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
https://jti.polinema.ac.id

Mata Kuliah	:	Pemrograman Web Lanjut (PWL)
Program Studi	:	D4 – Sistem Informasi Bisnis
Semester	:	5

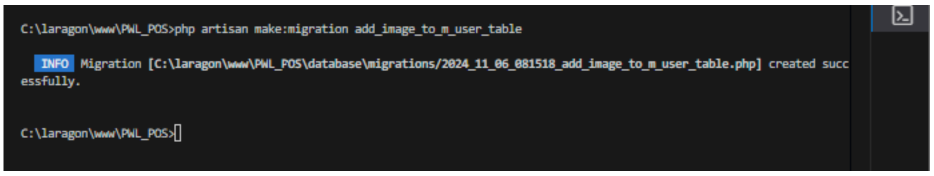
Kelas	:	SIB
NIM	:	2241760035
Nama	:	Fifi Novitasari
Jobsheet Ke-	:	11

Laporan Jobsheet 11

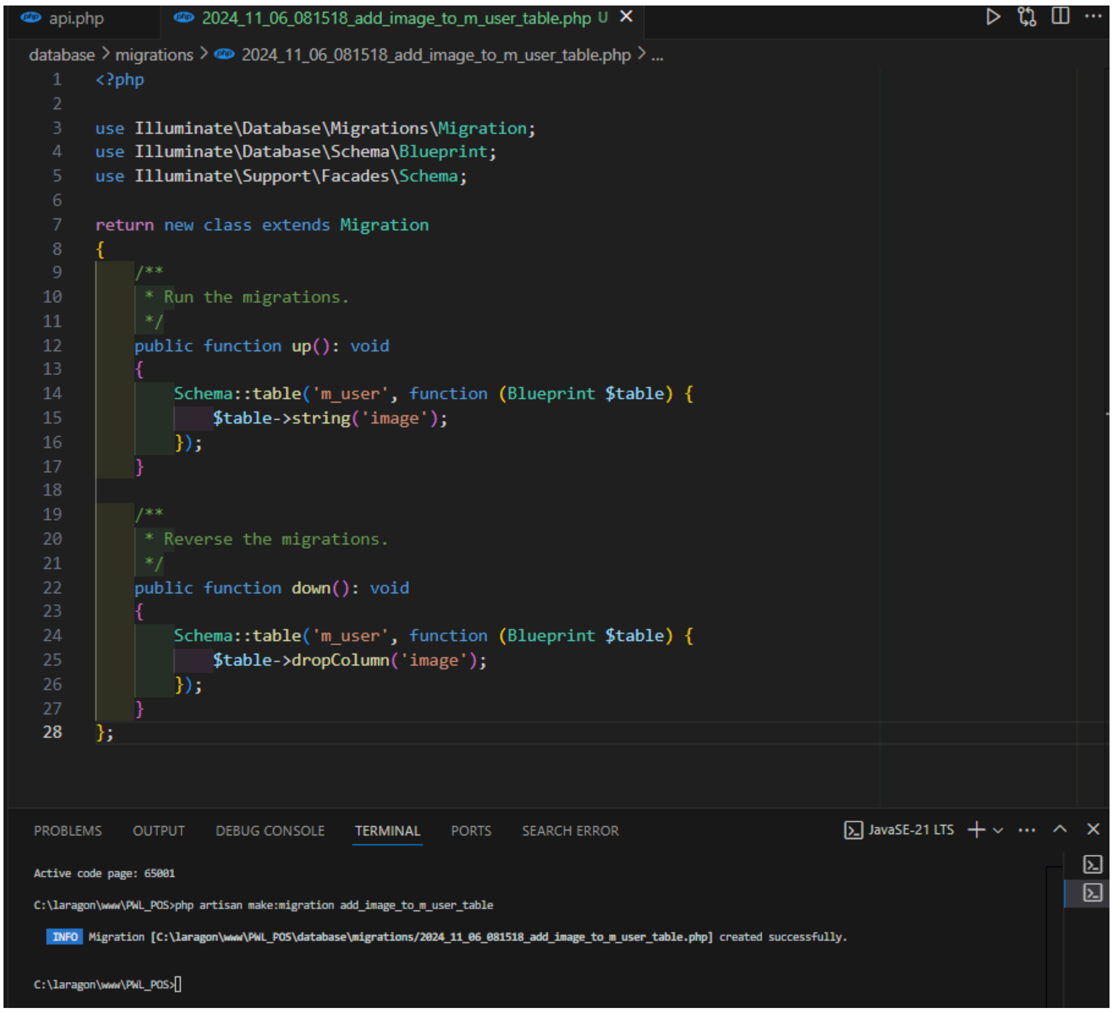
RESTFUL API 2

Praktikum Ke-1 Implementasi Eloquent Accessor

Langkah	Jawaban/Deskripsi

1	Memodifikasi Table m_user dengan menambahkan column : image, buka terminal lalu ketikkan **php artisan make:migration add_image_to_m_user_table**
	
	`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_11_06_081518_add_image_to_m_user_table.php] created succ`
	`essfully.`
	`C:\laragon\www\PWL_POS>`

2	Memodifikasi file migrasi



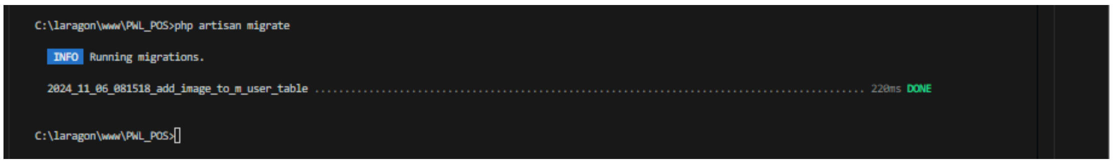
```
<?php

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');
        });
    }
};
```

```
Active code page: 65001

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_11_06_081518_add_image_to_m_user_table.php] created successfully.

C:\laragon\www\PWL_POS>
```

3

Melakukan jalankan update migrasi dengan cara:

php artisan migrate



```
C:\laragon\www\PWL_POS>php artisan migrate

  INFO  Running migrations.

  2024_11_06_081518_add_image_to_m_user_table ........................................................................ 220ms DONE

C:\laragon\www\PWL_POS>
```

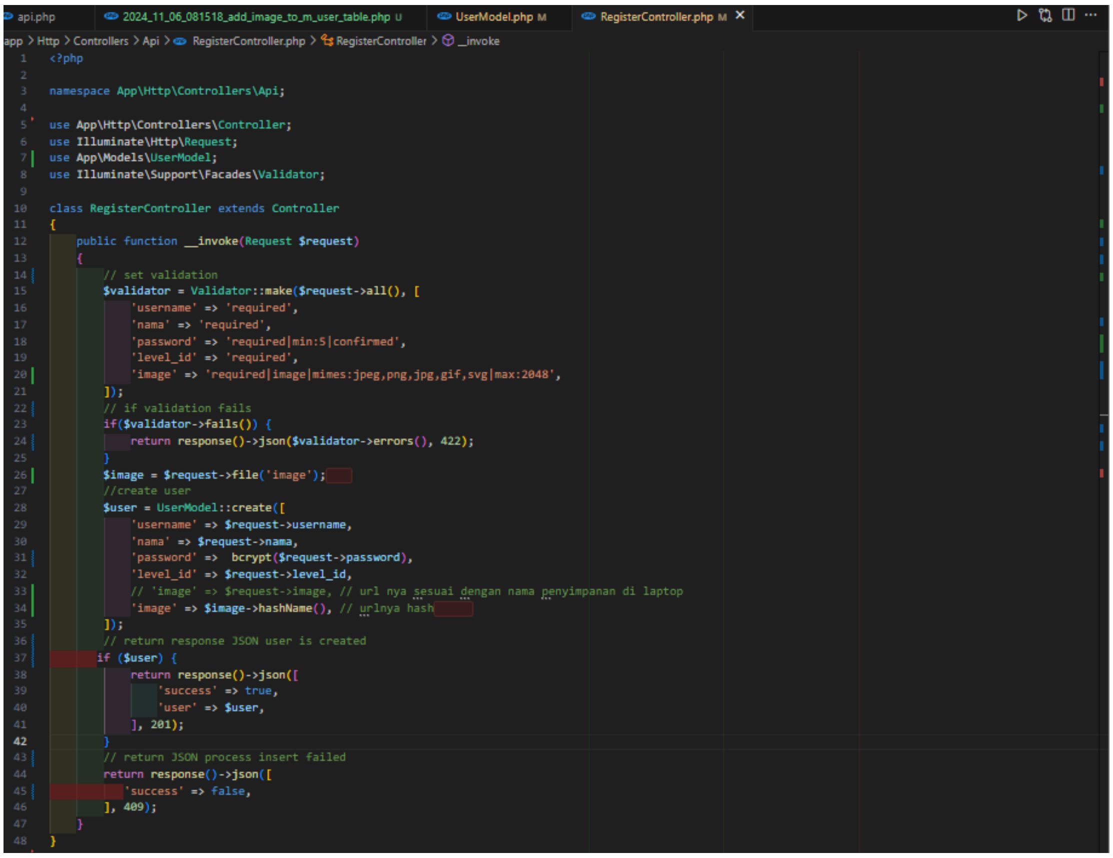
4

Memodifikasi models pada App/Models/UserModel.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
// use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Foundation\Auth\User as Authenticatable;
```

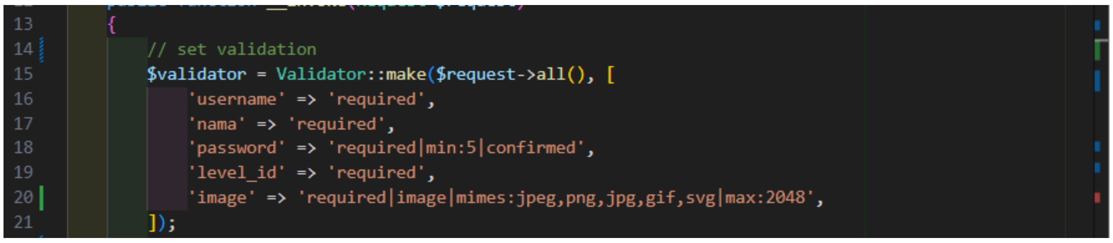
```php
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Database\Eloquent\Casts\Attribute;

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

    public function getJWTCustomClaims() {
        return [];
    }

    use HasFactory;
    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
    // protected $fillable =['level_id', 'username', 'nama', 'password', 'created_at', 
'profile_image', 'updated_at'];
    protected $fillable = [
        'username',
        'nama',
        'password',
        'level_id',
        'image'
    ];

    protected $casts = ['password' => 'hashed'];

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

    //nama role
    public function getRoleName(): string{
        return $this->level->level_nama;
    }

    // apakah user memiliki role tertentu
    public function hasRole($role): bool{
        return $this ->level->level_kode == $role;
    }

    //Mendapatkan kode role
    public function getRole()
    {
        return $this->level->level_kode;
    }

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

5	Memodifikasi pada Controllers/Api/RegisterController


api.php
2024_11_06_081518_add_image_to_m_user_table.php U
UserModel.php M
RegisterController.php M
app > Http > Controllers > Api > RegisterController.php > RegisterController > __invoke

```php
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\UserModel;
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|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);
        // if validation fails
        if($validator->fails()) {
            return response()->json($validator->errors(), 422);
        }
        $image = $request->file('image');
        //create user
        $user = UserModel::create([
            'username' => $request->username,
            'nama' => $request->nama,
            'password' =>  bcrypt($request->password),
            'level_id' => $request->level_id,
            // 'image' => $request->image, // url nya sesuai dengan nama penyimpanan di laptop
            'image' => $image->hashName(), // urlnya hash
        ]);
        // return response JSON user is created
       if ($user) {
            return response()->json([
                'success' => true,
                'user' => $user,
            ], 201);
        }
        // return JSON process insert failed
        return response()->json([
           'success' => false,
        ], 409);
    }
}
```

6	Menambahkan detail untuk spesifikasi image pada validator

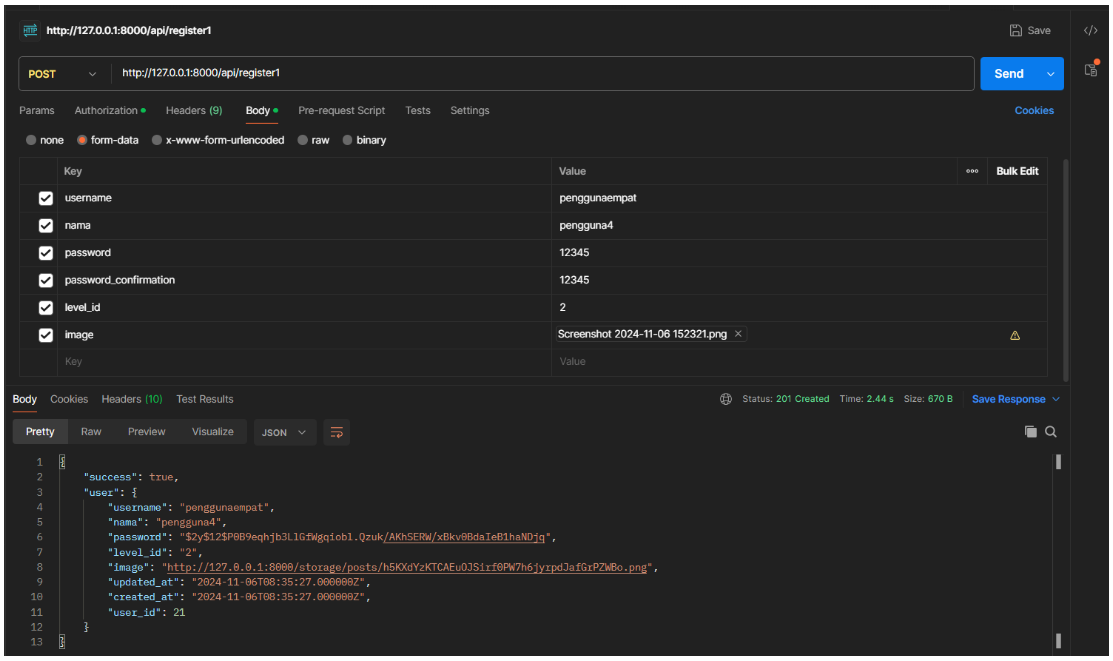


```
    {
        // set validation
        $validator = Validator::make($request->all(), [
            'username' => 'required',
            'nama' => 'required',
            'password' => 'required|min:5|confirmed',
            'level_id' => 'required',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',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

7 Menambahkan register1 pada routes/api.php

```
// praktikum 1 js 11
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



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

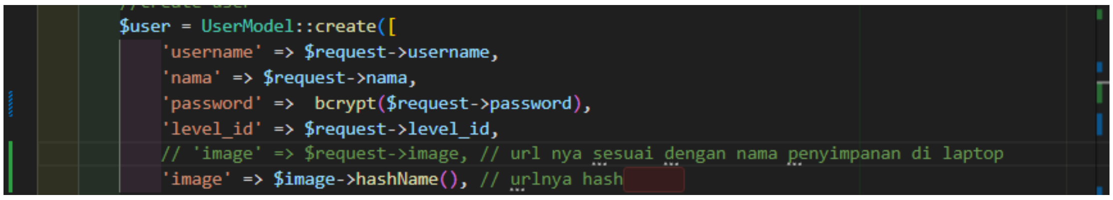
Body: form-data

Key	Value
username	penggunaempat
nama	pengguna4
password	12345
password_confirmation	12345
level_id	2
image	Screenshot 2024-11-06 152321.png

Status: 201 Created Time: 2.44 s Size: 670 B

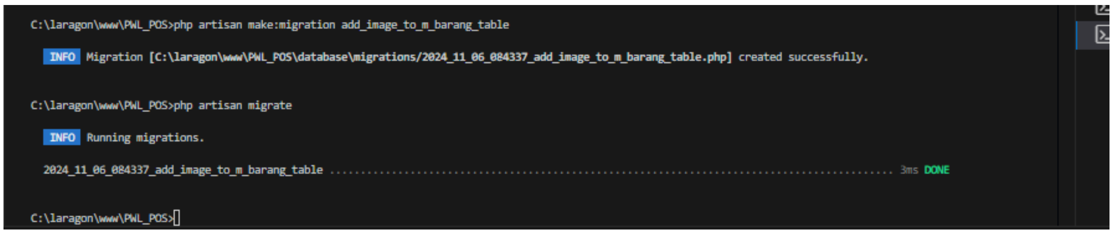
```
{
    "success": true,
    "user": {
        "username": "penggunaempat",
        "nama": "pengguna4",
        "password": "$2y$12$P0B9eqhjb3LlGfWgqiobl.Qzuk/AKhSERW/xBkv0BdaIeB1haNDjq",
        "level_id": "2",
        "image": "http://127.0.0.1:8000/storage/posts/h5KXdYzKTCAEuOJSirf0PW7h6jyrpdJafGrPZWBo.png",
        "updated_at": "2024-11-06T08:35:27.000000Z",
        "created_at": "2024-11-06T08:35:27.000000Z",
        "user_id": 21
    }
}
```

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

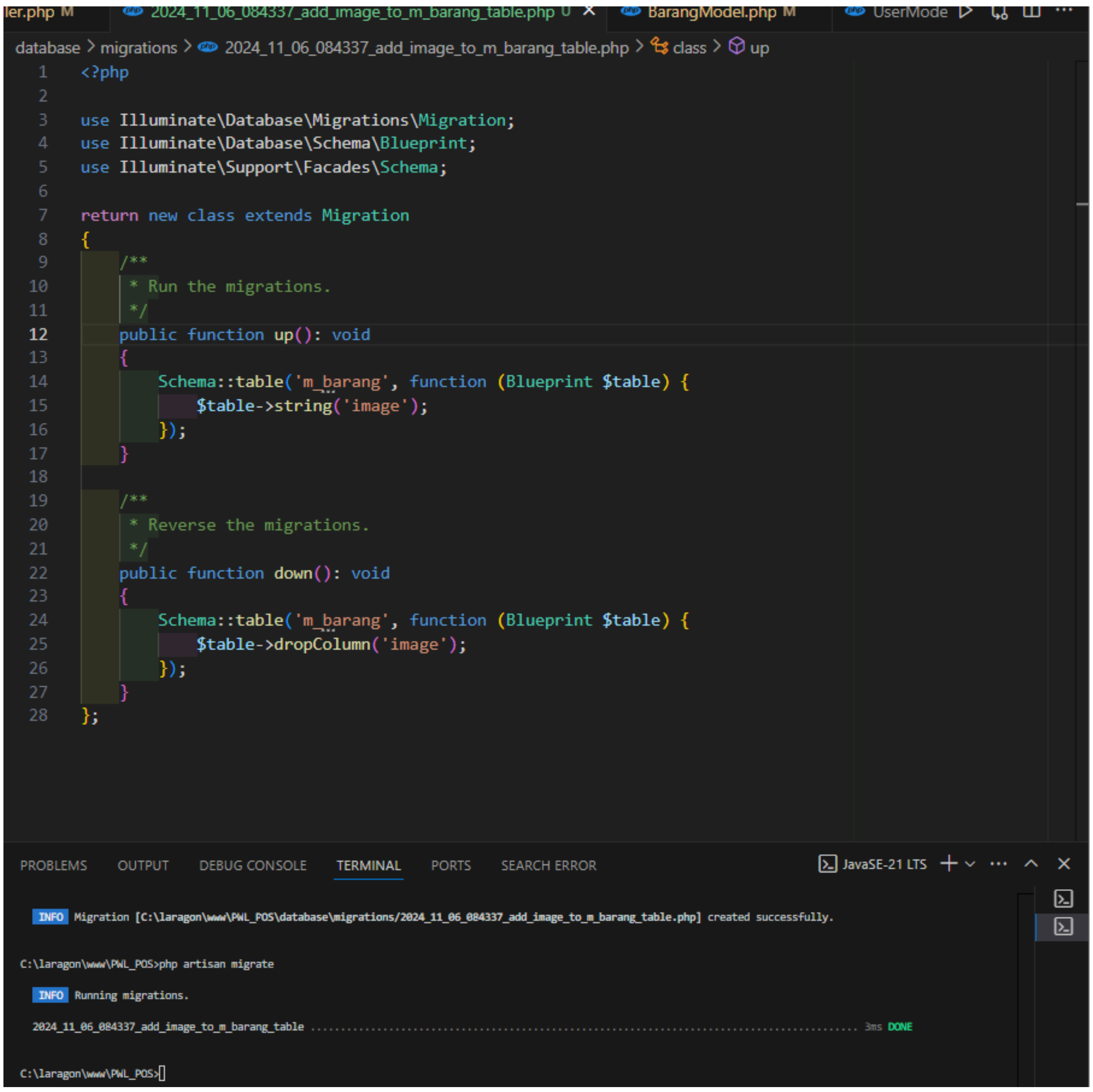


```
        $user = UserModel::create([
            'username' => $request->username,
            'nama' => $request->nama,
            'password' => bcrypt($request->password),
            'level_id' => $request->level_id,
            // 'image' => $request->image, // url nya sesuai dengan nama penyimpanan di laptop
            'image' => $image->hashName(), // urlnya hash
```

Tugas Praktikum

Langkah	Jawaban/Deskripsi
1	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.
2	Tabel m_barang 

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

  INFO  Migration [C:\laragon\www\PWL_POS\database\migrations/2024_11_06_084337_add_image_to_m_barang_table.php] created successfully.

C:\laragon\www\PWL_POS>php artisan migrate

  INFO  Running migrations.

  2024_11_06_084337_add_image_to_m_barang_table ........................................................................ 3ms DONE

C:\laragon\www\PWL_POS>
```

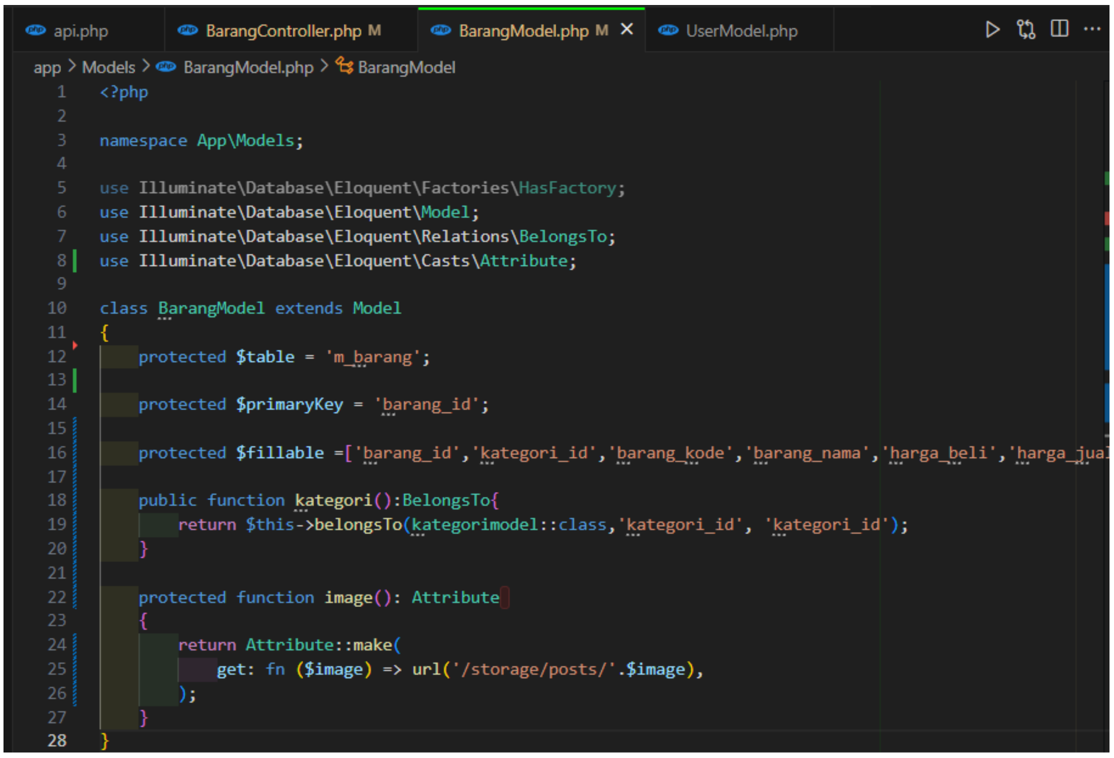
3	

ler.php M
2024_11_06_084337_add_image_to_m_barang_table.php U
BarangModel.php M
UserMode
database > migrations > 2024_11_06_084337_add_image_to_m_barang_table.php > class > up

```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');
        });
    }
};
```

```
  INFO  Migration [C:\laragon\www\PWL_POS\database\migrations/2024_11_06_084337_add_image_to_m_barang_table.php] created successfully.

C:\laragon\www\PWL_POS>php artisan migrate

  INFO  Running migrations.

  2024_11_06_084337_add_image_to_m_barang_table ........................................................................... 3ms DONE

C:\laragon\www\PWL_POS>
```

4	BarangModel.php



```php
<?php

namespace App\Models;

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
{
    protected $table = 'm_barang';

    protected $primaryKey = 'barang_id';

    protected $fillable =['barang_id','kategori_id','barang_kode','barang_nama','harga_beli','harga_jual

    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),
        );
    }
}
```

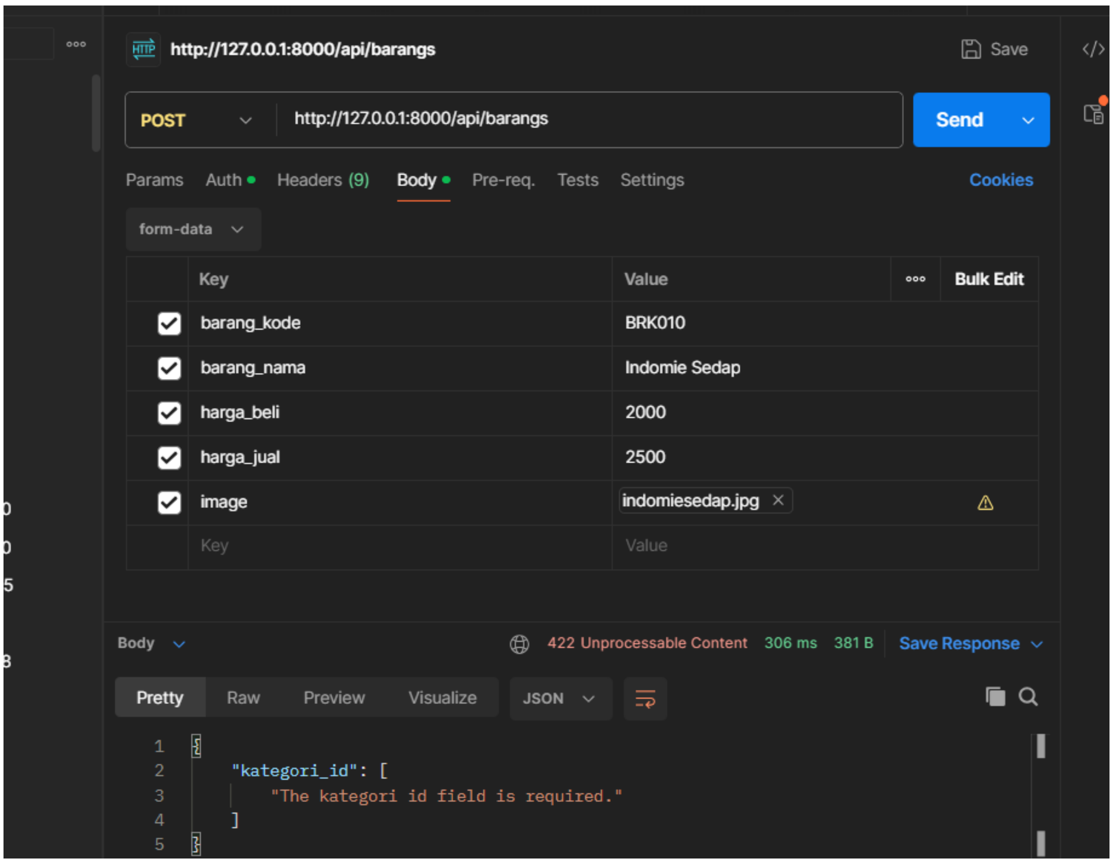
5

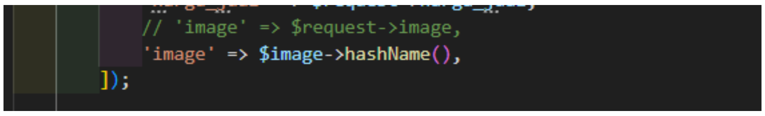
BarangController.php

```php
<?php

namespace App\Http\Controllers\Api;

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

class BarangController extends Controller
{

    public function index()
    {
        return BarangModel::all();
    }

    public function create() {}

    public function store(Request $request)
    {
        // Validasi data, termasuk validasi untuk gambar
        $validator = Validator::make($request->all(), [
            'barang_kode' => 'required|string|max:50',
            'barang_nama' => 'required|string|max:255',
            'kategori_id' => 'required|integer',
            'harga_beli' => 'required|numeric',
            'harga_jual' => 'required|numeric',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);

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

        // Proses upload gambar
        $image = $request->file('image');
        $imagePath = $image->store('public/posts'); // Simpan gambar di storage

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

        // Return response JSON jika barang berhasil dibuat
        return response()->json($barang, 201);

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

    public function show(BarangModel $barang)
    {
        return BarangModel::find($barang);
    }

    public function edit(string $id){
        // Cari data barang berdasarkan ID
        $barang = BarangModel::find($id);
        // Jika barang tidak ditemukan, kirimkan respons error
        if (!$barang) {
            return response()->json([
                'success' => false,
                'message' => 'Barang not found',
            ], 404);
        }

        // Kembalikan data barang sebagai respons JSON
        return response()->json([
            'success' => true,
            'barang' => $barang,
        ]);
    }

    public function update(Request $request, BarangModel $barang)
    {
        // Set validasi untuk data yang diupdate, termasuk gambar
        $validator = Validator::make($request->all(), [
            'barang_kode' => 'sometimes|string|max:50',
            'barang_nama' => 'sometimes|string|max:255',
            'kategori_id' => 'sometimes|integer',
            'harga_beli' => 'sometimes|numeric',
            'harga_jual' => 'sometimes|numeric',
            'image' => 'sometimes|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);

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

        // Proses upload gambar jika ada gambar baru
        if ($request->hasFile('image')) {
            // Hapus gambar lama jika ada
            $oldImage = public_path('posts/' . $barang->image);
            if (file_exists($oldImage)) {
                unlink($oldImage);
            }

            // Upload gambar baru
            $image = $request->file('image');
            $imageName = time() . '_' . $image->getClientOriginalName();
            $image->move(public_path('posts'), $imageName);
```

```php
            $barang->image = $imageName; // Simpan nama file gambar baru
        }

        // Update data barang
        $barang->update($request->except('image'));

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

    public function destroy(BarangModel $barang)
    {
        // Hapus gambar dari storage jika ada
        if ($barang->image) {
            Storage::delete('public/posts/' . $barang->image);
        }
        // Hapus data barang
        $barang->delete();

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

6	Memodifikasi api.php

7	
	http://127.0.0.1:8000/api/barangs
	POST http://127.0.0.1:8000/api/barangs
	Body: form-data
	barang_kode: BRK010
	barang_nama: Indomie Sedap
	harga_beli: 2000
	harga_jual: 2500
	image: indomiesedap.jpg
	422 Unprocessable Content 306 ms 381 B
	`{`
	`    "kategori_id": [`
	`        "The kategori id field is required."`
	`    ]`
	`}`
	image
	
	 katanya sih enak kalo dicampur???
	mediocreviews
	Indomie Mi Instan Mi Goreng Aceh
	Mie Sedaap Instant

8	
	`    // 'image' => $request->image,`
	`    'image' => $image->hashName(),`
	`]);`