PEMROGRAMAN WEB LANJUT

JOBSHEET 11

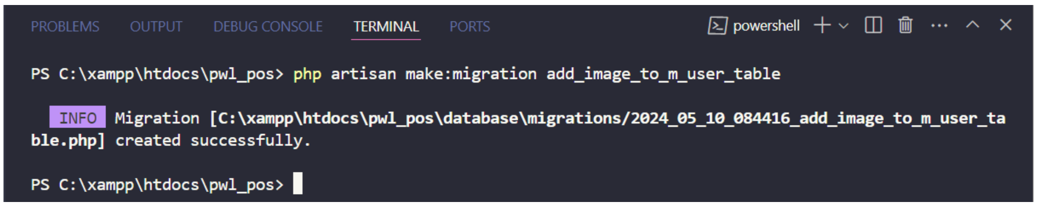
MUHAMMAD IQBAL MAKMUR AL-MUNIRI

TI-2C / 15 / 2241720099

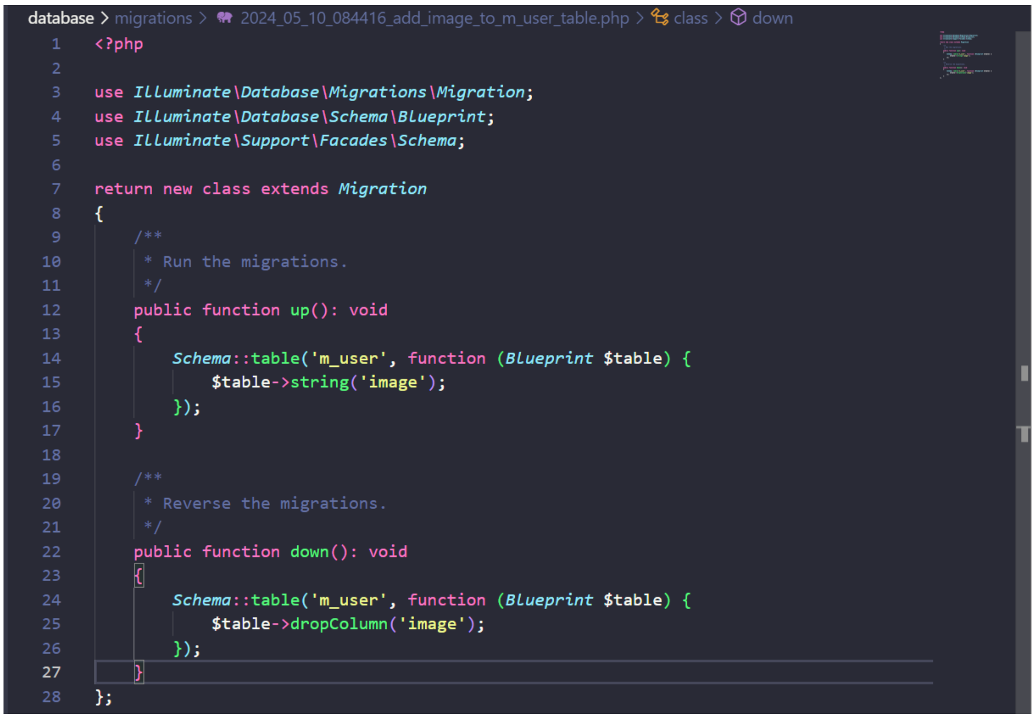
TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

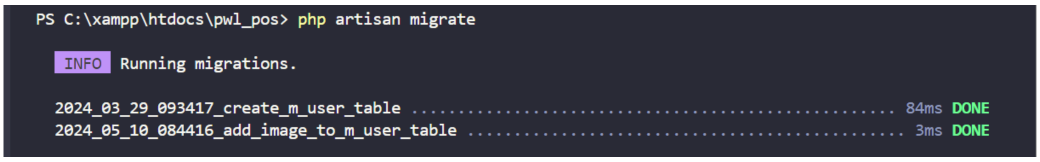
Praktikum 1 – Implementasi Eloquent Accessor

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

```
PS C:\xampp\htdocs\pwl_pos> php artisan make:migration add_image_to_m_user_table

   INFO  Migration [C:\xampp\htdocs\pwl_pos\database\migrations/2024_05_10_084416_add_image_to_m_user_ta
ble.php] created successfully.

PS C:\xampp\htdocs\pwl_pos>
```

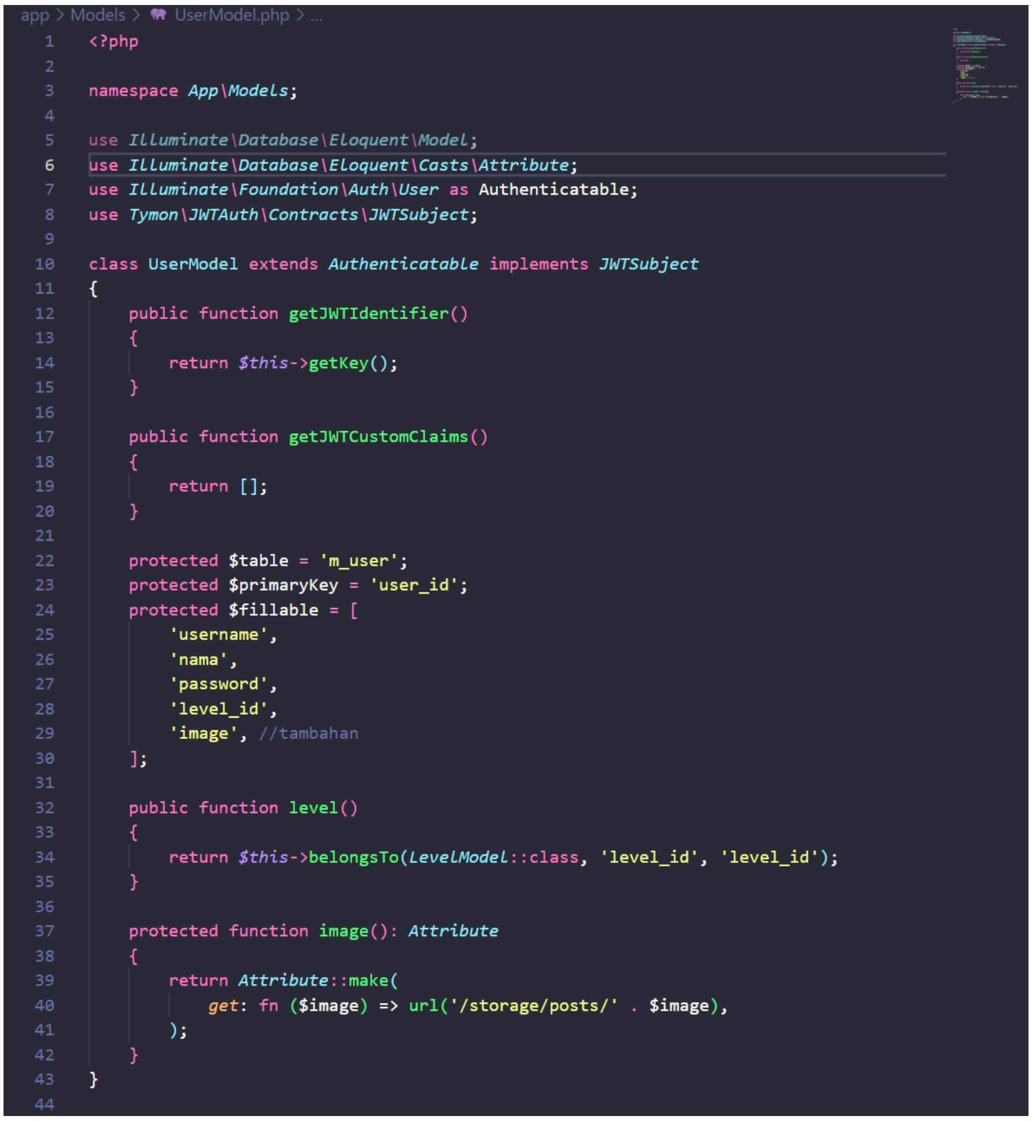
3. Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan
   Jawab:
   
   database > migrations > 2024_05_10_084416_add_image_to_m_user_table.php > class > down

```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
   Jawab:
   

```
PS C:\xampp\htdocs\pwl_pos> php artisan migrate

   INFO  Running migrations.

  2024_03_29_093417_create_m_user_table ................................................... 84ms DONE
  2024_05_10_084416_add_image_to_m_user_table ............................................. 3ms DONE
```

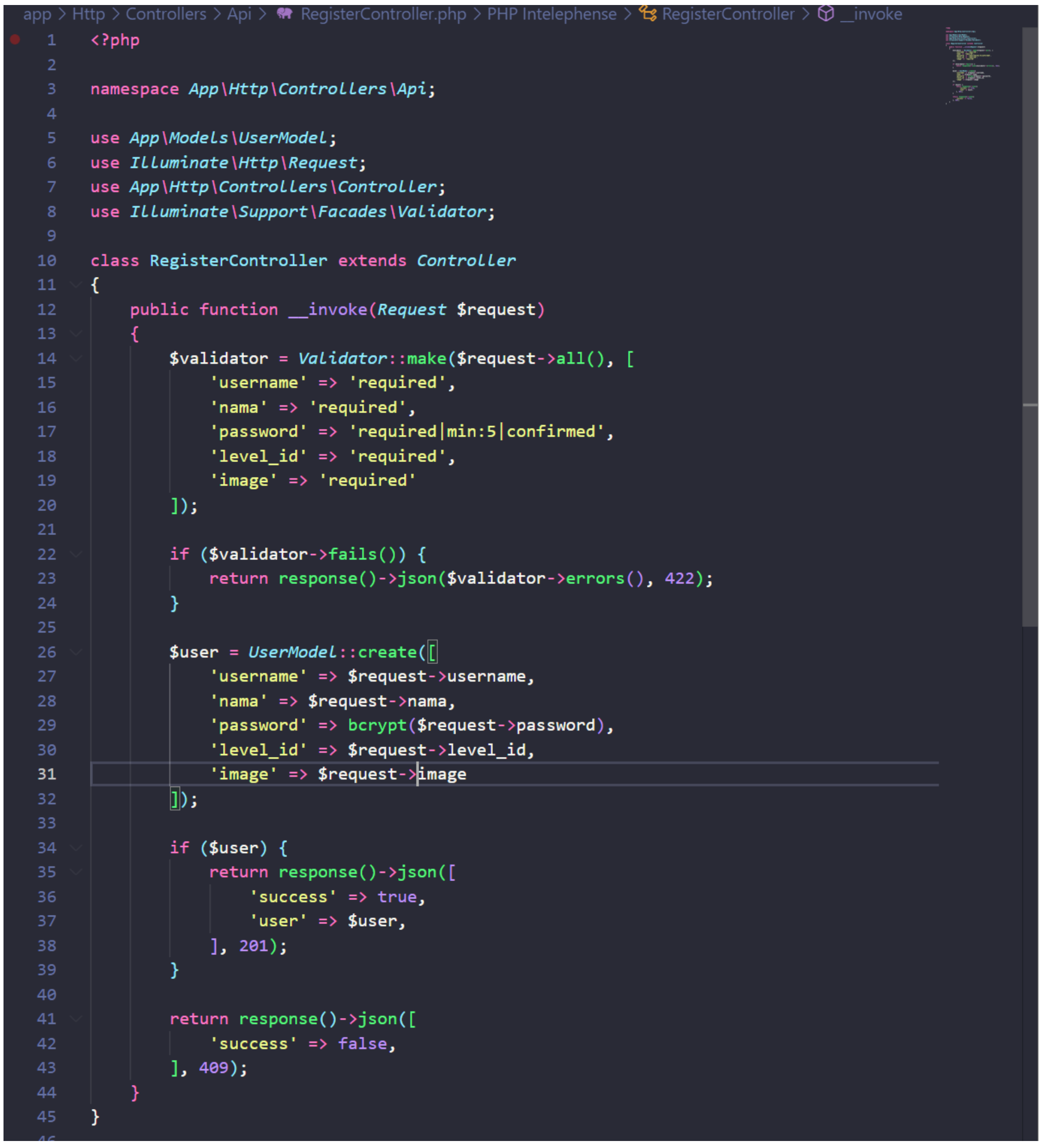
5. Lalu lakukan modifikasi models pada App/Models/UserModel.php
   Jawab:



app > Models > UserModel.php > ...

```php
<?php

namespace App\Models;

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

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),
        );
    }
}

```

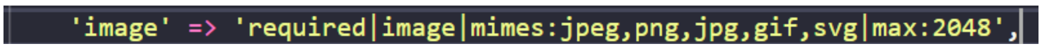
6. Lakukan modifikasi pada Controllers/Api/RegisterController
   Jawab:



app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController > __invoke

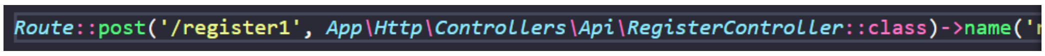
```php
<?php

namespace App\Http\Controllers\Api;

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

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

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

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

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

        return response()->json([
            'success' => false,
        ], 409);
    }
}
```

7. Anda dapat menambahkan detail untuk spesifikasi image pada validator
   Jawab:



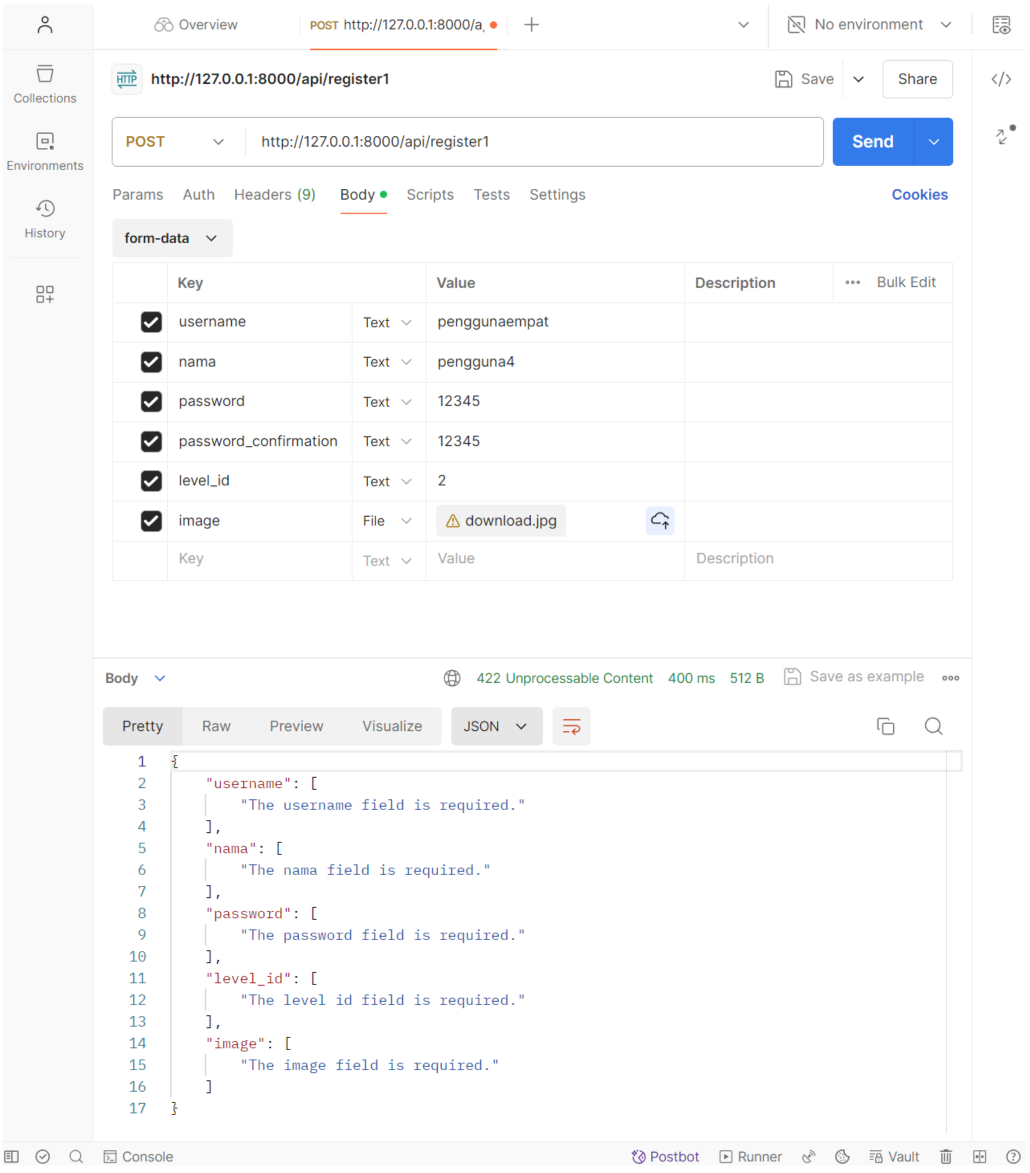
```php
    'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

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



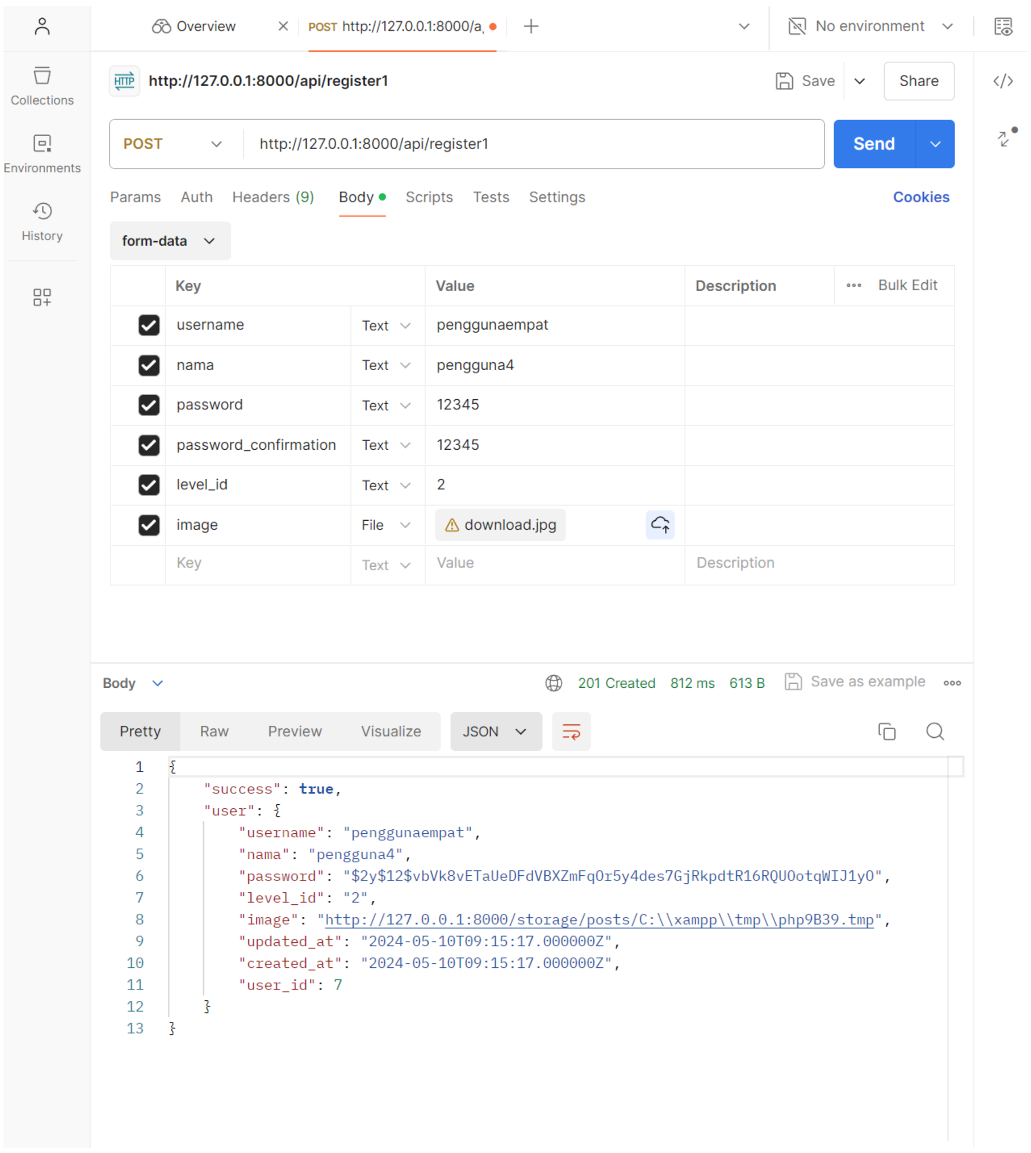
```php
Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('r
```

9. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual key dan Valuenya pada Key Image tambahkan value file dan upload gambar.
Jawab:



Key		Value	Description
✓ username	Text	penggunaempat	
✓ nama	Text	pengguna4	
✓ password	Text	12345	
✓ password_confirmation	Text	12345	
✓ level_id	Text	2	
✓ image	File	download.jpg	
Key	Text	Value	Description

```
{
    "username": [
        "The username field is required."
    ],
    "nama": [
        "The nama field is required."
    ],
    "password": [
        "The password field is required."
    ],
    "level_id": [
        "The level id field is required."
    ],
    "image": [
        "The image field is required."
    ]
}
```



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

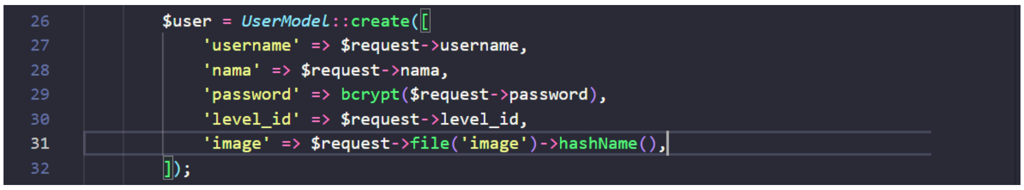
form-data

	Key		Value	Description
✓	username	Text	penggunaempat	
✓	nama	Text	pengguna4	
✓	password	Text	12345	
✓	password_confirmation	Text	12345	
✓	level_id	Text	2	
✓	image	File	download.jpg	

201 Created 812 ms 613 B

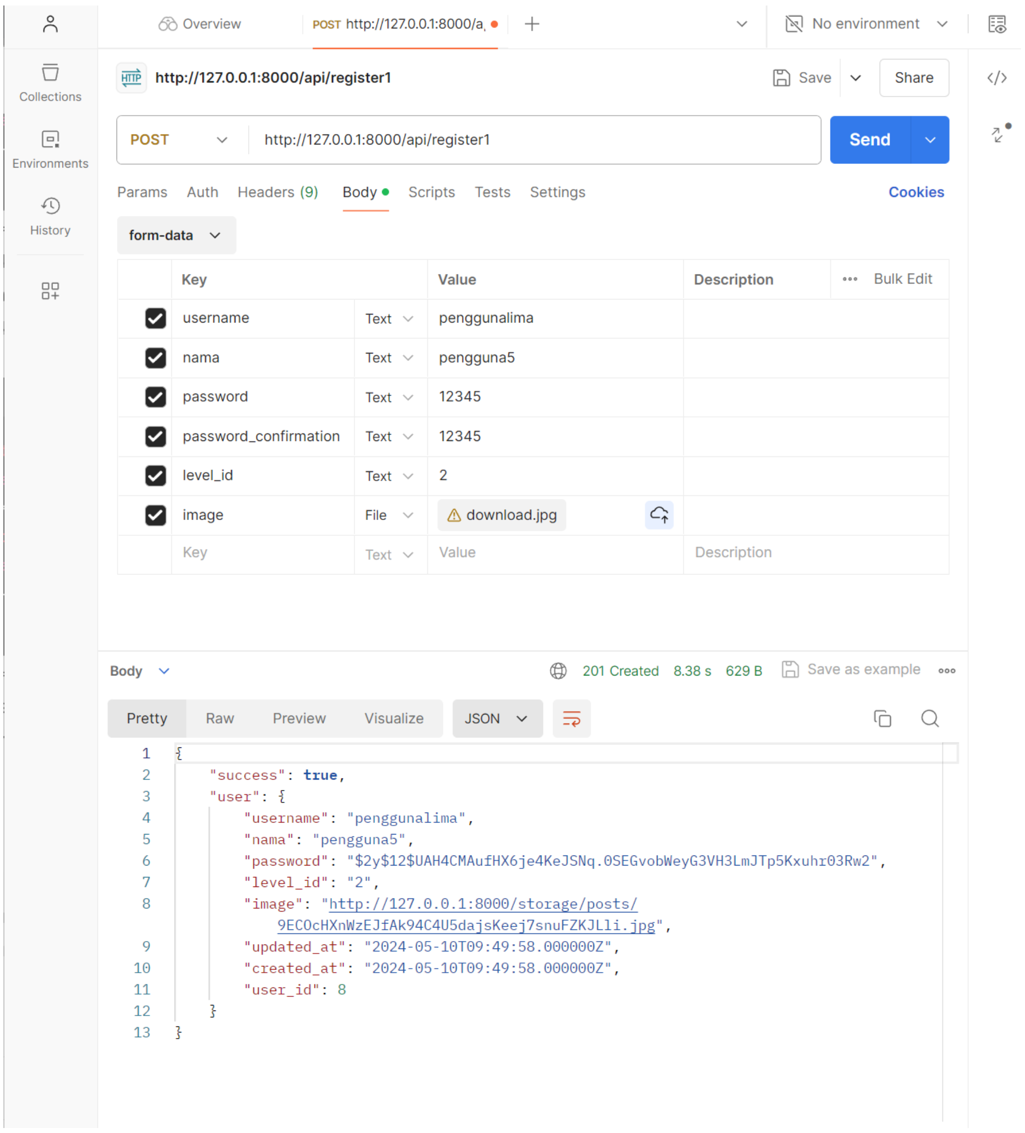
```
{
    "success": true,
    "user": {
        "username": "penggunaempat",
        "nama": "pengguna4",
        "password": "$2y$12$vbVk8vETaUeDFdVBXZmFqOr5y4des7GjRkpdtR16RQUOotqWIJ1yO",
        "level_id": "2",
        "image": "http://127.0.0.1:8000/storage/posts/C:\\xampp\\tmp\\php9B39.tmp",
        "updated_at": "2024-05-10T09:15:17.000000Z",
        "created_at": "2024-05-10T09:15:17.000000Z",
        "user_id": 7
    }
}
```

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



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

11. Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya.
    Jawab:



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

form-data

	Key		Value	Description
✓	username	Text	penggunalima	
✓	nama	Text	pengguna5	
✓	password	Text	12345	
✓	password_confirmation	Text	12345	
✓	level_id	Text	2	
✓	image	File	download.jpg	

201 Created 8.38 s 629 B

```
{
    "success": true,
    "user": {
        "username": "penggunalima",
        "nama": "pengguna5",
        "password": "$2y$12$UAH4CMAufHX6je4KeJSNq.0SEGvobWeyG3VH3LmJTp5Kxuhr03Rw2",
        "level_id": "2",
        "image": "http://127.0.0.1:8000/storage/posts/
            9ECOcHXnWzEJfAk94C4U5dajsKeej7snuFZKJLli.jpg",
        "updated_at": "2024-05-10T09:49:58.000000Z",
        "created_at": "2024-05-10T09:49:58.000000Z",
        "user_id": 8
    }
}
```

Penjelasan

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

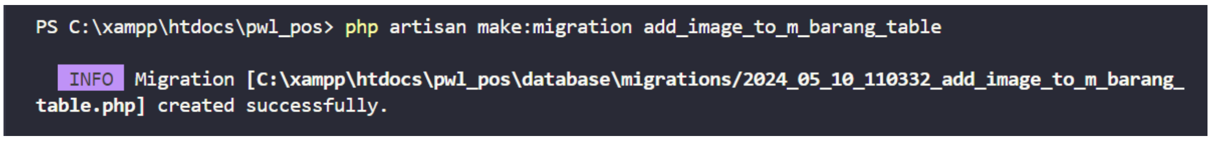
Penggunaan kode diatas bertujuan agar nama file diubah menjadi unique data sehingga file tidak akan bertabrakan dengan file lainnya. Digunakan untuk menghindari konflik nama file yang sama.

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.

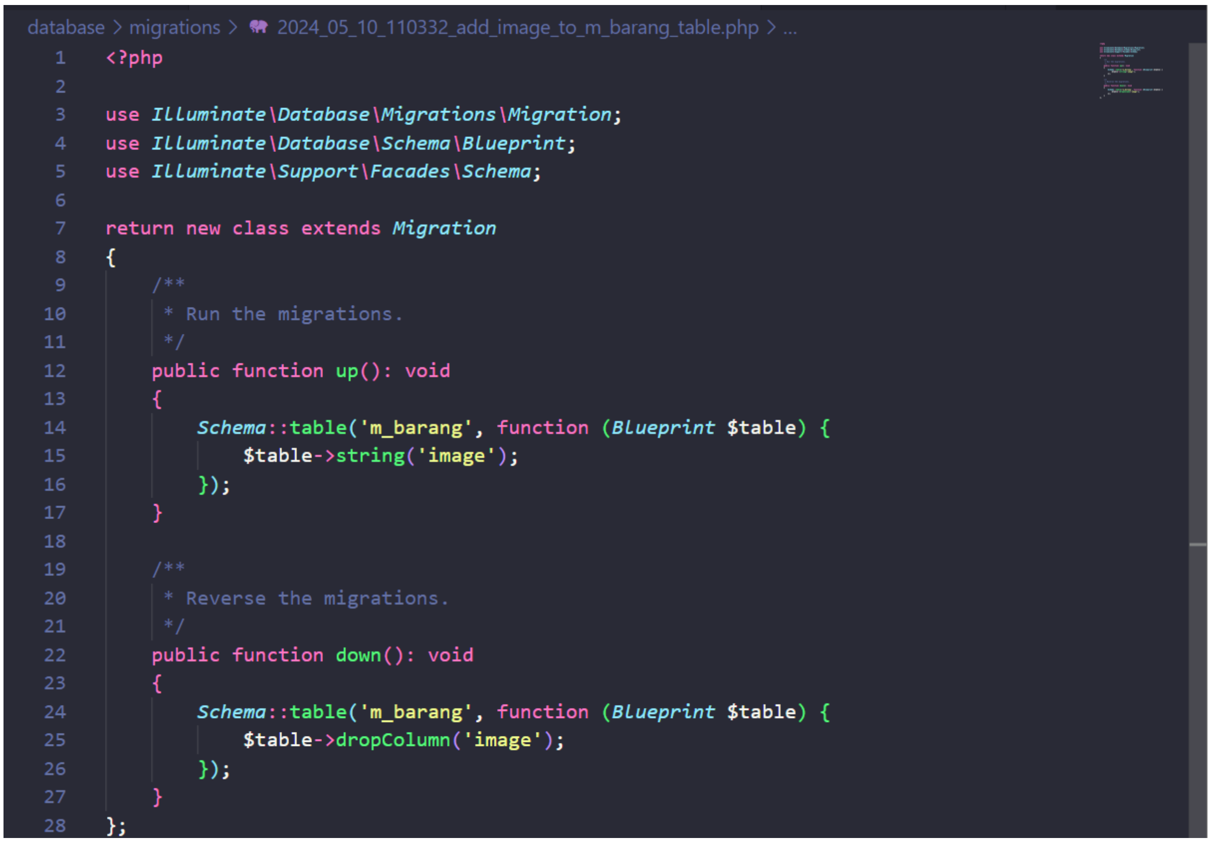
Jawab:

Php artisan make:migration add_image_to_m_barang_table



```
PS C:\xampp\htdocs\pwl_pos> php artisan make:migration add_image_to_m_barang_table

  INFO  Migration [C:\xampp\htdocs\pwl_pos\database\migrations/2024_05_10_110332_add_image_to_m_barang_
table.php] created successfully.
```

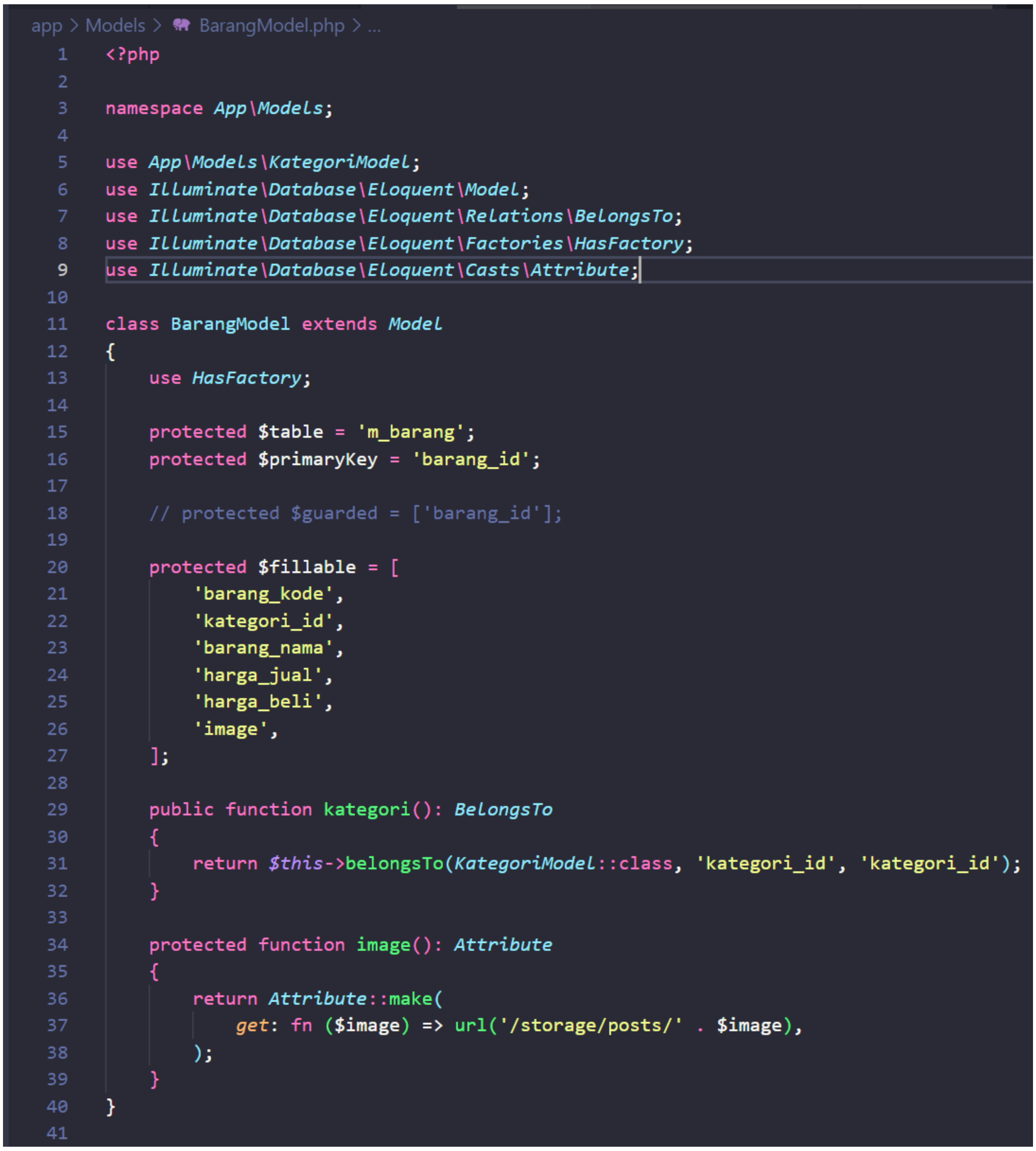
Add_image_to_m_barang



database > migrations > 2024_05_10_110332_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
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('m_barang', function (Blueprint $table) {
            $table->string('image');
        });
    }

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

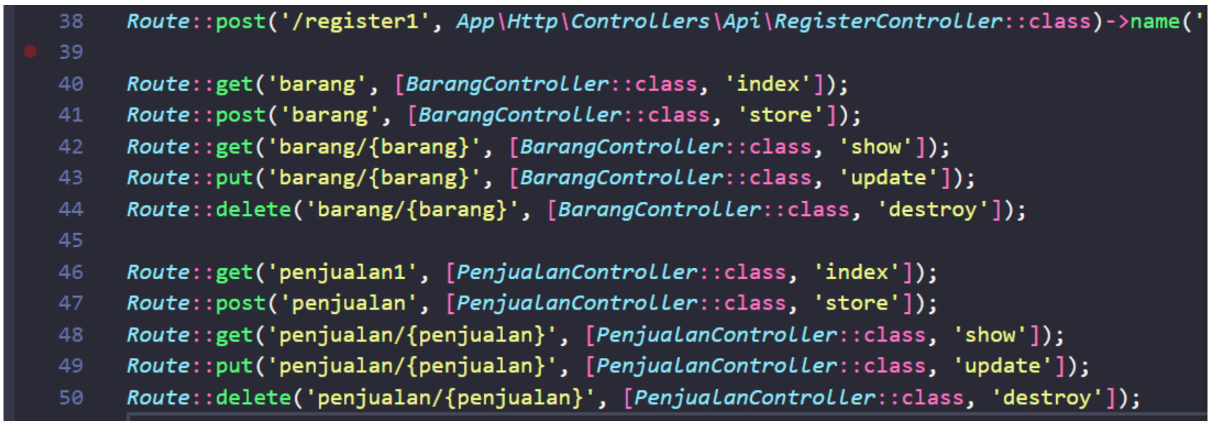
BarangModel.php



app > Models > BarangModel.php > ...

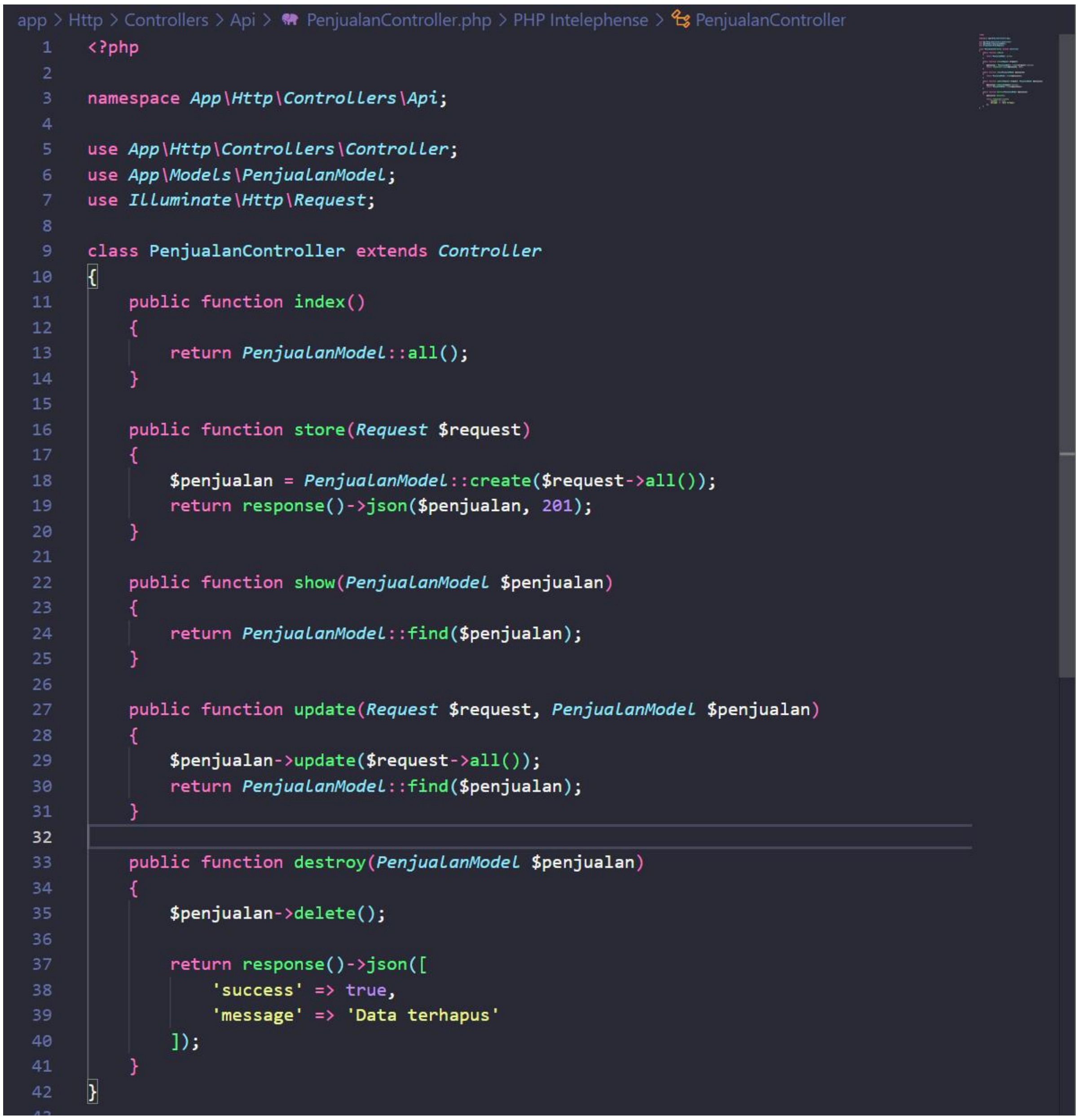
```php
<?php

namespace App\Models;

use App\Models\KategoriModel;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Casts\Attribute;

class BarangModel extends Model
{
    use HasFactory;

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

    // protected $guarded = ['barang_id'];

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

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

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

```

Routes



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

Route::get('barang', [BarangController::class, 'index']);
Route::post('barang', [BarangController::class, 'store']);
Route::get('barang/{barang}', [BarangController::class, 'show']);
Route::put('barang/{barang}', [BarangController::class, 'update']);
Route::delete('barang/{barang}', [BarangController::class, 'destroy']);

Route::get('penjualan1', [PenjualanController::class, 'index']);
Route::post('penjualan', [PenjualanController::class, 'store']);
Route::get('penjualan/{penjualan}', [PenjualanController::class, 'show']);
Route::put('penjualan/{penjualan}', [PenjualanController::class, 'update']);
Route::delete('penjualan/{penjualan}', [PenjualanController::class, 'destroy']);
```

PenjualanController.php (API)



app > Http > Controllers > Api > PenjualanController.php > PHP Intelephense > PenjualanController

```php
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\PenjualanModel;
use Illuminate\Http\Request;

class PenjualanController extends Controller
{
    public function index()
    {
        return PenjualanModel::all();
    }

    public function store(Request $request)
    {
        $penjualan = PenjualanModel::create($request->all());
        return response()->json($penjualan, 201);
    }

    public function show(PenjualanModel $penjualan)
    {
        return PenjualanModel::find($penjualan);
    }

    public function update(Request $request, PenjualanModel $penjualan)
    {
        $penjualan->update($request->all());
        return PenjualanModel::find($penjualan);
    }

    public function destroy(PenjualanModel $penjualan)
    {
        $penjualan->delete();

        return response()->json([
            'success' => true,
            'message' => 'Data terhapus'
        ]);
    }
}
```