Nama	**: Dini Elminingtyas**
Kelas/No.absen	**: SIB 2B / 07**
NIM	**: 2341760180**

PEMROGRAMAN WEB LANJUT

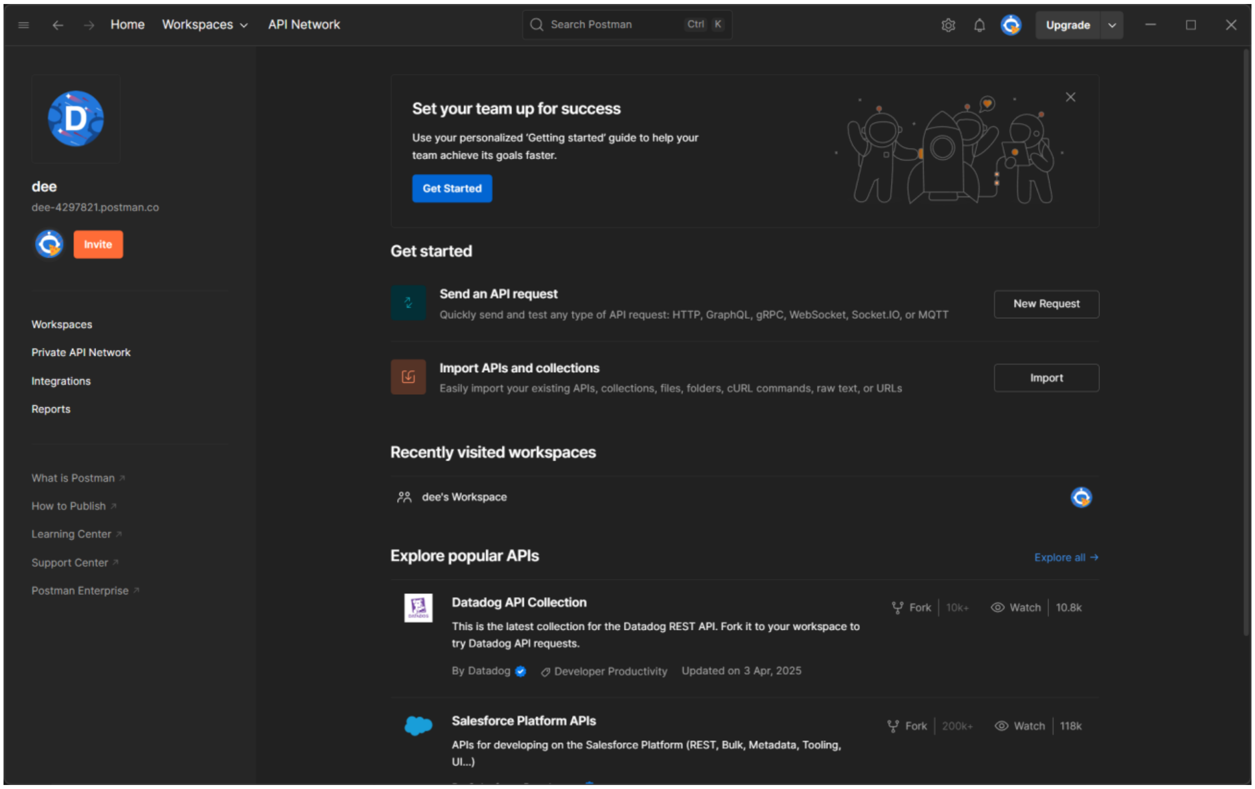
JOBSHEET

RESTFUL API 2

Praktikum 1 – Implementasi Eloquent Accessor

1. Sebelum memulai pastikan REST API, terlebih dahulu pastikan sudah ter instal aplikasi Postman.

⇨ Postman



Set your team up for success

Use your personalized 'Getting started' guide to help your team achieve its goals faster.

Get started

Send an API request

Quickly send and test any type of API request: HTTP, GraphQL, gRPC, WebSocket, Socket.IO, or MQTT

Import APIs and collections

Easily import your existing APIs, collections, files, folders, cURL commands, raw text, or URLs

Recently visited workspaces

dee's Workspace

Explore popular APIs

Datadog API Collection

This is the latest collection for the Datadog REST API. Fork it to your workspace to try Datadog API requests.

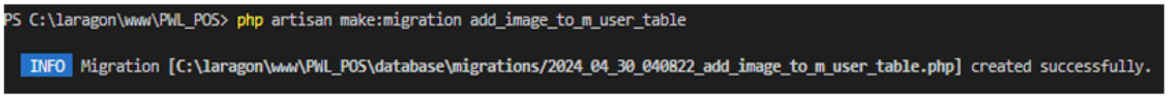
By Datadog · Developer Productivity · Updated on 3 Apr, 2025

Salesforce Platform APIs

APIs for developing on the Salesforce Platform (REST, Bulk, Metadata, Tooling, UI...)

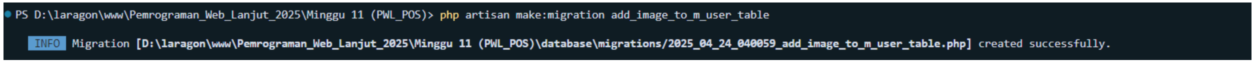
2. Kita akan memodifikasi Table m_user dengan menambahkan column : image, buka terminal lalu ketikkan

php artisan make:migration add_image_to_m_user_table



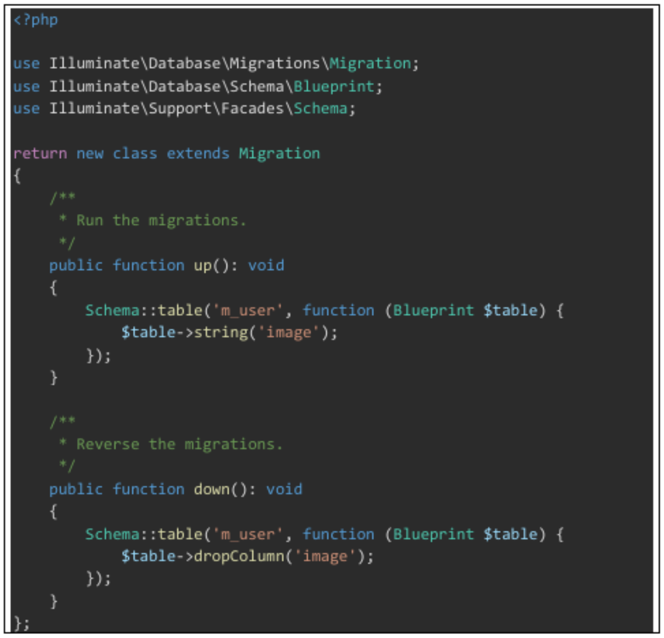
```
PS C:\laragon\www\PWL_POS> php artisan make:migration add_image_to_m_user_table

  INFO  Migration [C:\laragon\www\PWL_POS\database\migrations/2024_04_30_040822_add_image_to_m_user_table.php] created successfully.
```

⇨ Add migration



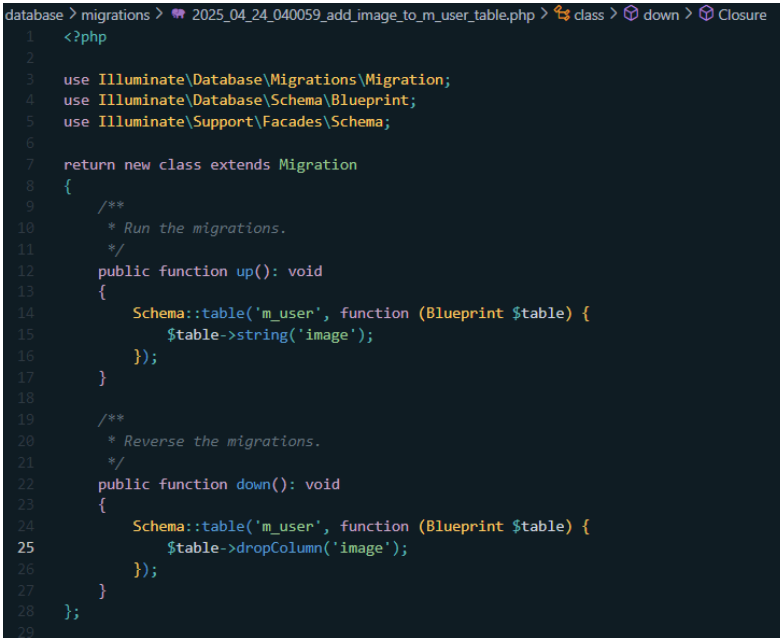
```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 11 (PWL_POS)> php artisan make:migration add_image_to_m_user_table

  INFO  Migration [D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 11 (PWL_POS)\database\migrations/2025_04_24_040059_add_image_to_m_user_table.php] created successfully.
```

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



```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) {
            $table->string('image');
        });
    }

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

⇨ Migration



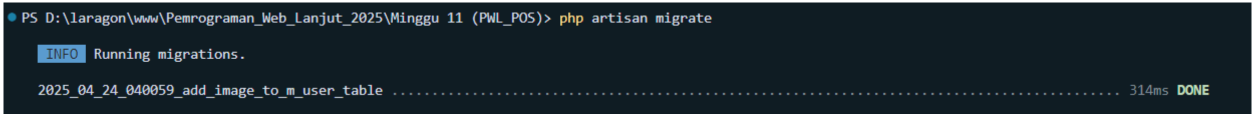
database > migrations > 2025_04_24_040059_add_image_to_m_user_table.php > class > down > Closure

```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) {
            $table->string('image');
        });
    }

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

4. Lakukan jalankan update migrasi dengan cara:

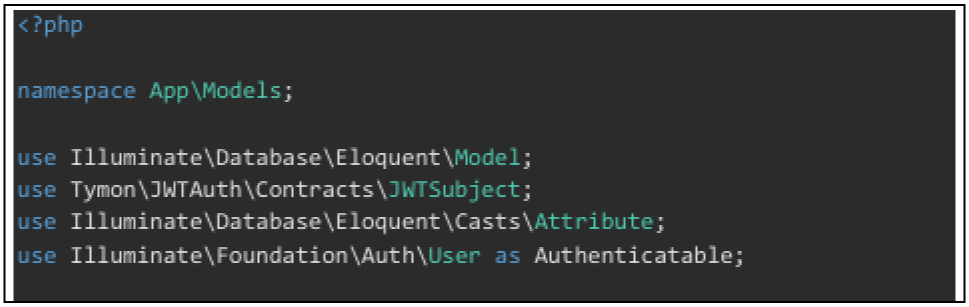
```
php artisan migrate
```

⇨ Migrate



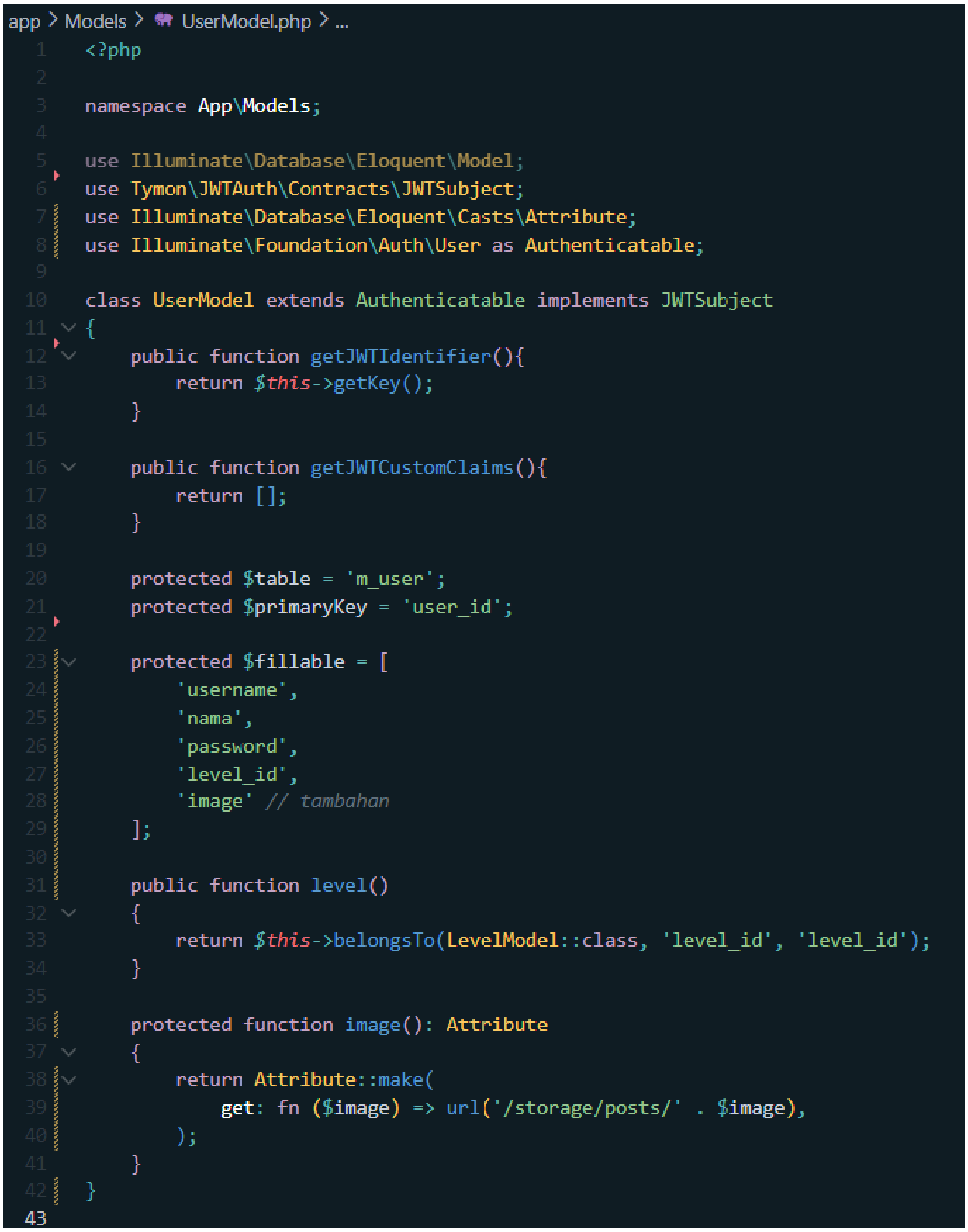
```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 11 (PWL_POS)> php artisan migrate

   INFO  Running migrations.

   2025_04_24_040059_add_image_to_m_user_table ........................................................................................ 314ms DONE
```

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



```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;
```

⇨ UserModel

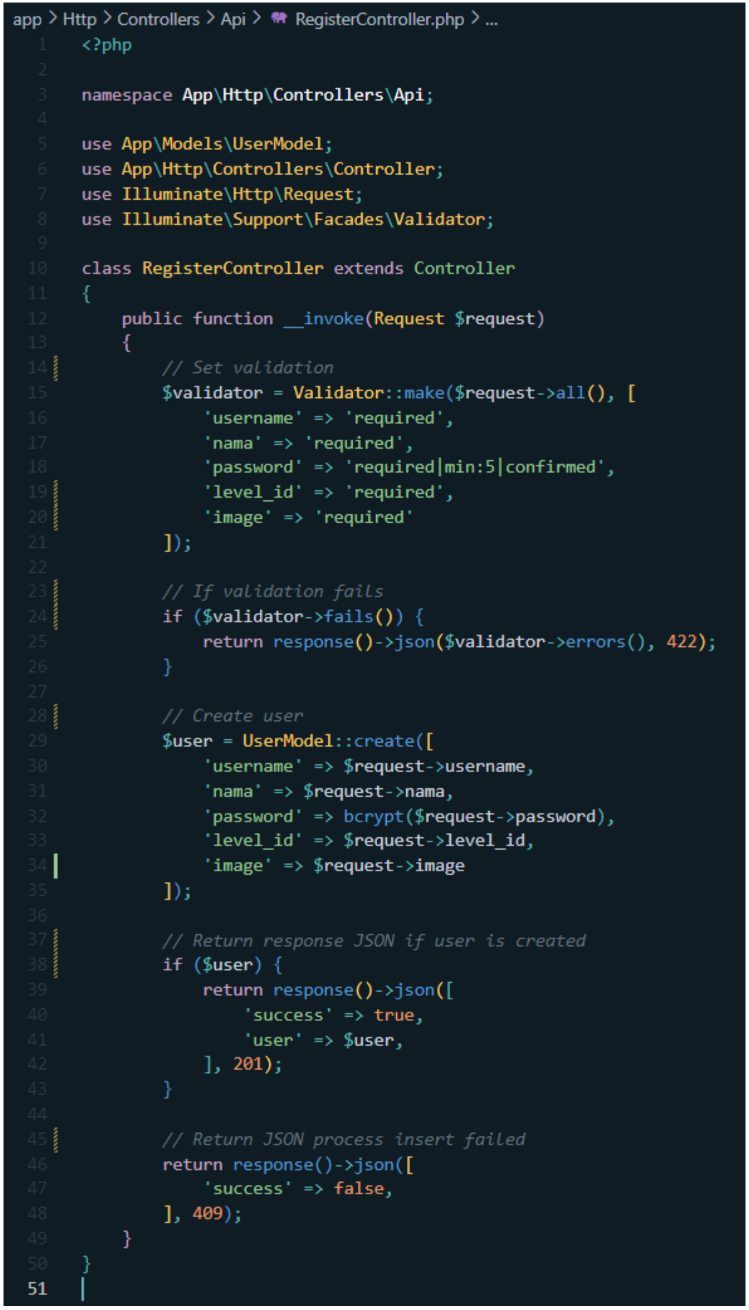


```php
<?php

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(){
        return $this->getKey();
    }

    public function getJWTCustomClaims(){
        return [];
    }

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';

    protected $fillable = [
        'username',
        'nama',
        'password',
        'level_id',
        'image' // tambahan
    ];

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

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

6. Lakukan modifikasi pada Controllers/Api/RegisterController

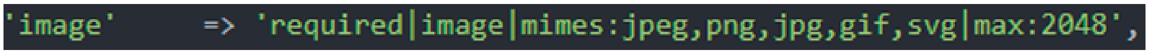
```php
<?php

namespace App\Http\Controllers\Api;

use App\Models\UserModel;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
```

⇨ RegisterController


app > Http > Controllers > Api > RegisterController.php > ...

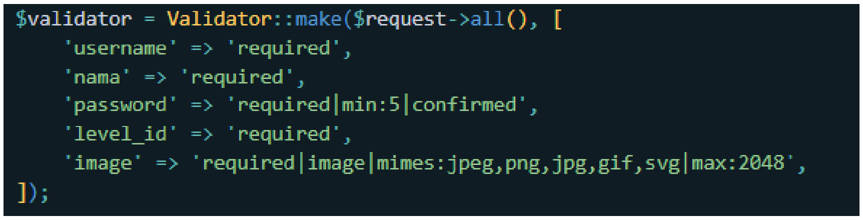
```php
<?php

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)
    {
        // Set validation
        $validator = Validator::make($request->all(), [
            'username' => 'required',
            'nama' => 'required',
            'password' => 'required|min:5|confirmed',
            'level_id' => 'required',
            'image' => 'required'
        ]);

        // If validation fails
        if ($validator->fails()) {
            return response()->json($validator->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 if 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



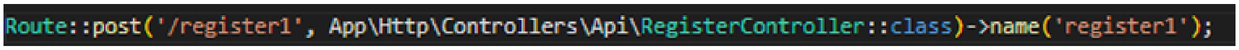
```
'image'      =>  'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

⇨ Validator



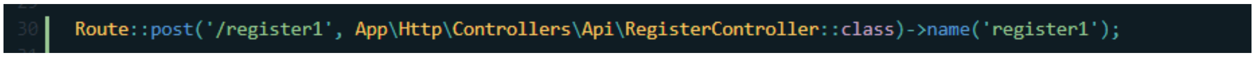
```
$validator = Validator::make($request->all(), [
    'username' => 'required',
    'nama' => 'required',
    'password' => 'required|min:5|confirmed',
    'level_id' => 'required',
    'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
]);
```

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



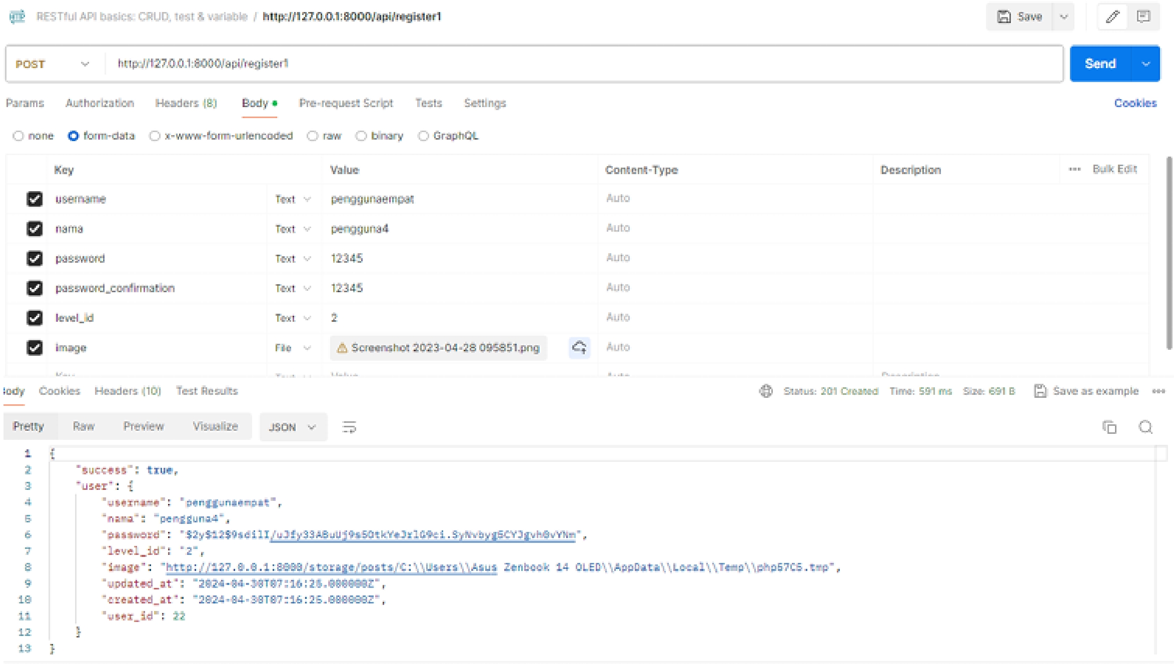
```
Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('register1');
```

⇨ Routes



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

8. 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

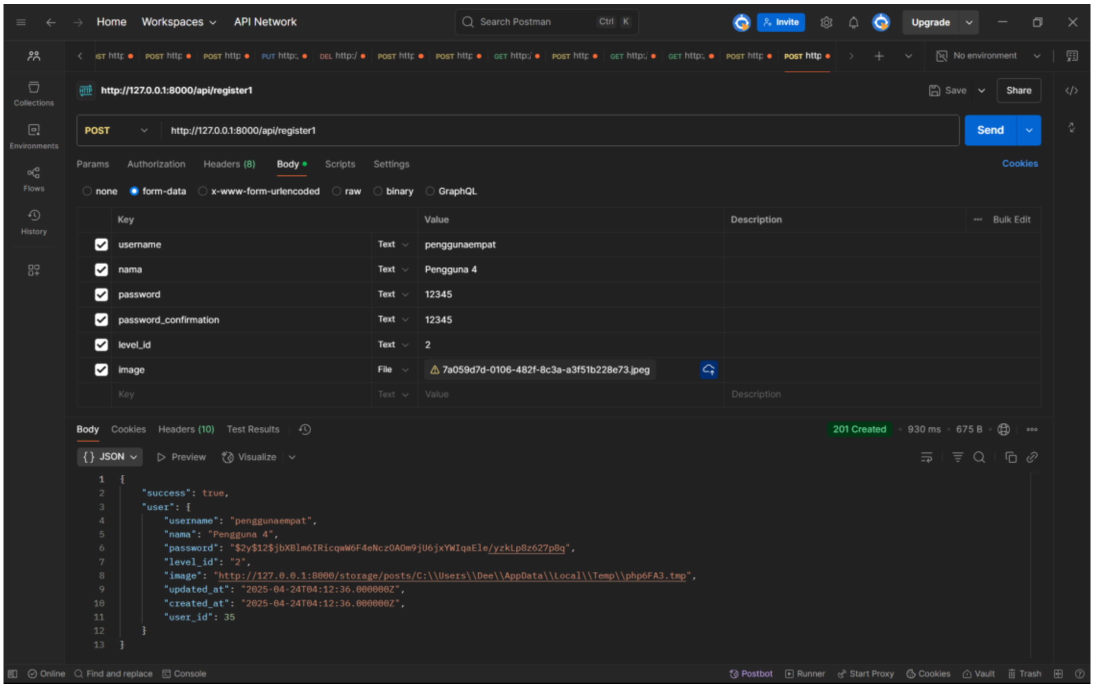


RESTful API basics: CRUD, test & variable / http://127.0.0.1:8000/api/register1
Save
POST
http://127.0.0.1:8000/api/register1
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Content-Type	Description
☑	username	Text	penggunaempat	Auto	
☑	nama	Text	pengguna4	Auto	
☑	password	Text	12345	Auto	
☑	password_confirmation	Text	12345	Auto	
☑	level_id	Text	2	Auto	
☑	image	File	Screenshot 2023-04-28 095851.png	Auto	

```
{
    "success": true,
    "user": {
        "username": "penggunaempat",
        "nama": "pengguna4",
        "password": "$2y$12$9sdilI/uJfy33ABuUj9s5OtkYeJrlG9ci.SyNvbygSCYJgvh0vYNm",
        "level_id": "2",
        "image": "http://127.0.0.1:8000/storage/posts/C:\\Users\\Asus Zenbook 14 OLED\\AppData\\Local\\Temp\\php57C5.tmp",
        "updated_at": "2024-04-30T07:16:25.000000Z",
        "created_at": "2024-04-30T07:16:25.000000Z",
        "user_id": 22
    }
}
```

⇨ Postman



POST http://127.0.0.1:8000/api/register1

Body: ○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Description
☑	username	Text	penggunaempat	
☑	nama	Text	Pengguna 4	
☑	password	Text	12345	
☑	password_confirmation	Text	12345	
☑	level_id	Text	2	
☑	image	File	7a059d7d-0106-482f-8c3a-a3f51b228e73.jpeg	

201 Created · 930 ms · 675 B

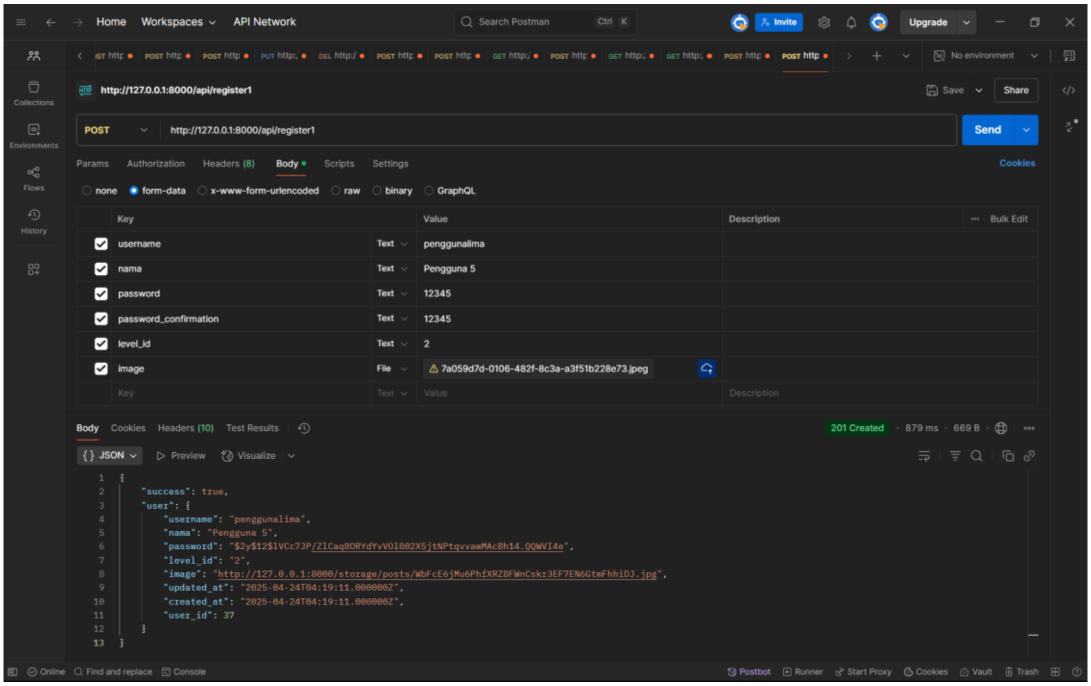
```
{
    "success": true,
    "user": {
        "username": "penggunaempat",
        "nama": "Pengguna 4",
        "password": "$2y$12$jbXBlm6IRicqwW6F4eNcz0A0m9jU6jxYWIqaEle/yzkLp8z627p8q",
        "level_id": "2",
        "image": "http://127.0.0.1:8000/storage/posts/C:\\Users\\Dee\\AppData\\Local\\Temp\\php6FA3.tmp",
        "updated_at": "2025-04-24T04:12:36.000000Z",
        "created_at": "2025-04-24T04:12:36.000000Z",
        "user_id": 35
    }
}
```

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

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

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

⇨ Create user



POST http://127.0.0.1:8000/api/register1

Body: ○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key		Value	Description
☑	username	Text	penggunalima	
☑	nama	Text	Pengguna 5	
☑	password	Text	12345	
☑	password_confirmation	Text	12345	
☑	level_id	Text	2	
☑	image	File	7a059d7d-0106-482f-8c3a-a3f51b228e73.jpeg	

201 Created · 879 ms · 669 B

```
{
    "success": true,
    "user": {
        "username": "penggunalima",
        "nama": "Pengguna 5",
        "password": "$2y$12$lVCc7JP/ZlCaq8ORYdYvVOl802X5jtNPtqvvawMAcBh14.QQWVI4e",
        "level_id": "2",
        "image": "http://127.0.0.1:8000/storage/posts/WbFcE6jMu6PhfXRZ8FWnCskr3EF7EN6GtmFhhiDJ.jpg",
        "updated_at": "2025-04-24T04:19:11.000000Z",
        "created_at": "2025-04-24T04:19:11.000000Z",
        "user_id": 37
    }
}
```

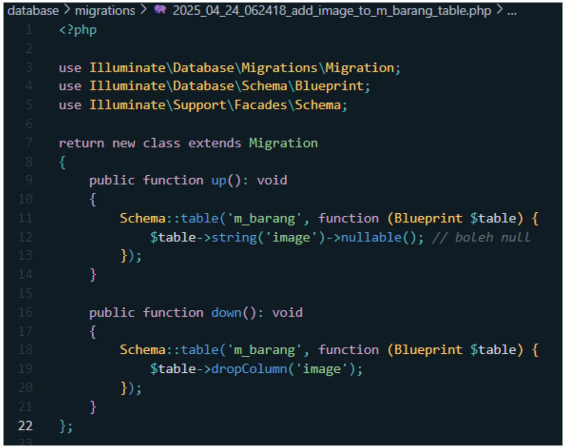
Perbedaannya terletak pada path gambar. Pada percobaan sebelumnya, path image terlihat secara jelas sesuai dengan nama file aslinya. Sedangkan pada percobaan yang sekarang, path-nya disamarkan menggunakan nama acak karena sudah di-*hash*.

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.

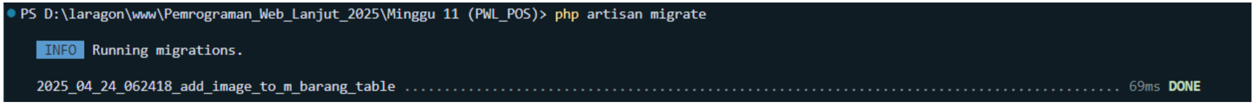
⇨ Barang

- Upload image m_barang migrations



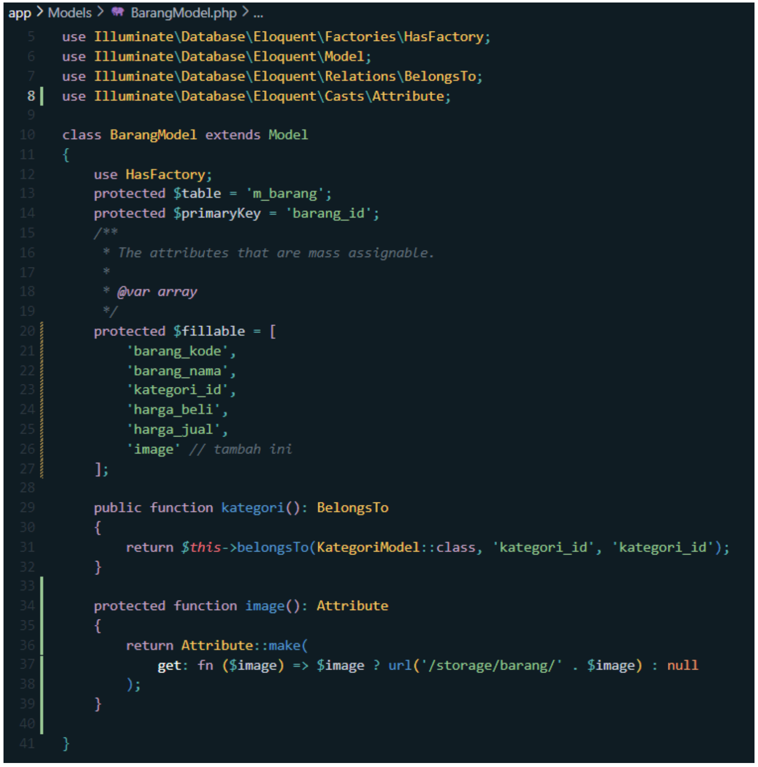
database > migrations > 2025_04_24_062418_add_image_to_m_barang_table.php > ...

```php
<?php

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

return new class extends Migration
{
    public function up(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->string('image')->nullable(); // boleh null
        });
    }

    public function down(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->dropColumn('image');
        });
    }
};
```



```
PS D:\laragon\www\Pemrograman_Web_Lanjut_2025\Minggu 11 (PWL_POS)> php artisan migrate

   INFO  Running migrations.

   2025_04_24_062418_add_image_to_m_barang_table ............................................................................ 69ms DONE
```

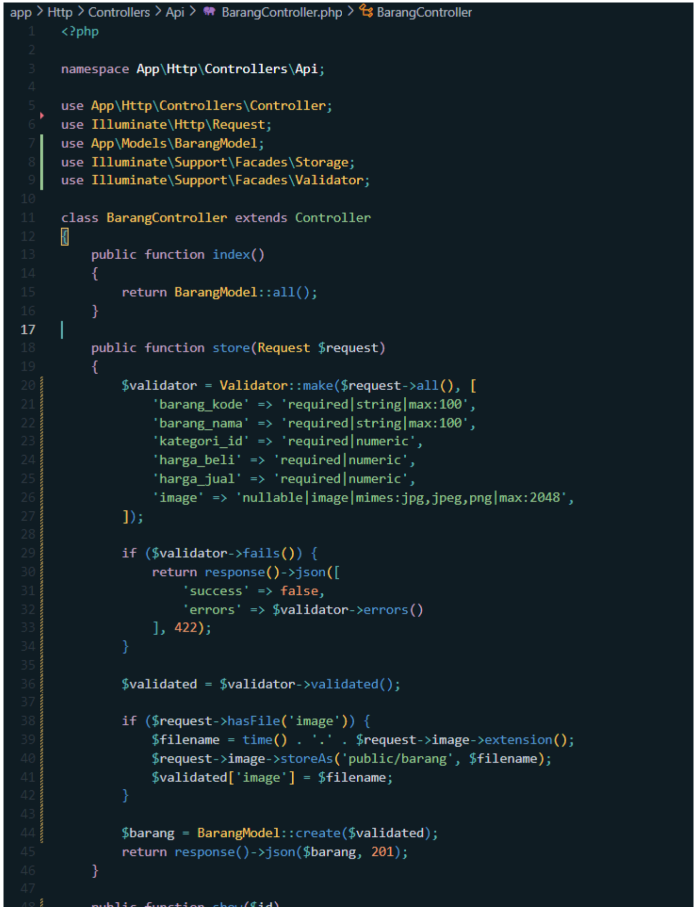
- BarangModel



app > Models > BarangModel.php > ...

```php
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Casts\Attribute;

class BarangModel extends Model
{
    use HasFactory;
    protected $table = 'm_barang';
    protected $primaryKey = 'barang_id';
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'barang_kode',
        'barang_nama',
        'kategori_id',
        'harga_beli',
        'harga_jual',
        'image' // tambah ini
    ];

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

    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn ($image) => $image ? url('/storage/barang/' . $image) : null
        );
    }

}
```

- Barang Controller

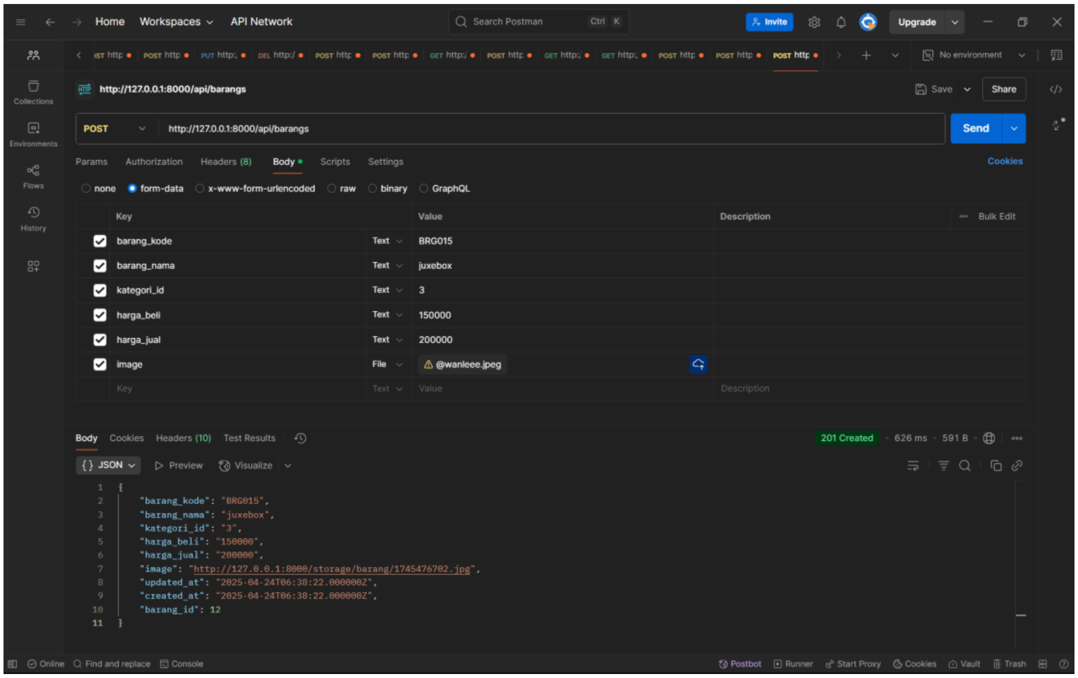


app > Http > Controllers > Api > BarangController.php > BarangController

```php
<?php

namespace App\Http\Controllers\Api;

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

class BarangController extends Controller
{
    public function index()
    {
        return BarangModel::all();
    }

    public function store(Request $request)
    {
        $validator = Validator::make($request->all(), [
            'barang_kode' => 'required|string|max:100',
            'barang_nama' => 'required|string|max:100',
            'kategori_id' => 'required|numeric',
            'harga_beli' => 'required|numeric',
            'harga_jual' => 'required|numeric',
            'image' => 'nullable|image|mimes:jpg,jpeg,png|max:2048',
        ]);

        if ($validator->fails()) {
            return response()->json([
                'success' => false,
                'errors' => $validator->errors()
            ], 422);
        }

        $validated = $validator->validated();

        if ($request->hasFile('image')) {
            $filename = time() . '.' . $request->image->extension();
            $request->image->storeAs('public/barang', $filename);
            $validated['image'] = $filename;
        }

        $barang = BarangModel::create($validated);
        return response()->json($barang, 201);
    }

    public function show($id)
```

- Postman

Post



http://127.0.0.1:8000/api/barangs

POST http://127.0.0.1:8000/api/barangs

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

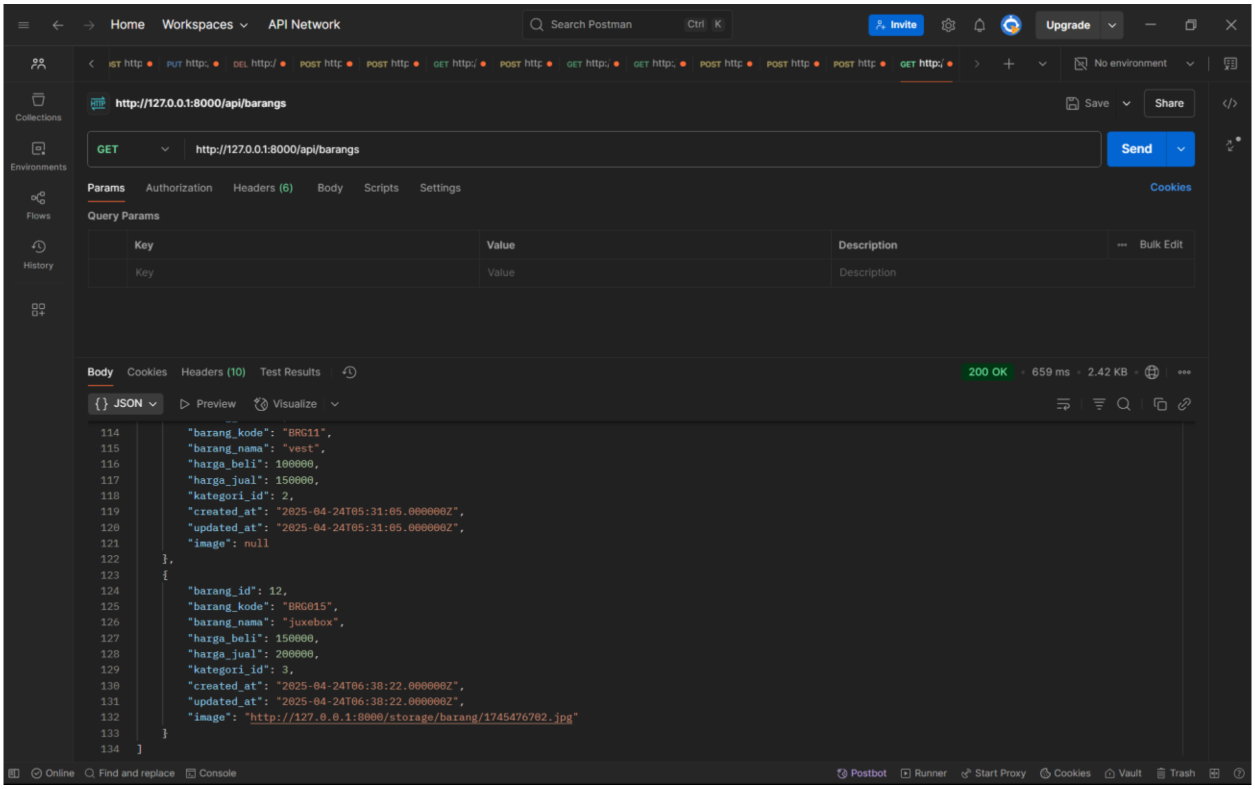
	Key		Value	Description
☑	barang_kode	Text	BRG015	
☑	barang_nama	Text	juxebox	
☑	kategori_id	Text	3	
☑	harga_beli	Text	150000	
☑	harga_jual	Text	200000	
☑	image	File	⚠ @wanleee.jpeg	

201 Created · 626 ms · 591 B

{} JSON

```json
{
    "barang_kode": "BRG015",
    "barang_nama": "juxebox",
    "kategori_id": "3",
    "harga_beli": "150000",
    "harga_jual": "200000",
    "image": "http://127.0.0.1:8000/storage/barang/1745476702.jpg",
    "updated_at": "2025-04-24T06:38:22.000000Z",
    "created_at": "2025-04-24T06:38:22.000000Z",
    "barang_id": 12
}
```

Get



http://127.0.0.1:8000/api/barangs

GET http://127.0.0.1:8000/api/barangs

Query Params

Key	Value	Description
Key	Value	Description

200 OK · 659 ms · 2.42 KB

```
            "barang_kode": "BRG11",
            "barang_nama": "vest",
            "harga_beli": 100000,
            "harga_jual": 150000,
            "kategori_id": 2,
            "created_at": "2025-04-24T05:31:05.000000Z",
            "updated_at": "2025-04-24T05:31:05.000000Z",
            "image": null
        },
        {
            "barang_id": 12,
            "barang_kode": "BRG015",
            "barang_nama": "juxebox",
            "harga_beli": 150000,
            "harga_jual": 200000,
            "kategori_id": 3,
            "created_at": "2025-04-24T06:38:22.000000Z",
            "updated_at": "2025-04-24T06:38:22.000000Z",
            "image": "http://127.0.0.1:8000/storage/barang/1745476702.jpg"
        }
]
```