KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
http://www.polinema.ac.id

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

```php
protected function image(): Attribute
{
    return Attribute::make(
        get: fn ($image) => url('/storage/posts/' . $image),
    );
}
```

Dengan begitu dapat melakukan import Eloquent Attribute dengan

```php
use Illuminate\Database\Eloquent\Casts\Attribute;
```

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

```php
get: fn ($image) => url('/storage/posts/' . $image),
```

Hasil akhir memanggil attribute image

```
domain.com/storage/posts/nama_file_image.png
```

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`

```
PS D:\Apps\laragon\www\PWL2025\week 11\jobsheet> php artisan make:migration add_image_to
_m_user_table

  INFO  Migration [D:\Apps\laragon\www\PWL2025\week 11\jobsheet\database\migrations/202
5_04_29_155540_add_image_to_m_user_table.php] created successfully.
```

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

2025_04_29_155540_add_image_to_m_user_table.php U

database > migrations > 2025_04_29_155540_add_image_to_m_user_table.php > ...

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('m_user', function (Blueprint $table): void {
            $table->string('image');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('m_user', function (Blueprint $table): void {
            $table->dropColumn('image');
        });
    }
};
```

4. Lakukan jalankan update migrasi dengan cara:

```
php artisan migrate
```

```
PS D:\Apps\laragon\www\PWL2025\week 11\jobsheet> php artisan migrate

  INFO  Running migrations.

  2025_04_29_155540_add_image_to_m_user_table ..... 98ms DONE
```

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

```php
namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Database\Eloquent\Casts\Attribute;
use Illuminate\Foundation\Auth\User as Authenticatable;

class UserModel extends Authenticatable implements JWTSubject
{
    public function getJWTIdentifier(): mixed
    {
        return $this->getKey();
    }

    public function getJWTCustomClaims(): array
    {
        return [];
    }

    protected $table = 'm_user';

    protected $primaryKey = 'user_id';

    protected $fillable = [
        'username',
        'name',
        'password',
        'level_id',
        'image'
    ];

    public function level(): BelongsTo
    {
        return $this->belongsTo(LevelModel::class, 'level_id');
    }

    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn ($image): string|UrlGenerator => url('/storage/posts/'. $image),
        );
    }
```

6. Lakukan modifikasi pada Controllers/Api/RegisterController

```php
<?php

namespace App\Http\Controllers\Api;

use App\Models\UserModel;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class RegisterController extends Controller
{
    public function invoke(Request $request): JsonResponse|mixed
    {
        //set validation
        $validation = Validator::make($request->all(), [
            'username' => 'required',
            'nama' => 'required',
            'password' => 'required|min:5|confirmed',
            'level_id' => 'required',
            'image' => 'required'
        ]);

        //if validations fails
        if ($validation->fails()) {
            return response()->json($validation->errors(), 422);
        }

        //create user
        $user = UserModel::create([
            'username' => $request->username,
            'nama' => $request->nama,
            'password' => bcrypt($request->password),
            'level_id' => $request->level_id,
            'image' => $request->image
        ]);

        //return response JSON user is created
        if ($user) {
            return response()->json([
                'success' => true,
                'user' => $user,
            ], 201);
        }

        //return JSON process insert failed
        return response()->json([
            'success' => false,
        ], 409);
    }
}
```

7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

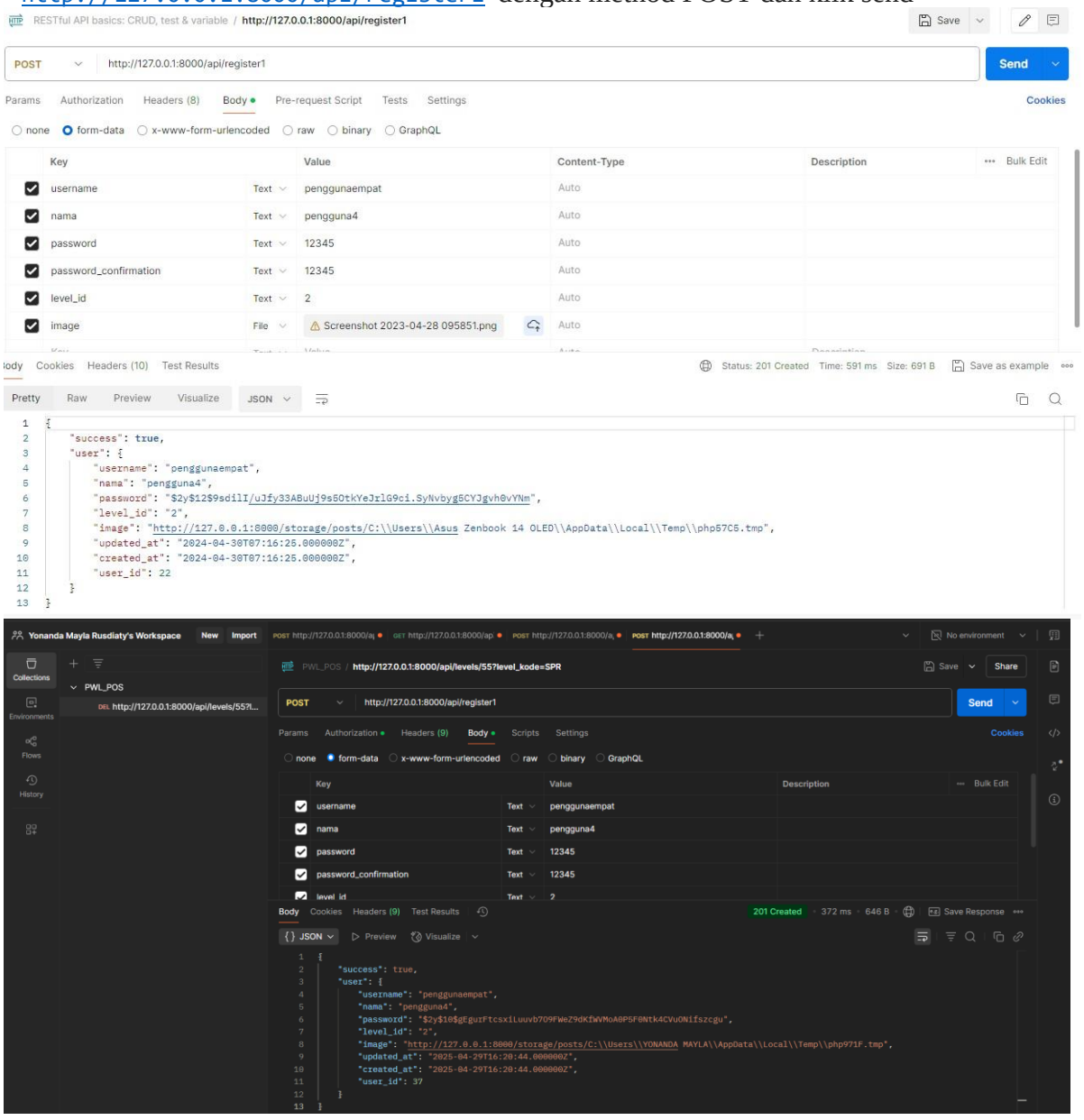
```
'image'      => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

```
'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

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

```
Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('register1');
```

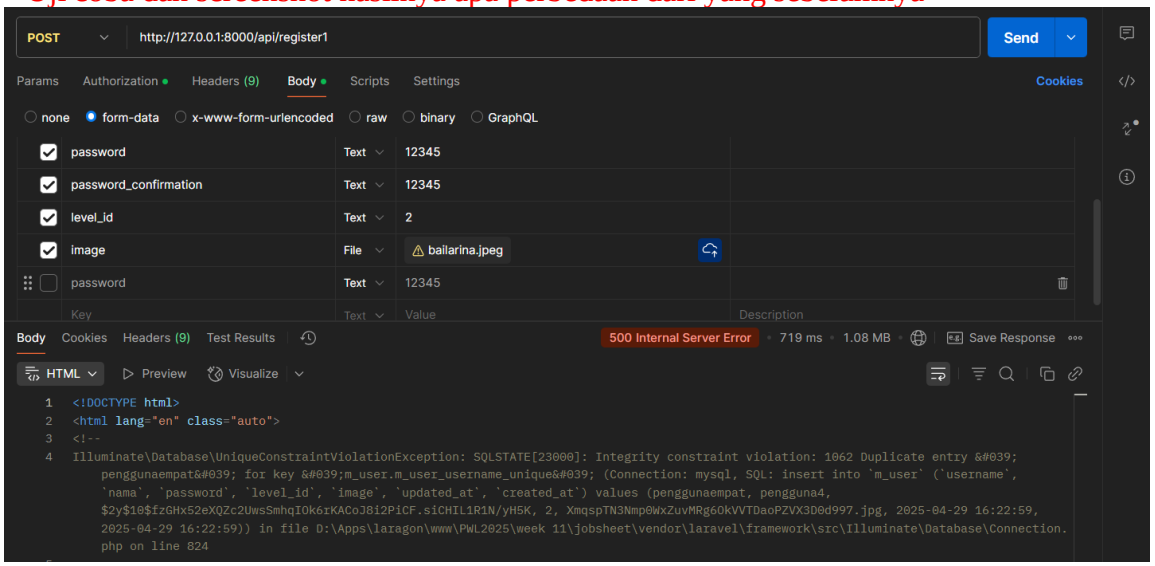
9. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar
   http://127.0.0.1:8000/api/register1 dengan method POST dan klik send



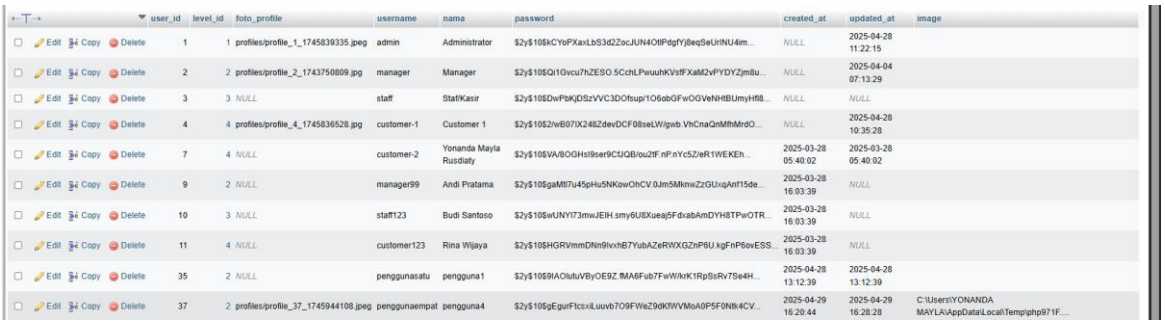
10. Pada Controllers/Api/RegisterController bagian create user ganti dengan

```
'image'     => $image->hashName(),
```

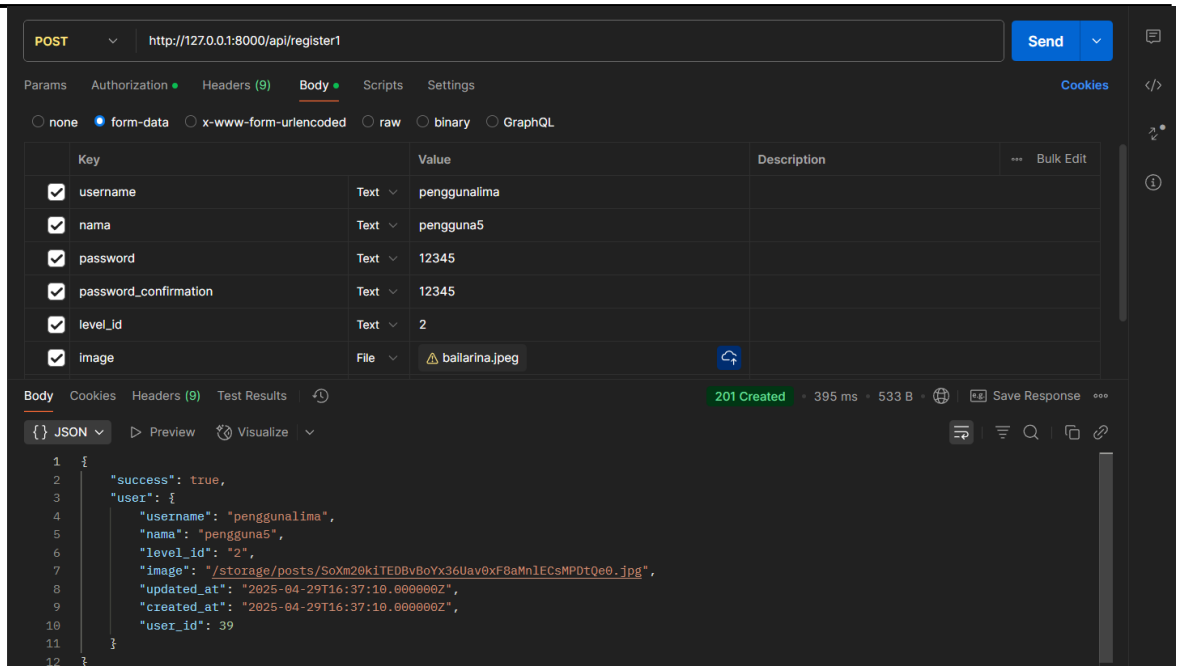
8. Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya



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.



	user_id	level_id	foto_profile	username	nama	password	created_at	updated_at	image
Edit Copy Delete	1	1	profiles/profile_1_1745839335.jpeg	admin	Administrator	$2y$10$kCYoPXaxLbS3d2ZocJUN4OtlPdgfYj8eqSeUrlNU4im...	NULL	2025-04-28 11:22:15	
Edit Copy Delete	2	2	profiles/profile_2_1743750809.jpg	manager	Manager	$2y$10$Qi1Gvcu7hZESO.5CchLPwuuhKVstFXaM2vPYDYZjm8u...	NULL	2025-04-04 07:13:29	
Edit Copy Delete	3	3	NULL	staff	StafiKasir	$2y$10$DwPbKjDSzVVC3DOfsup/1O6obGFwOGVeNHtBUmyHf8...	NULL	NULL	
Edit Copy Delete	4	4	profiles/profile_4_1745836528.jpg	customer-1	Customer 1	$2y$10$2/wB07iX248ZdevDCF08seLW/gwb.VhCnaQnMfhMrdO...	NULL	2025-04-28 10:35:28	
Edit Copy Delete	7	4	NULL	customer-2	Yonanda Mayla Rusdiaty	$2y$10$VA/8OGHsl9ser9CtJQB/ou2tF.nPnYc5Z/eR1WEKEh...	2025-03-28 05:40:02	2025-03-28 05:40:02	
Edit Copy Delete	9	2	NULL	manager99	Andi Pratama	$2y$10$gaMtl7u45pHu5NKowOhCV.0Jm5MknwZzGUxqAnf15de...	2025-03-28 16:03:39	NULL	
Edit Copy Delete	10	3	NULL	staff123	Budi Santoso	$2y$10$wUNYI73mwJEIH.smy6U8Xueaj5FdxabAmDYH8TPwOTR...	2025-03-28 16:03:39	NULL	
Edit Copy Delete	11	4	NULL	customer123	Rina Wijaya	$2y$10$HGRVmmDNn9IvxhB7YubAZeRWXGZnP6U.kgFnP5ovESS...	2025-03-28 16:03:39	NULL	
Edit Copy Delete	35	2	NULL	penggunasatu	pengguna1	$2y$10$9IAOlutuVByOE9Z.fMA6Fub7FwW/krK1RpSsRv7Se4H...	2025-04-28 13:12:39	2025-04-28 13:12:39	
Edit Copy Delete	37	2	profiles/profile_37_1745944108.jpeg	penggunaempat	pengguna4	$2y$10$gEgurFlcsxiLuuvb7O9FWeZ9dKfWVMoA0P5F0N8k4CV...	2025-04-29 16:20:44	2025-04-29 16:28:28	C:\Users\YONANDA MAYLA\AppData\Local\Temp\php971F....



TUGAS

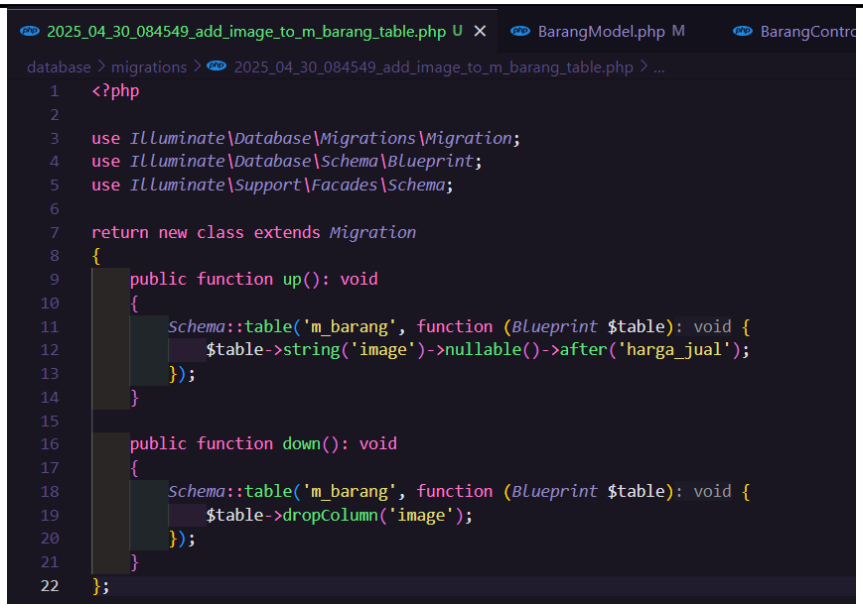
Implementasikan API untuk upload file/gambar pada tabel lainnya yaitu tabel m_barang dan gunakan pada transaksi. Uji coba dengan method GET untuk memanggil data yang sudah di inputkan.

JAWAB

- **Tabel m_barang**
  a. Kita akan memodifikasi Table m_barang dengan menambahkan column : image, buka terminal lalu ketikkan

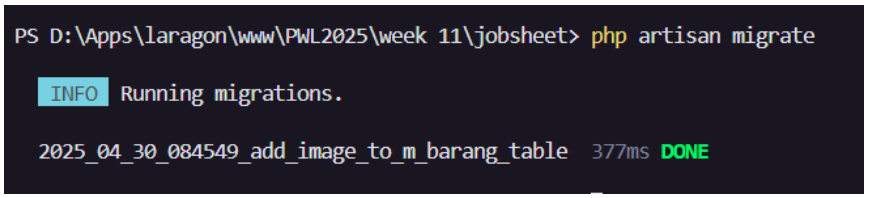
```
PS D:\Apps\laragon\www\PWL2025\week 11\jobsheet> php artisan make:migration add_image_to_m_baran__m__m_m__m____
_m_barang_table

  INFO  Migration [D:\Apps\laragon\www\PWL2025\week 11\jobsheet\database\migrations/2025_04_30_084549_add_imag
e_to_m_barang_table.php] created successfully.
```

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



c. Lakukan jalankan update migrasi dengan cara:

php artisan migrate



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

```php
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;

class BarangModel extends Model
{
    use HasFactory;

    protected $table = 'm_barang';
    protected $primaryKey = 'barang_id';

    protected $fillable = [
        'barang_kode',
        'barang_nama',
        'harga_beli',
        'harga_jual',
        'kategori_id',
        'image'
    ];

    /**
     * Get kategori barang
     */
    public function kategori(): BelongsTo
    {
        return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
    }

    /**
     * Get image URL attribute
     */
    public function getImageUrlAttribute(): string|null
    {
        if ($this->image) {
            return asset('storage/barang/' . $this->image);
        }
        return null;
    }
}
```

e. Lakukan modifikasi pada Controllers/Api/BarangController

app > Http > Controllers > Api > BarangController.php > PHP > BarangController > upload()

You, 8 minutes ago | 1 author (You)

```php
<?php

namespace App\Http\Controllers\Api;

use App\Models\BarangModel;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

8 references | 0 implementations | You, 8 minutes ago | 1 author (You)
class BarangController extends Controller
{
    /**
     * Handle image upload and create new barang
     */
    1 reference | 0 overrides
    public function upload(Request $request): JsonResponse|mixed
    {
        // Validate request
        $validator = Validator::make($request->all(), [
            'barang_kode' => 'required|string|max:20|unique:m_barang,barang_kode',
            'barang_nama' => 'required|string|max:100',
            'harga_beli' => 'required|numeric|min:0',
            'harga_jual' => 'required|numeric|min:0',
            'kategori_id' => 'required|exists:m_kategori,kategori_id',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);

        if ($validator->fails()) {
            return response()->json([
                'success' => false,
                'message' => 'Validation failed',      You, 8 minutes ago • Uncomm
                'errors' => $validator->errors()
            ], 422);
        }

        // Handle image upload
        if ($request->hasFile('image')) {
            $image = $request->file('image');
            $image->store('barang', 'public');
        } else {
            return response()->json([
                'success' => false,
                'message' => 'Image upload failed',
            ], 400);
        }
```

```php
    // Create barang with image
    $barang = BarangModel::create([
        'barang_kode' => $request->barang_kode,
        'barang_nama' => $request->barang_nama,
        'harga_beli' => $request->harga_beli,
        'harga_jual' => $request->harga_jual,
        'kategori_id' => $request->kategori_id,
        'image' => $image->hashName(),
    ]);

    // Return response
    if ($barang) {
        return response()->json([
            'success' => true,
            'message' => 'Barang created successfully',
            'data' => $barang,
            'image_url' => $barang->image_url,
        ], 201);
    }

    return response()->json([
        'success' => false,
        'message' => 'Failed to create barang',
    ], 409);
}

/**
 * Get barang data (with image_url)
 */
1 reference | 0 overrides
public function getBarang($id = null): JsonResponse|mixed
{
    if ($id) {
        $barang = BarangModel::with('kategori')->find($id);

        if (!$barang) {
            return response()->json([
                'success' => false,
                'message' => 'Barang not found',
            ], 404);
        }

        return response()->json([
            'success' => true,
            'data' => $barang,
            'image_url' => $barang->image_url,
        ]);
    } else {
        } else {
            $barang = BarangModel::with('kategori')->get();

            return response()->json([
                'success' => true,
                'data' => $barang,
            ]);
        }
    }
}
```

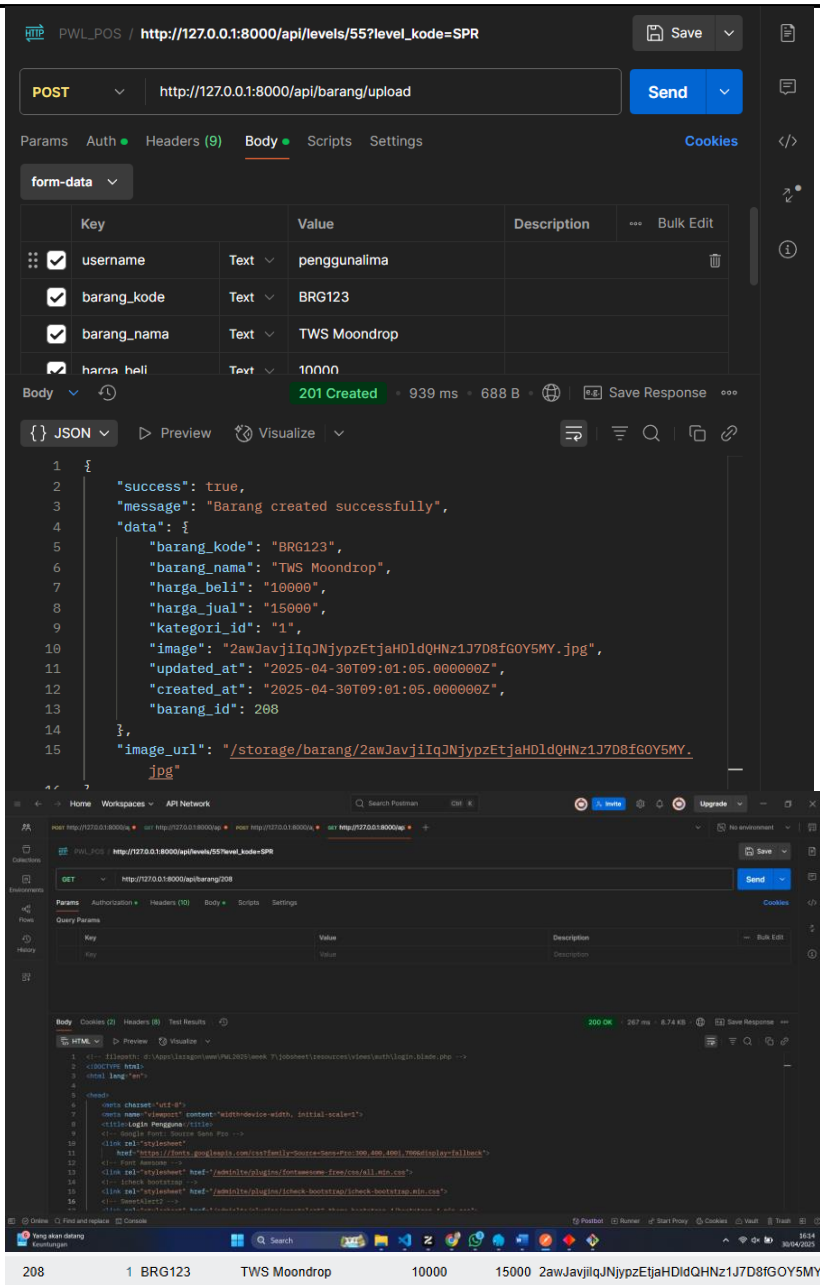
f. Anda dapat menambahkan detail untuk spesifikasi image pada validator

```php
'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

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

```php
Route::post('/barang/upload', [App\Http\Controllers\Api\BarangController::class, 'upload']);
Route::get('/barang/{id?}', [App\Http\Controllers\Api\BarangController::class, 'getBarang']);
```

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



208	1	BRG123	TWS Moondrop	10000	15000	2awJavjiIqJNjypzEtjaHDldQHNz1J7D8fGOY5MY.jpg	2025-04-30 09:01:05	2025-04-30 09:01:05

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