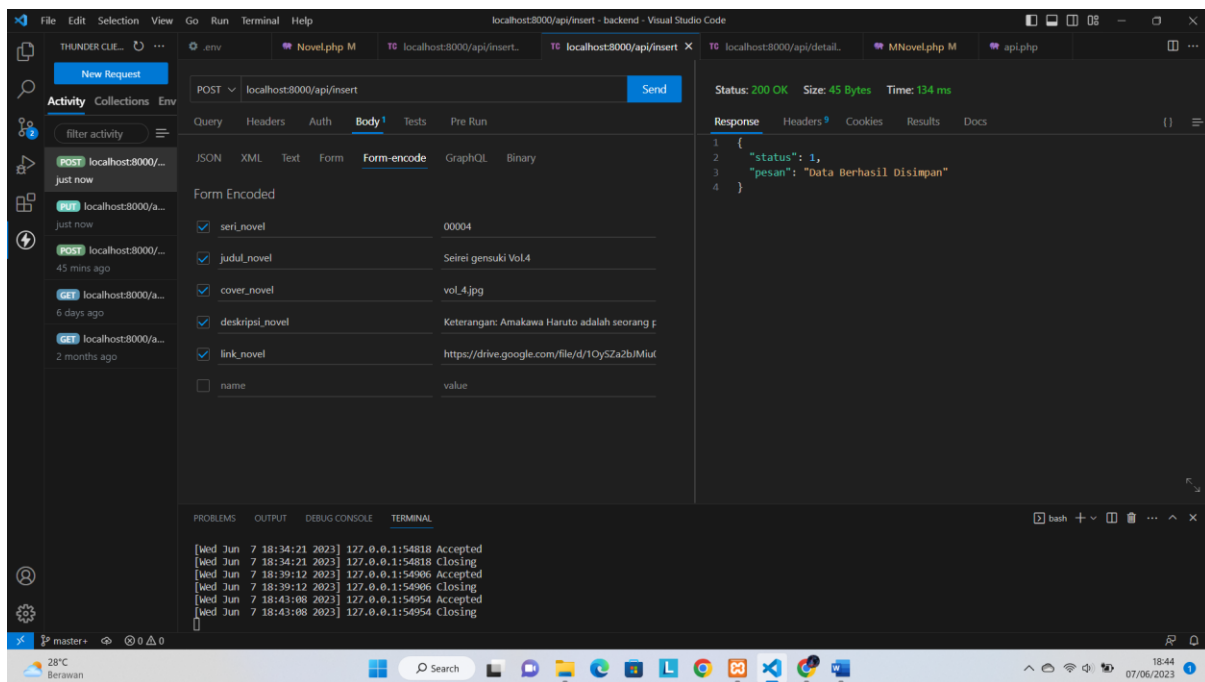
DETAIL



program ini digunakan untuk menampilkan detail data novel berdasarkan parameter yang diberikan. Method ini pertama-tama memanggil method `detailData($parameter)` dari objek `$this->model` untuk mengambil data detail novel. Data tersebut disimpan dalam variabel `$data`. Kemudian, method ini mengembalikan sebuah response dalam bentuk JSON yang berisi data novel detail yang telah diambil. Status kode HTTP yang digunakan dalam response ditentukan oleh `http_response_code()`, yang kemungkinan dipanggil dari fungsi lain di dalam program.

Serta Fungsi ini digunakan untuk mengambil detail data novel berdasarkan parameter yang diberikan. Fungsi ini juga menggunakan query builder untuk membuat query yang mengambil kolom-kolom tertentu dari tabel "tb_novel" berdasarkan kondisi WHERE. Parameter yang diberikan diubah menjadi base64 dan dibandingkan dengan data dalam tabel. Hasil query kemudian diurutkan berdasarkan kolom "seri" dan dikembalikan.

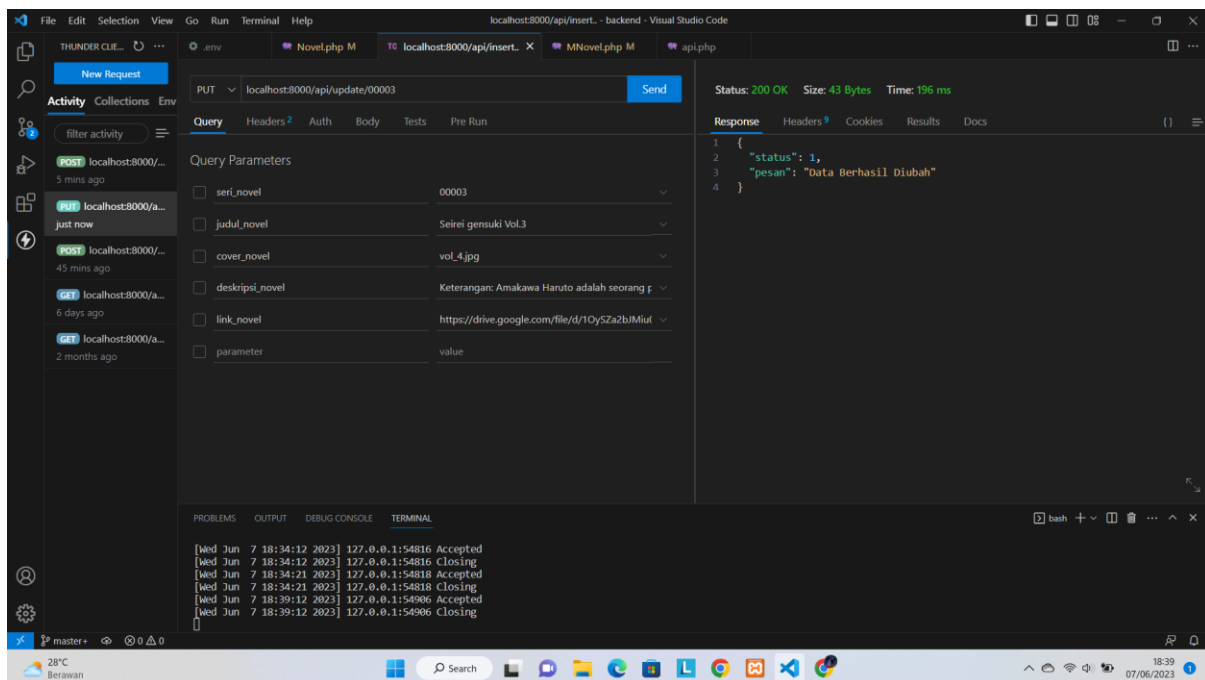
ADD



Program ini digunakan untuk menyimpan data novel baru. Data yang akan disimpan diambil dari input yang diberikan melalui objek `$req`. Data tersebut kemudian disimpan dalam variabel `$data`. Selanjutnya, method ini memeriksa apakah data novel dengan seri yang sama sudah pernah tersimpan sebelumnya menggunakan method `detailData($parameter)` dari objek `$this->model`. Jika data tidak ditemukan, method `saveData()` dari objek `$this->model` akan dipanggil untuk menyimpan data novel. Setelah itu, method ini mengembalikan sebuah response dalam bentuk JSON yang berisi status (berhasil atau gagal) dan pesan terkait hasil operasi penyimpanan data.

Selanjutnya, `saveData($seri, $deskripsi_singkat, $judul, $cover, $deskripsi, $link)`: Fungsi ini digunakan untuk menyimpan data novel baru ke dalam tabel "tb_novel". Fungsi ini menggunakan query builder untuk membuat query yang melakukan operasi INSERT dengan nilai-nilai yang diberikan sebagai parameter.

UPDATE

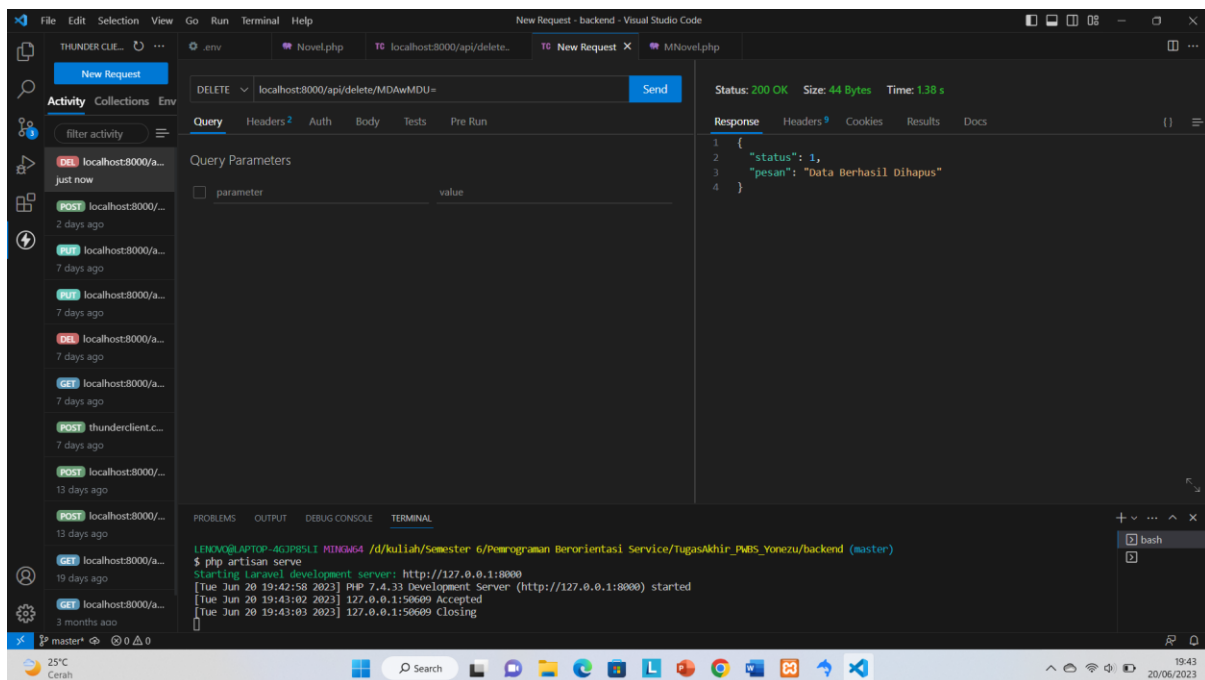


Program ini digunakan untuk mengubah data novel berdasarkan parameter yang diberikan. Data yang akan diubah diambil dari input yang diberikan melalui objek `$req` dan disimpan dalam variabel `$data`. Method ini pertama-tama memeriksa apakah data novel dengan seri yang sama sudah pernah tersimpan atau belum menggunakan method `checkUpdate($parameter, $data["seri"])` dari objek `$this->model`. Jika data tidak ditemukan, method `updateData()` dari objek `$this->model` akan dipanggil untuk mengubah data novel. Setelah itu, method ini mengembalikan sebuah response dalam bentuk JSON yang berisi status (berhasil atau gagal) dan pesan terkait hasil operasi pembaruan data.

Sebelum diupdate program akan mengecek menggunakan `checkUpdate($seri_lama, $seri_baru)`: Fungsi ini digunakan untuk memeriksa apakah data novel dengan seri baru yang ingin diubah sudah ada dalam tabel atau tidak. Fungsi ini menggunakan query builder untuk membuat query yang mengambil data seri yang sama dengan seri baru, tetapi berbeda dari seri lama. Hasil query ini kemudian dikembalikan.

Dan di update untuk membuat query melakukan operasi UPDATE dengan nilai-nilai yang diberikan sebagai parameter. Data yang akan diubah dipilih berdasarkan seri lama, yang diubah menjadi base64/atribut data untuk dibandingkan dengan data dalam tabel.

Delete



Tanda fungsi: function delete(\$parameter).

Fungsi ini menerima parameter `$parameter`, yang digunakan untuk mengidentifikasi data yang akan dihapus.

Memeriksa ketersediaan data:

Kode tersebut pertama-tama memeriksa apakah data (yang diidentifikasi oleh parameter) ada dengan memanggil metode `detailData()` dari sebuah model (tidak ditampilkan dalam potongan kode yang diberikan).

Hasilnya disimpan dalam variabel `$cek_data`.

Menghapus data:

Jika data ditemukan (yaitu jumlah `$cek_data` sama dengan 1), kode tersebut melanjutkan untuk menghapus data dengan memanggil metode `deleteData()` dari model.

Data dihapus berdasarkan parameter yang diberikan.

Mengatur status dan pesan:

Setelah menghapus data, kode tersebut mengatur variabel `$status` dan `$pesan` untuk mengindikasikan status dari proses penghapusan.

Jika data berhasil dihapus, `$status` diatur menjadi 1 dan `$pesan` diatur menjadi "Data Berhasil Dihapus".

Jika data tidak ditemukan, `$status` diatur menjadi 0 dan `$pesan` diatur menjadi "Data Gagal Dihapus (Seri Novel Tidak Ditemukan!)".

Mengembalikan respons:

Kode tersebut mengembalikan respons JSON yang berisi variabel status dan pesan, yang mewakili status dan pesan dari proses penghapusan.

Kode respons HTTP diatur berdasarkan kode respons HTTP saat ini.

Tanda fungsi: function deleteData(\$parameter).

Fungsi ini menerima parameter \$parameter, yang digunakan sebagai nilai untuk mencocokkan data yang akan dihapus.

Menghapus data dari tabel:

Kode tersebut menggunakan metode delete() untuk menghapus data dari tabel tb_novel.

Pada bagian DB::table('tb_novel'), DB::table() digunakan untuk membuat instance dari objek query builder untuk tabel tb_novel.

Mengatur kondisi penghapusan:

Pada bagian ->where(DB::raw("TO_BASE64(seri)"), "=", \$parameter), digunakan untuk menentukan kondisi penghapusan data.

DB::raw("TO_BASE64(seri)") digunakan untuk melakukan ekspresi SQL yang mengubah kolom seri menjadi format BASE64.

\$parameter digunakan untuk membandingkan nilai dalam kolom seri dengan nilai parameter yang diberikan.

Melakukan penghapusan:

Pada bagian ->delete(), metode delete() dipanggil untuk menjalankan operasi penghapusan pada tabel tb_novel.

Baris ini akan menghapus semua baris yang memenuhi kondisi yang ditentukan sebelumnya.