|  |  |  |
| --- | --- | --- |
| Mata Kuliah | **:** | Pemrograman Web Lanjut (PWL) |
| Program Studi | **:** | D4 – Teknik Informatika **/** D4 – Sistem Informasi Bisnis |
| Semester | **:** | 4 (empat) / 6 (enam) |
| Pertemuan ke- | **:** | 11 (sebelas) |

# JOBSHEET 11

**RESTFUL API 2**

Sebelumnya kita sudah membahas mengenai *RESTFUL API dan JSON Web Token (JWT)* pada Laravel. Dimana kita telah membuat RESTful API Register, RESTful API Login, RESTful API Logout, dan implementasi CRUD dalam RESTful API pada halaman web. Pada pertemuan kali ini, kita akan mempelajari penerapan RESTFUL API lanjutan di dalam project Laravel.

Sebelum kita masuk materi, kita buat dulu project baru yang akan kita gunakan untuk membangun aplikasi sederhana dengan topik *Point of Sales (PoS),* sesuai dengan **Studi Kasus PWL.pdf**.

Jadi kita bikin project Laravel 10 dengan nama **PWL\_POS.**

*Project* **PWL\_POS** akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

# A. ELOQUENT ACCESSOR

Laravel memiliki fitur yang bernama mutator, accessor dan casting, fitur-fitur ini digunakan untuk melakukan manipulasi data di dalam attribute database dengan sangat mudah. Contohnya pada saat insert data dengan enkripsi ke dalam database dan melakukan deskripsi saat menampilkan dari database secara otomatis.

Accessor dapat mengubah nilai saat attribute atau field eloquent diakses. Untuk mendefinisikan Accessor dapat membuat method di dalam model untuk menentukan attribute yang akan diakses. Nama method yang dibuat harus sama dengan nama attribute yang akan di format: contoh :

Attribute/field pada tabel first\_name maka methodnya firstName()

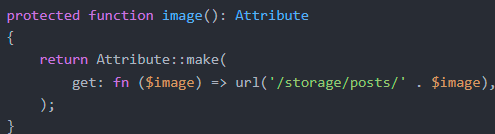
protected function firstName(): Attribute

{

//...

}

Jika membuat attribute/field image yang ada ditable m\_user kita akan memberikan nilai full path dari direktori dimana file gambar tersebut disimpan.contohnya pada UserModel ditambahkan



Dengan begitu dapat melakukan import Eloquent Attribute dengan



Lalu method baru dengan nama image() melakukan return path nama file image itu berada



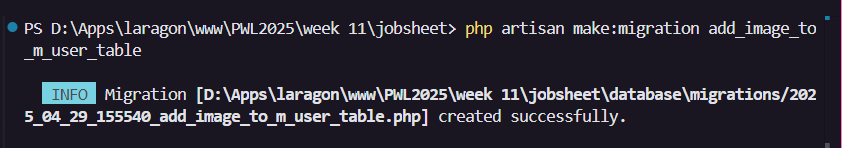
Hasil akhir memanggil attribute image



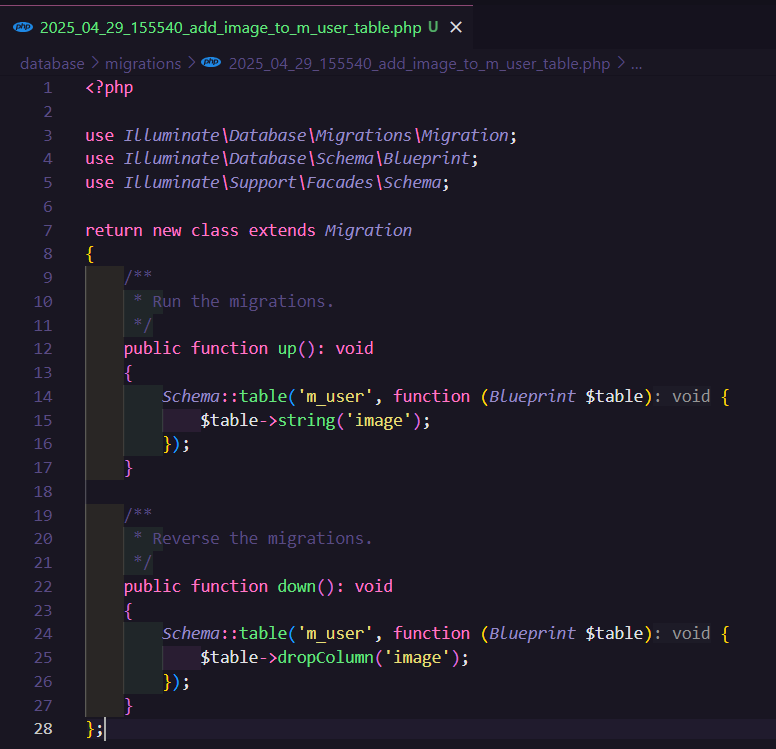
**Praktikum 1** – Implementasi Eloquent Accessor

1. Sebelum memulai pastikan REST API, terlebih dahulu pastikan sudah ter instal aplikasi Postman.
2. Kita akan memodifikasi Table m\_user dengan menambahkan column : image, buka terminal lalu ketikkan

**php artisan make:migration add\_image\_to\_m\_user\_table**

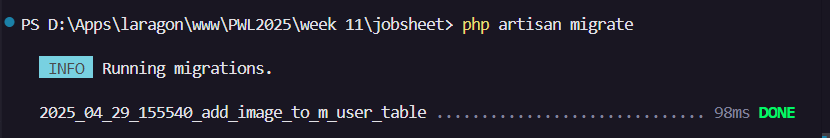
****

1. Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan:

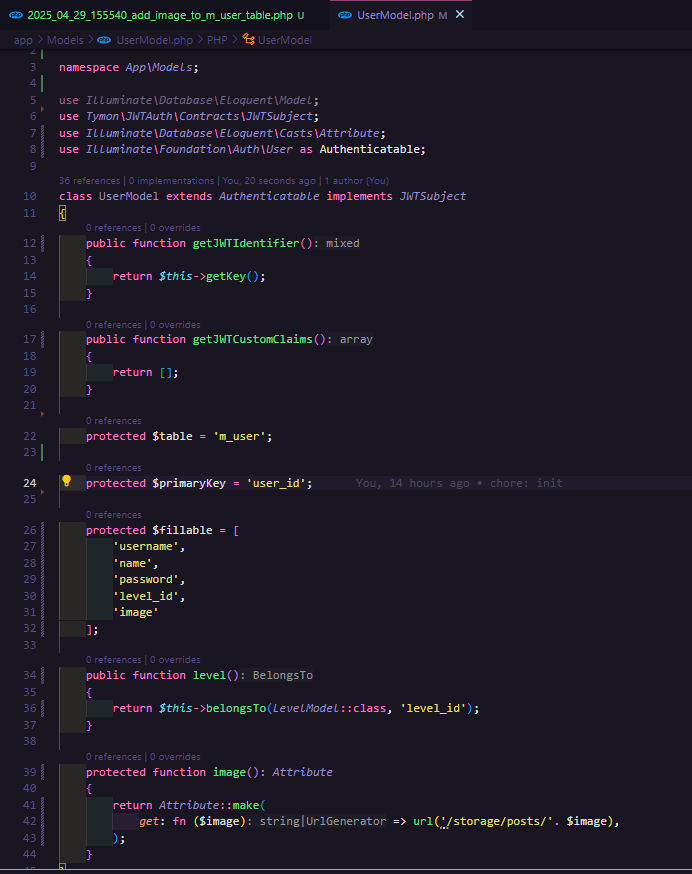


1. Lakukan jalankan update migrasi dengan cara:

php artisan migrate



1. Lalu lakukan modifikasi models pada App/Models/UserModel.php



1. Lakukan modifikasi pada Controllers/Api/RegisterController



1. Anda dapat menambahkan detail untuk spesifikasi image pada validator

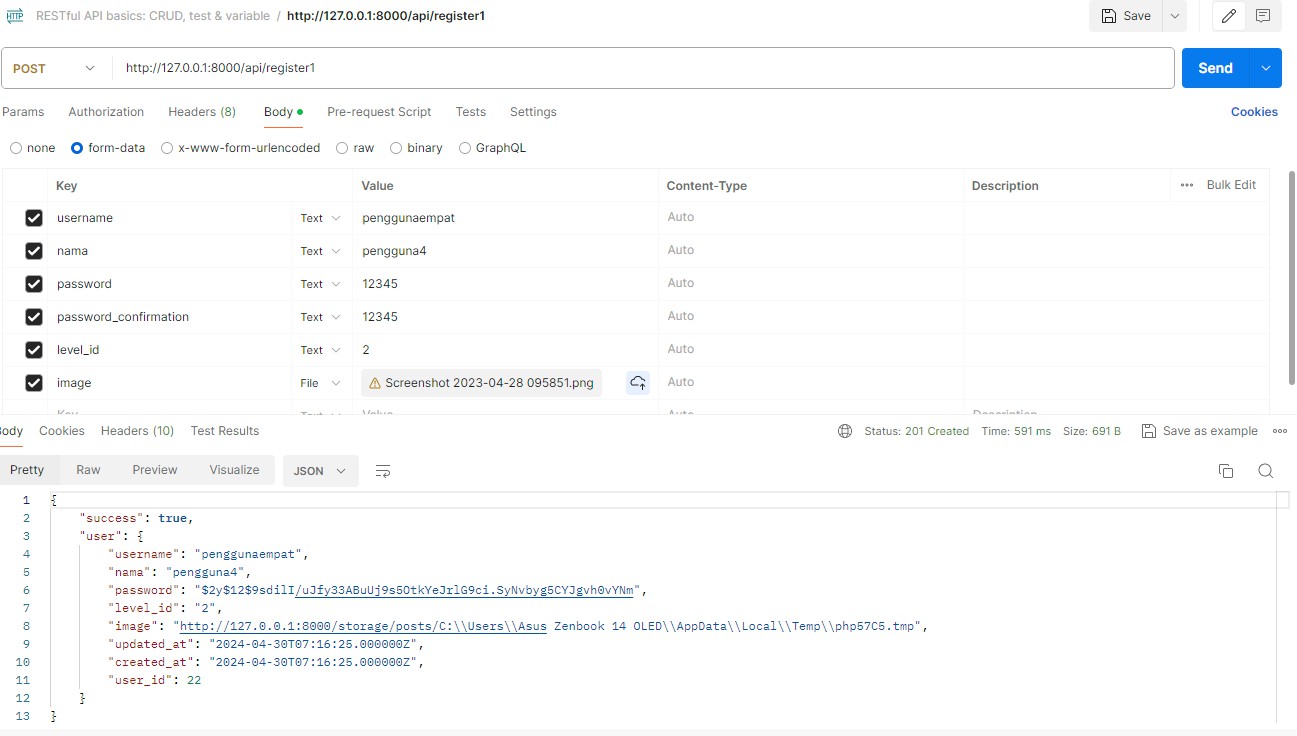


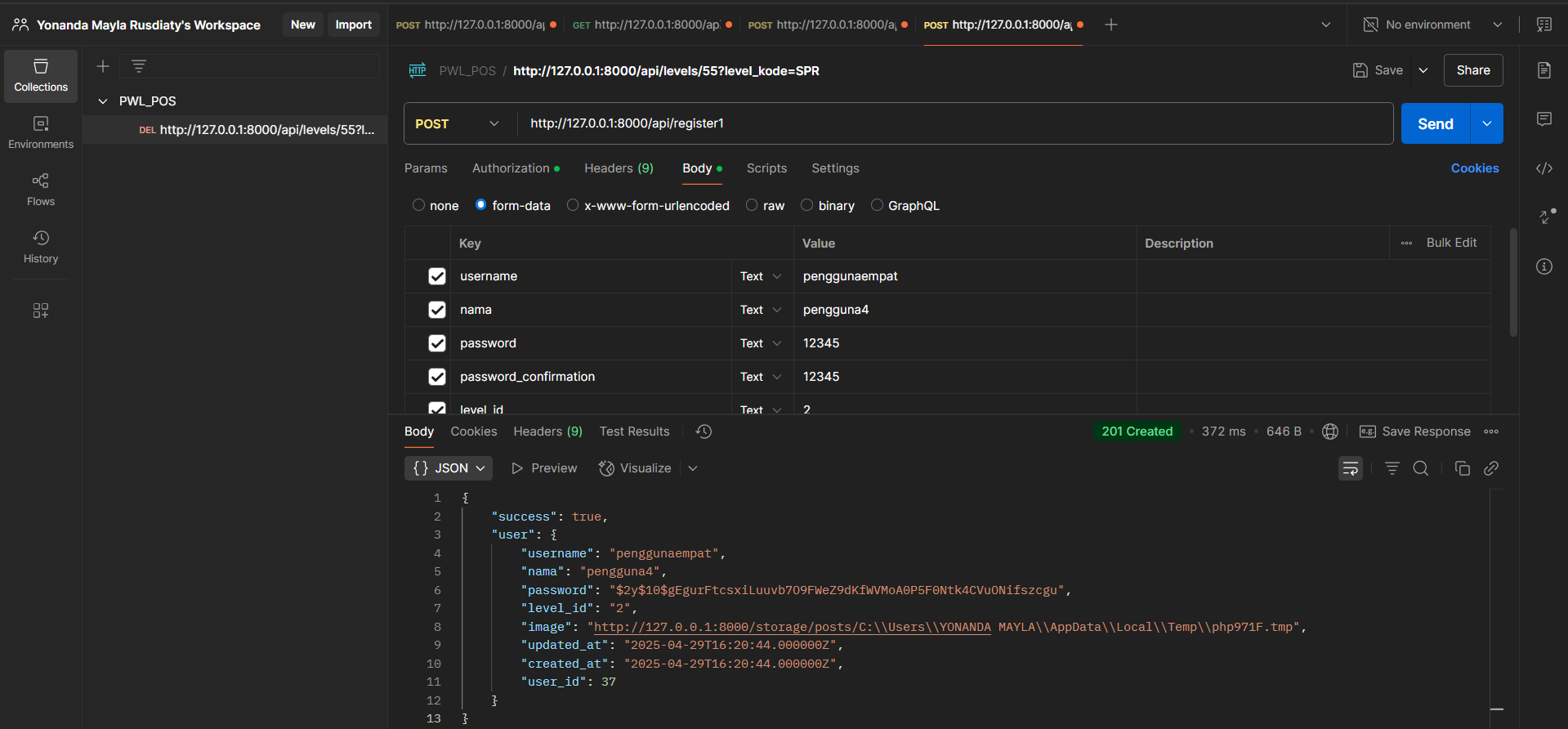


1. Ubah atau tambahkan register1 pada routes/api.php



1. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar <http://127.0.0.1:8000/api/register1> dengan method POST dan klik send

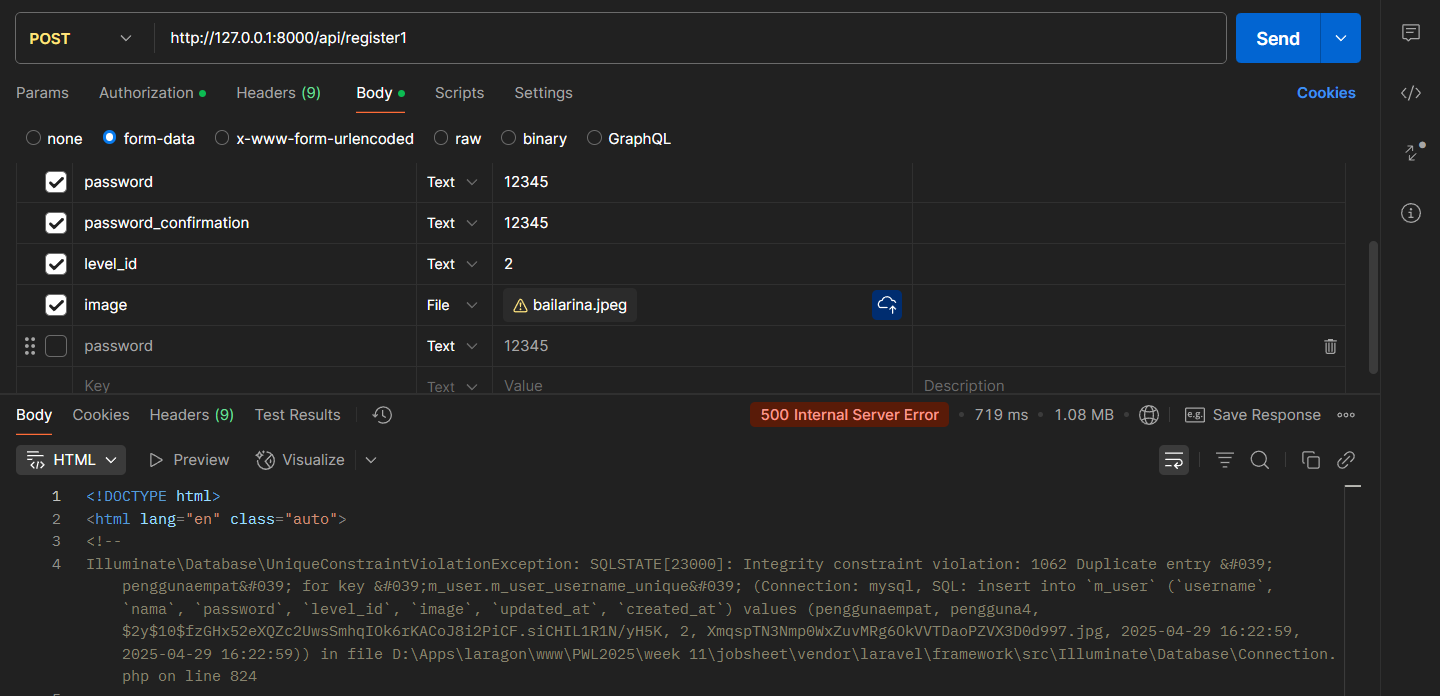




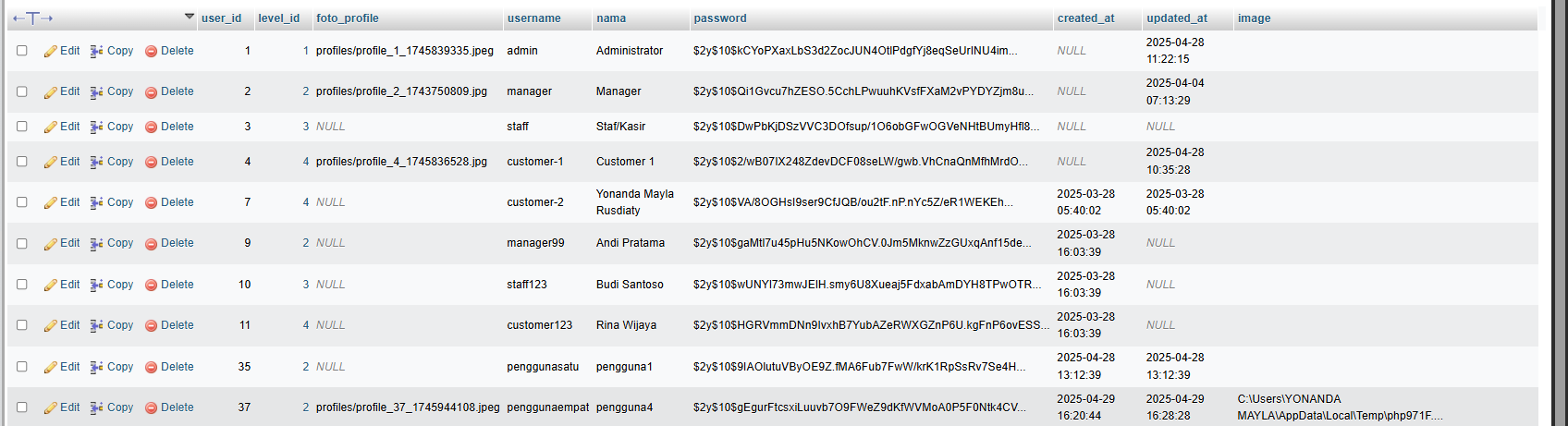
1. Pada Controllers/Api/RegisterController bagian create user ganti dengan

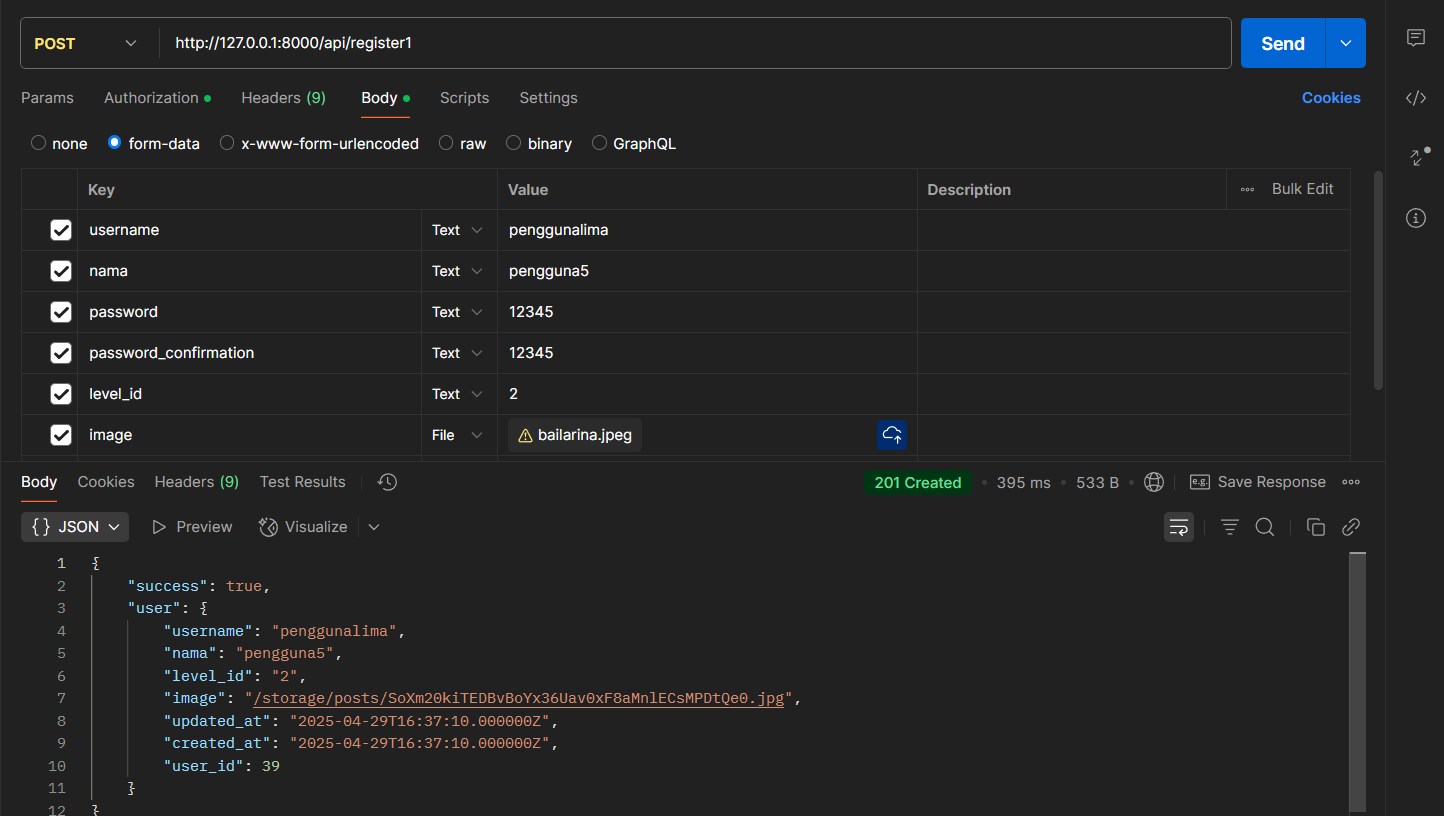


1. Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya



Error tersebut menunjukkan bahwa ada duplikasi data untuk nilai yang seharusnya unik dalam database, yaitu kita mencoba mendaftarkan pengguna dengan username "penggunaempat". Database menolak permintaan ini karena username tersebut sudah terdaftar sebelumnya. Namun fungsi 'image' => $image->hashName(), memiliki fungsi untuk menyimpan nama file gambar yang telah di-hash ke dalam database.







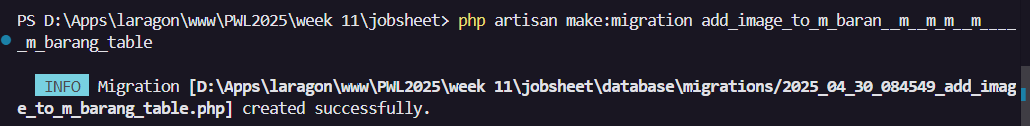
# TUGAS

Implementasikan API untuk upload file/gambar pada tabel lainnya yaitu tabel m\_barang dan gunakan pada transaksi. Uji coba dengan method GET untuk memanggil data yang sudah di inputkan.

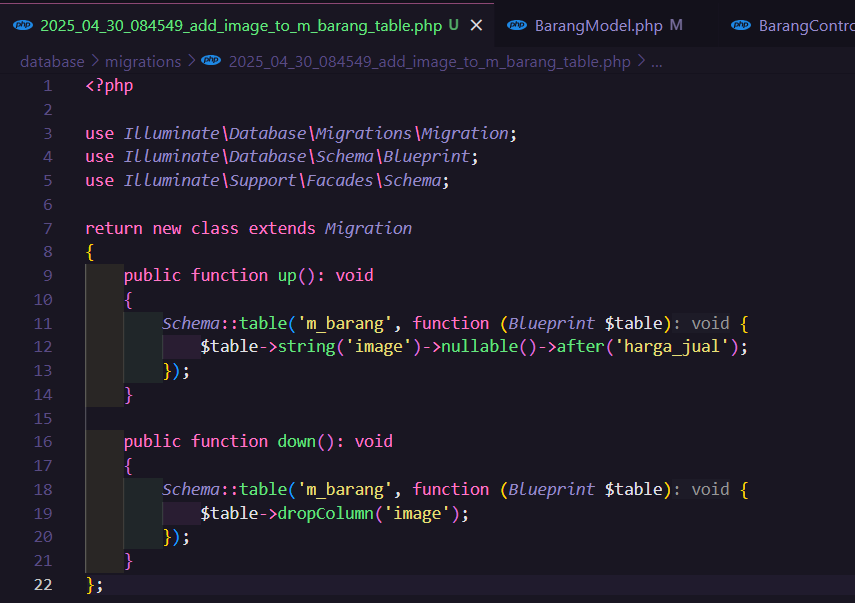
**JAWAB**

* **Tabel m\_barang**

1. Kita akan memodifikasi Table m\_barang dengan menambahkan column : image, buka terminal lalu ketikkan

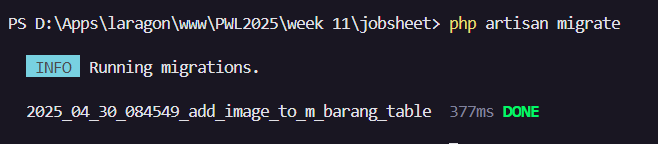


1. Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan:

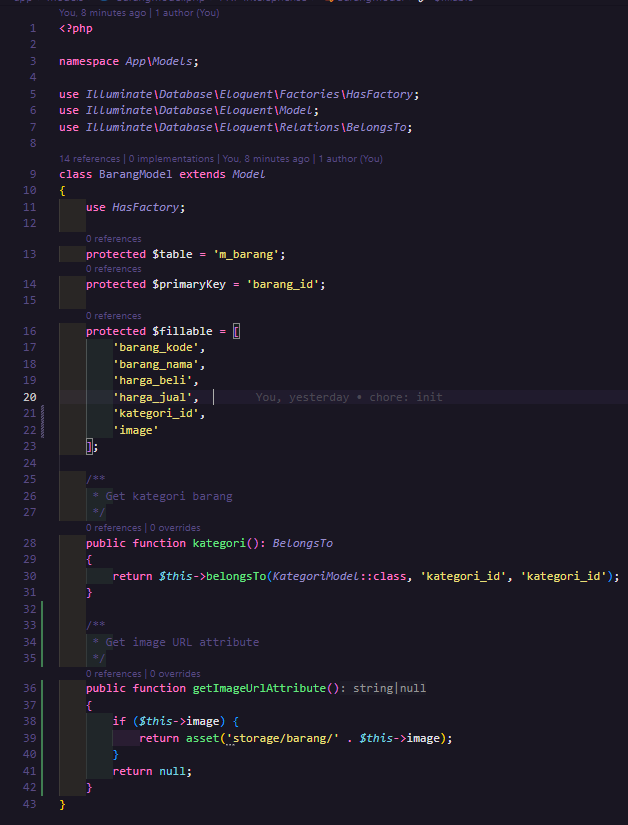


1. Lakukan jalankan update migrasi dengan cara:

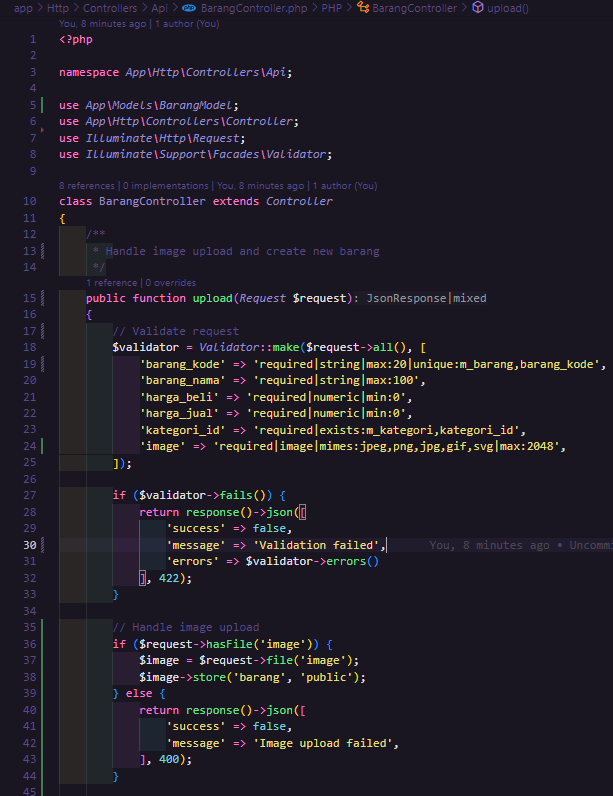
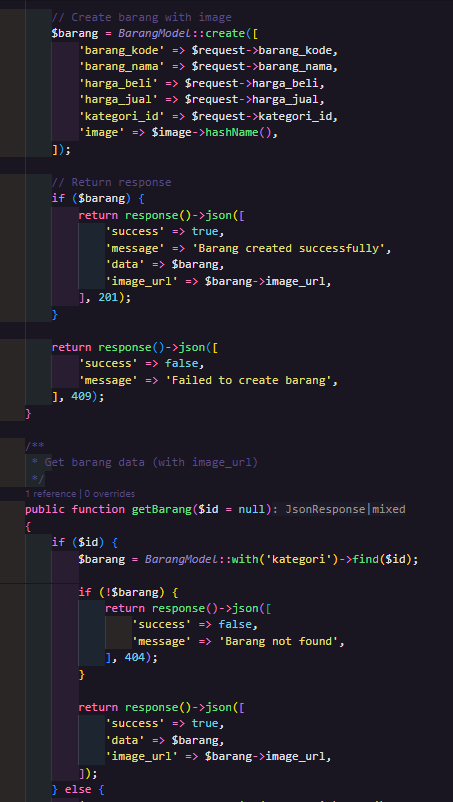
php artisan migrate

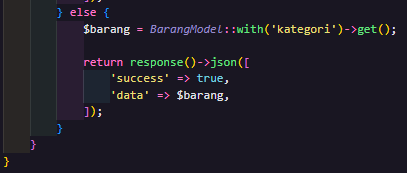


1. Lalu lakukan modifikasi models pada App/Models/BarangModel.php



1. Lakukan modifikasi pada Controllers/Api/BarangController



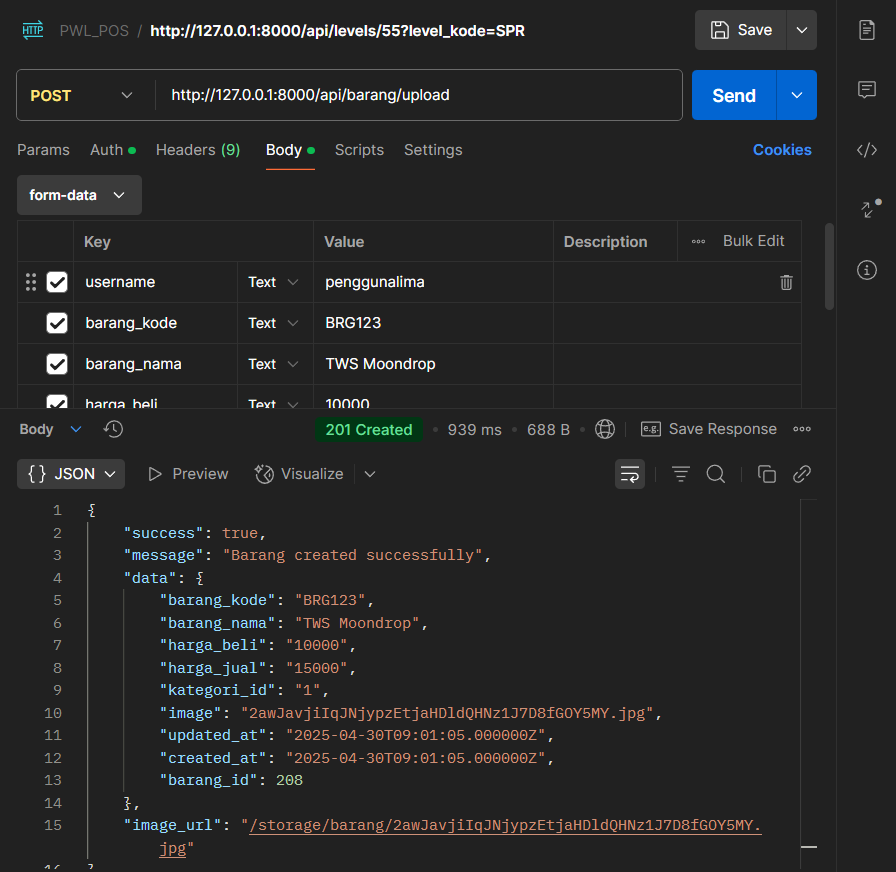
1. Anda dapat menambahkan detail untuk spesifikasi image pada validator

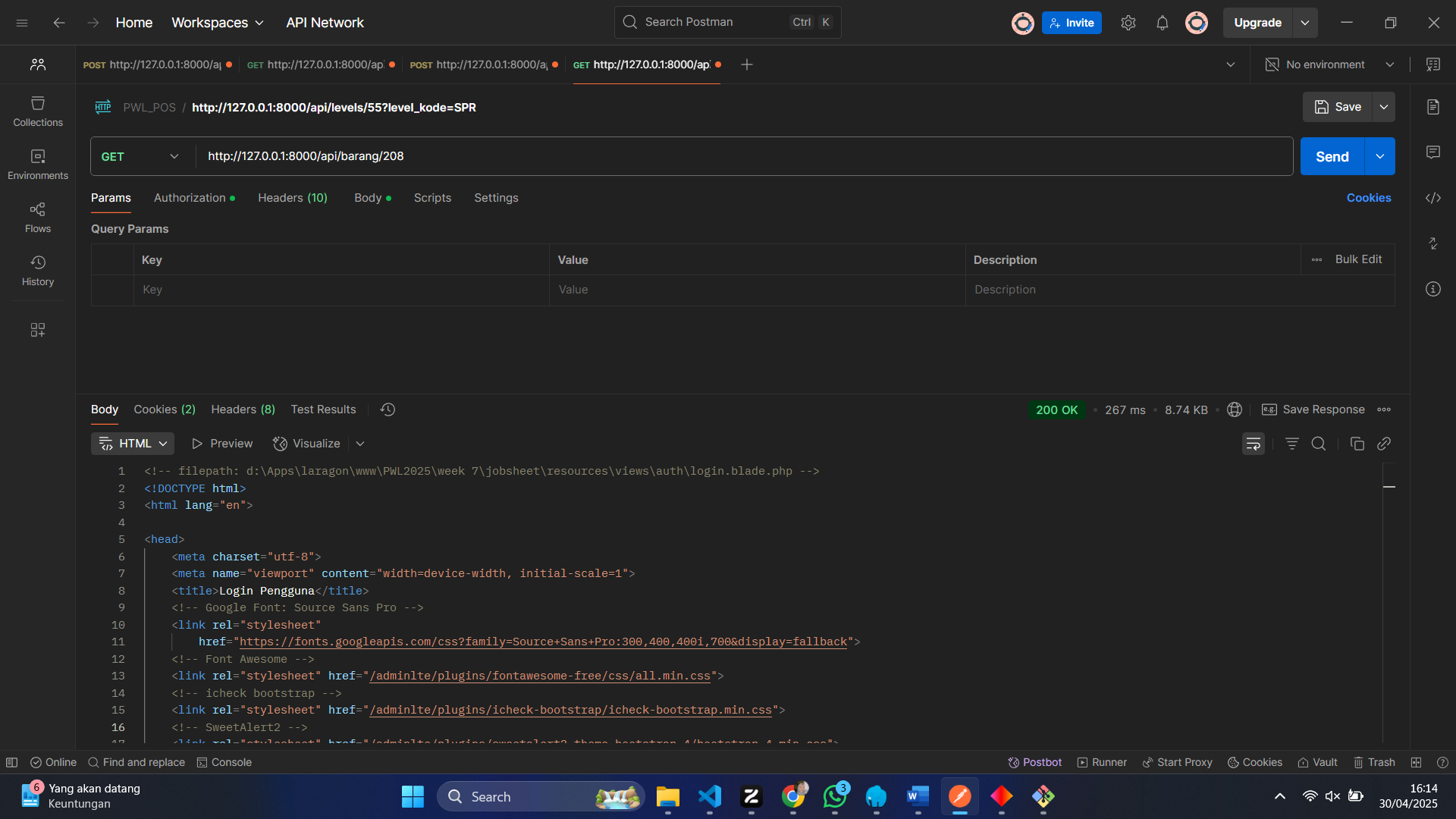


1. Ubah atau tambahkan barang1 pada routes/api.php



1. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar







*\*\*\* Sekian, dan selamat belajar \*\*\**