PEMROGRAMAN WEB LANJUT

“RESTFUL API 2”

Kelas : TI-2H

Disusun Oleh :

Fanesabhirawaning Sulistyo

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

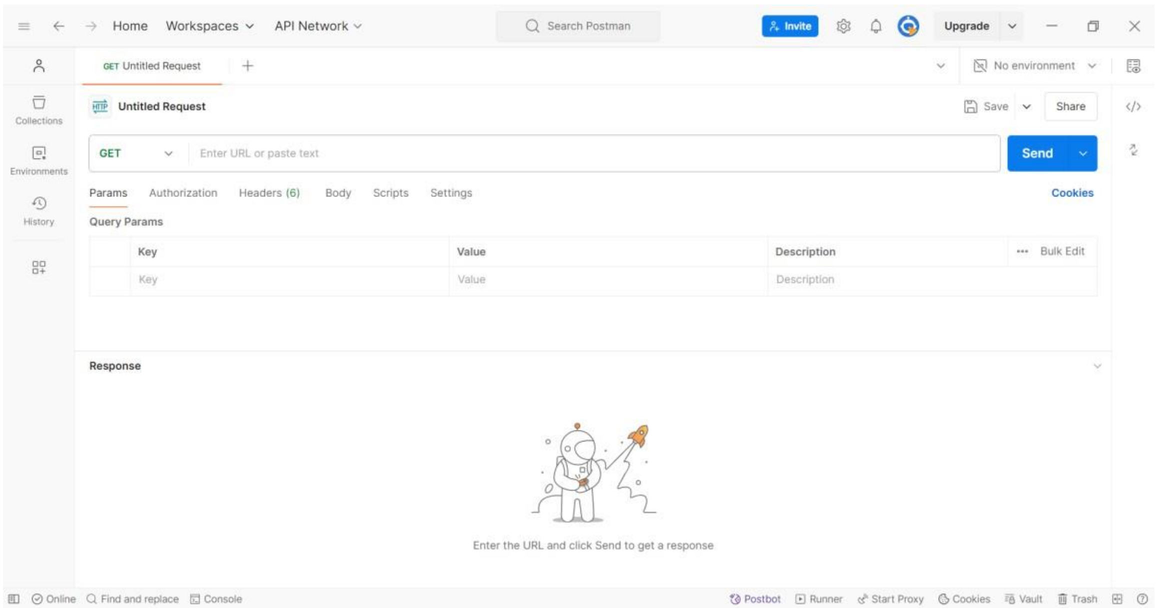
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

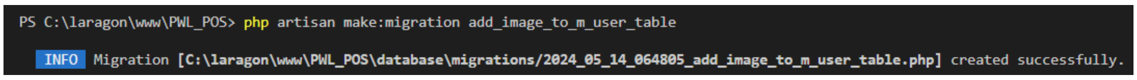
Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur 65141

Praktikum 1 – Implementasi Eloquent Accessor

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

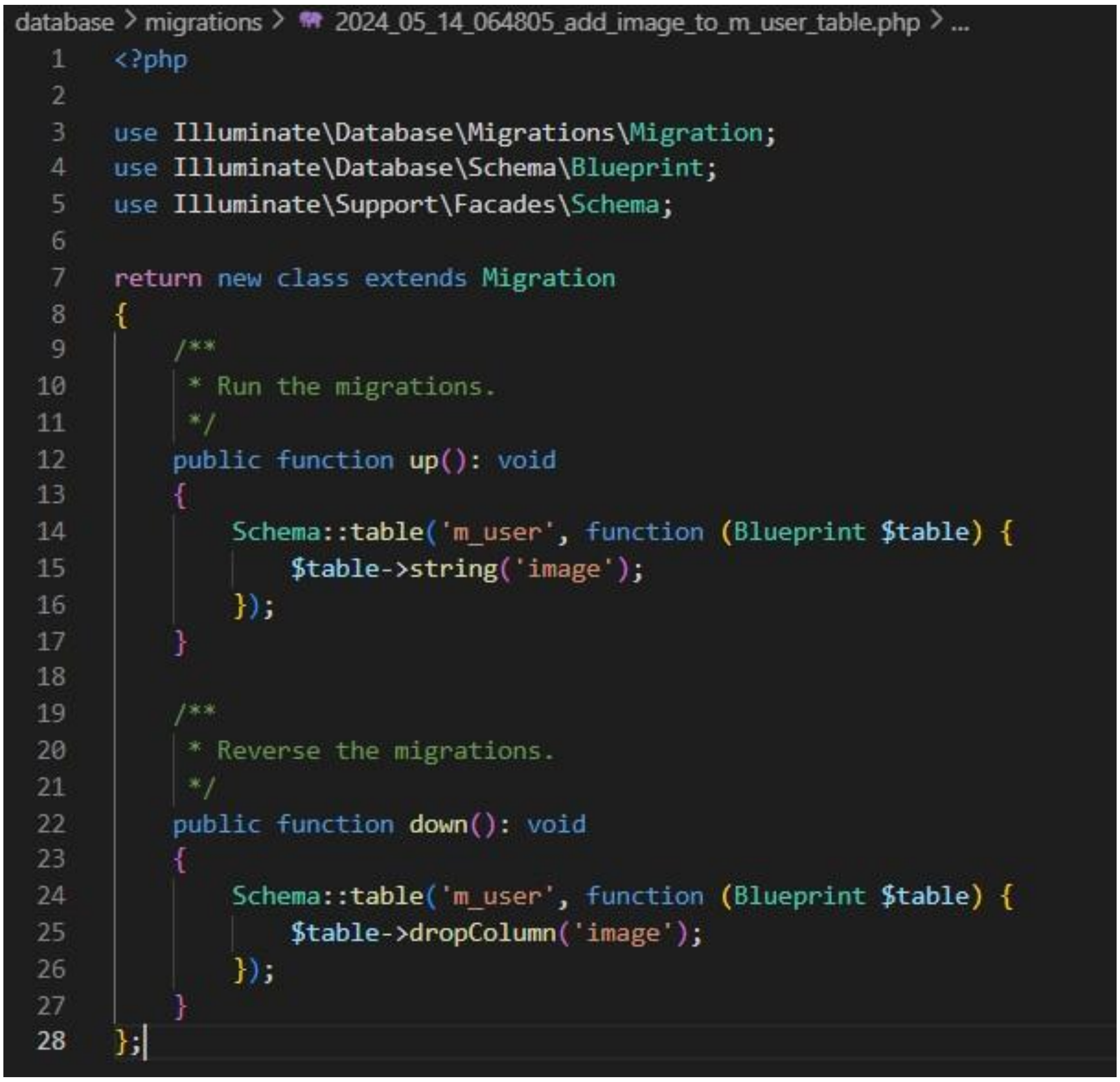


2. Kita akan memodifikasi Table m_user dengan menambahkan column : image, buka terminal lalu ketikkan
`php artisan make:migration add_image_to_m_user_table`



```
PS 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_05_14_064805_add_image_to_m_user_table.php] created successfully.
```

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

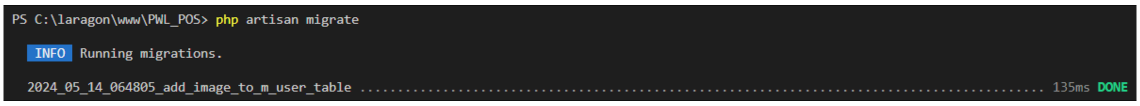


database > migrations > 2024_05_14_064805_add_image_to_m_user_table.php > ...

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

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

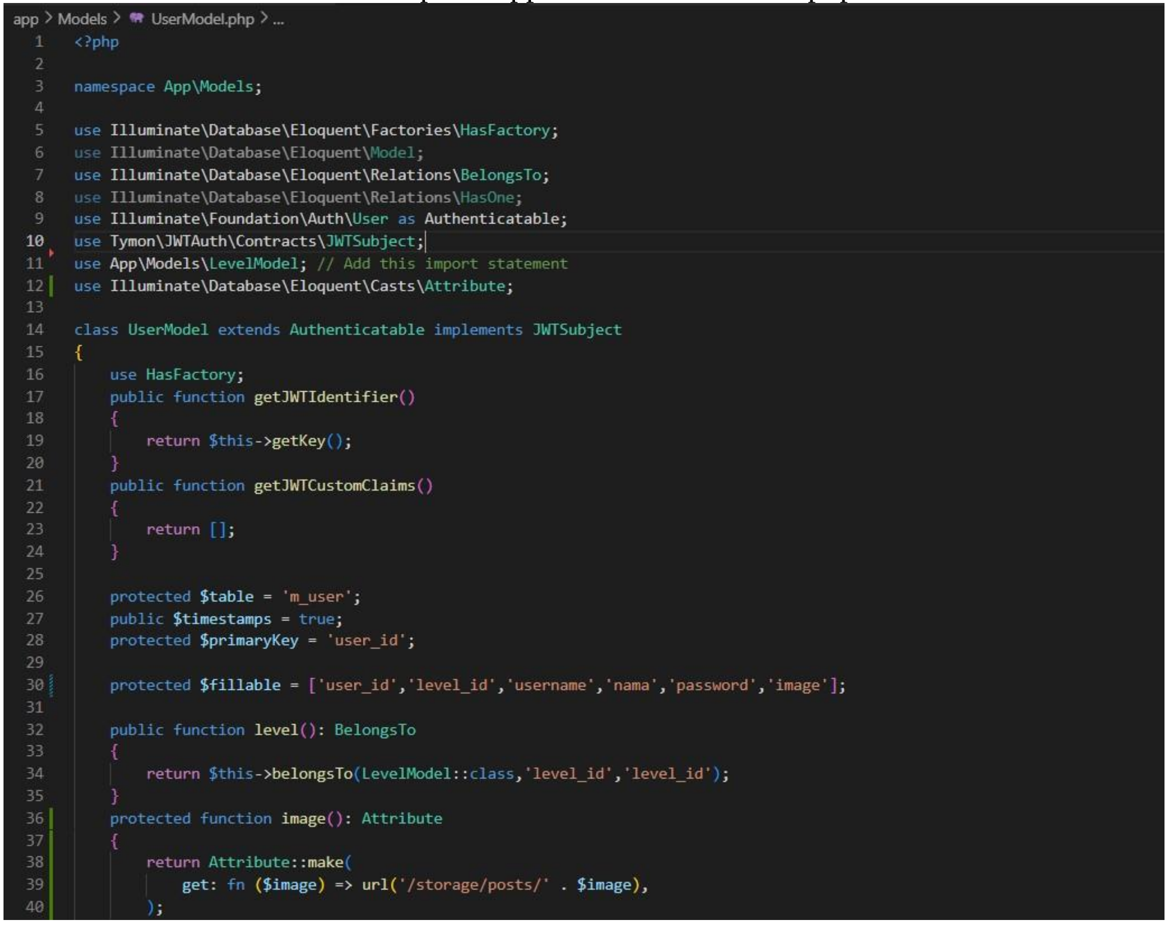
4. Lakukan jalankan update migrasi dengan cara:

```
php artisan migrate
```



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

  INFO  Running migrations.

  2024_05_14_064805_add_image_to_m_user_table .................................................................................... 135ms DONE
```

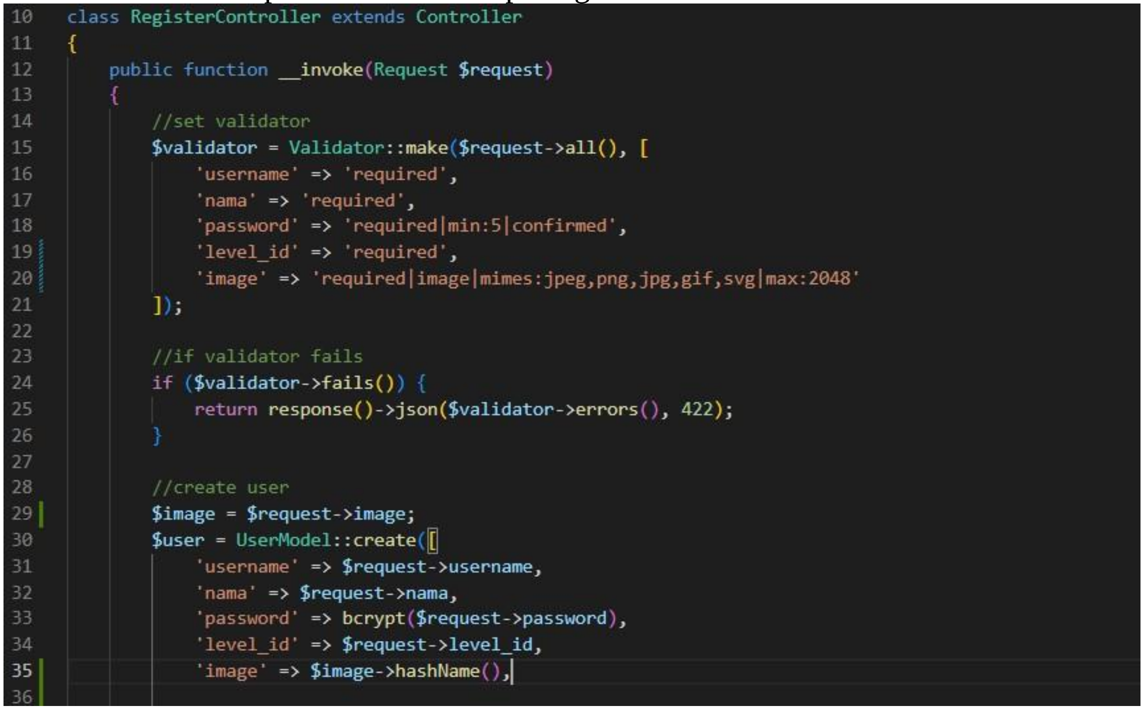
5. Lalu lakukan modifikasi models pada App/Models/UserModel.php



app > Models > UserModel.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\Relations\HasOne;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Tymon\JWTAuth\Contracts\JWTSubject;
use App\Models\LevelModel; // Add this import statement
use Illuminate\Database\Eloquent\Casts\Attribute;

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

    protected $table = 'm_user';
    public $timestamps = true;
    protected $primaryKey = 'user_id';

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

    public function level(): BelongsTo
    {
        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
class RegisterController extends Controller
{
    public function __invoke(Request $request)
    {
        //set 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'
        ]);

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

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

```

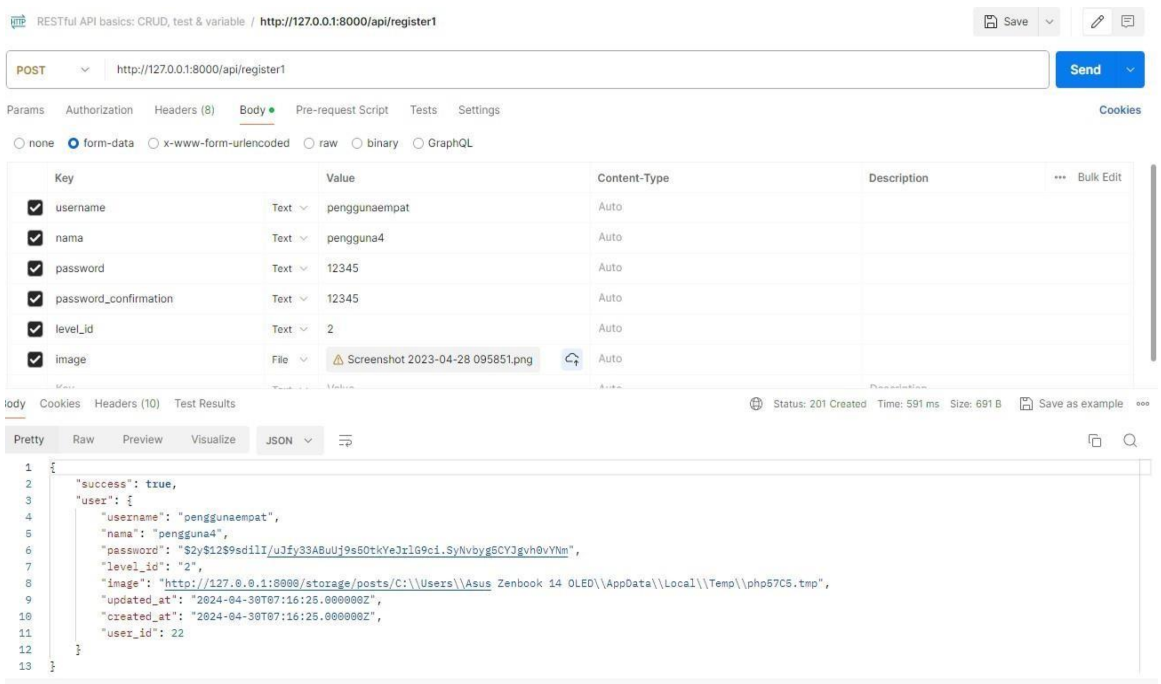
7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

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

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

```
63  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



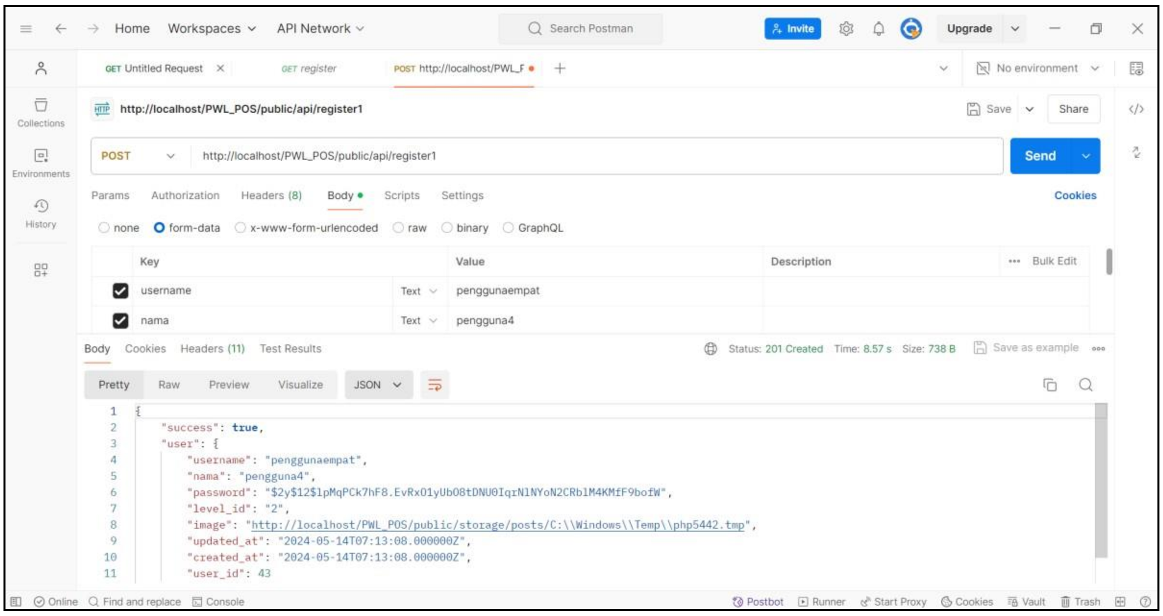
POST http://127.0.0.1:8000/api/register1 — Body: form-data

Key		Value	Content-Type
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

Status: 201 Created Time: 591 ms Size: 691 B

```
{
    "success": true,
    "user": {
        "username": "penggunaempat",
        "nama": "pengguna4",
        "password": "$2y$12$9sdilI/uJfy33ABuUj9s5OtkYeJrlG9ci.SyNvbyg5CYJgvh0vYNm",
        "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
    }
}
```

Jawab:



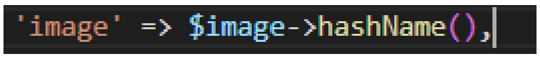
POST http://localhost/PWL_POS/public/api/register1 — Body: form-data

Key		Value
username	Text	penggunaempat
nama	Text	pengguna4

Status: 201 Created Time: 8.57 s Size: 738 B

```
{
    "success": true,
    "user": {
        "username": "penggunaempat",
        "nama": "pengguna4",
        "password": "$2y$12$1pMqPCk7hF8.EvRx01yUb08tDNU0IqrN1NYoN2CRblM4KMfF9bofW",
        "level_id": "2",
        "image": "http://localhost/PWL_POS/public/storage/posts/C:\\Windows\\Temp\\php5442.tmp",
        "updated_at": "2024-05-14T07:13:08.000000Z",
        "created_at": "2024-05-14T07:13:08.000000Z",
        "user_id": 43
```

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

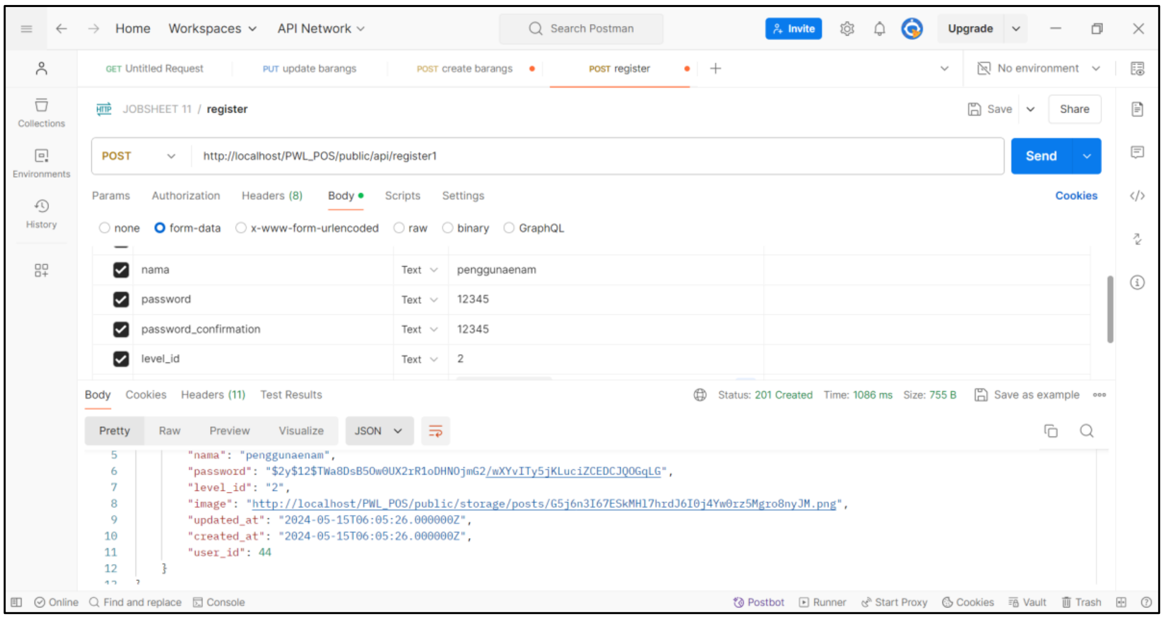


```
'image' => $image->hashName(),
```

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

Jawab:

Dengan perubahan 'image' => $request->image menjadi 'image' => $request->hashName(),
Maka nama image akan dienkripsi



POST http://localhost/PWL_POS/public/api/register1

	Key	Type	Value
✓	nama	Text	penggunaenam
✓	password	Text	12345
✓	password_confirmation	Text	12345
✓	level_id	Text	2

Status: 201 Created Time: 1086 ms Size: 755 B

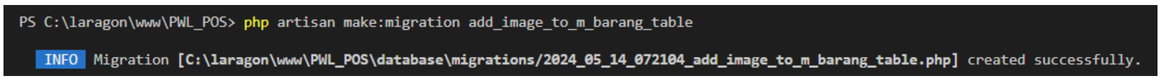
```
        "nama": "penggunaenam",
        "password": "$2y$12$TWa8DsB5Ow0UX2rR1oDHNOjmG2/wXYvITy5jKLuciZCEDCJQOGqLG",
        "level_id": "2",
        "image": "http://localhost/PWL_POS/public/storage/posts/G5j6n3I67ESkMHl7hrdJ6I0j4Yw0rz5Mgro8nyJM.png",
        "updated_at": "2024-05-15T06:05:26.000000Z",
        "created_at": "2024-05-15T06:05:26.000000Z",
        "user_id": 44
    }
```

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.

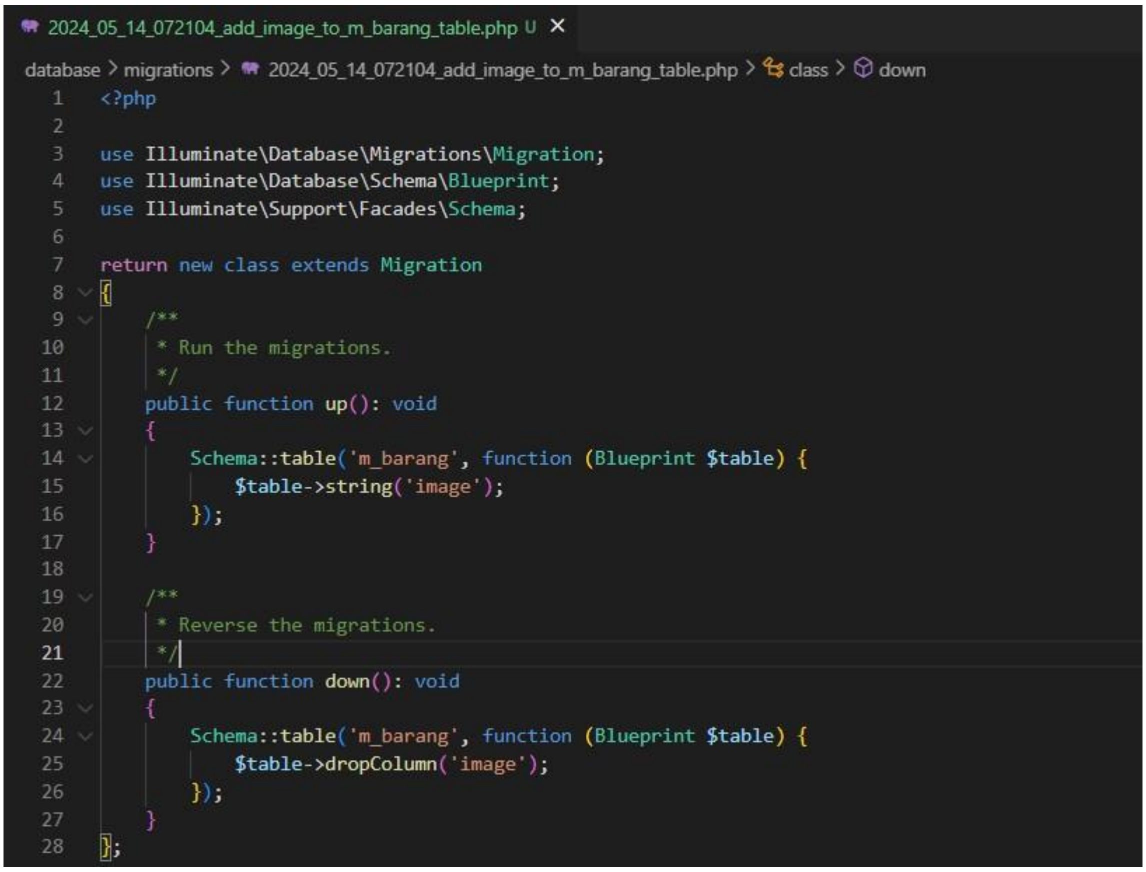
Jawab:

- Kita akan memodifikasi Table m_barang dengan menambahkan column : image, buka terminal lalu ketikkan
  php artisan make:migration add_image_to_m_barang_table



```
PS 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_05_14_072104_add_image_to_m_barang_table.php] created successfully.
```

- Buka file migrasi tersebut lalu modifikasi seperti ini lalu simpan:



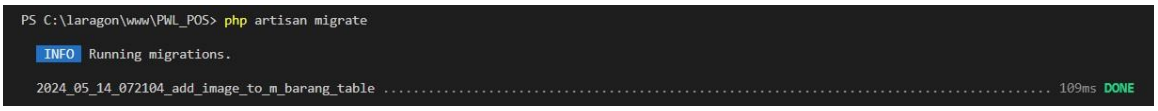
2024_05_14_072104_add_image_to_m_barang_table.php U

database > migrations > 2024_05_14_072104_add_image_to_m_barang_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_barang', function (Blueprint $table) {
            $table->string('image');
        });
    }

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

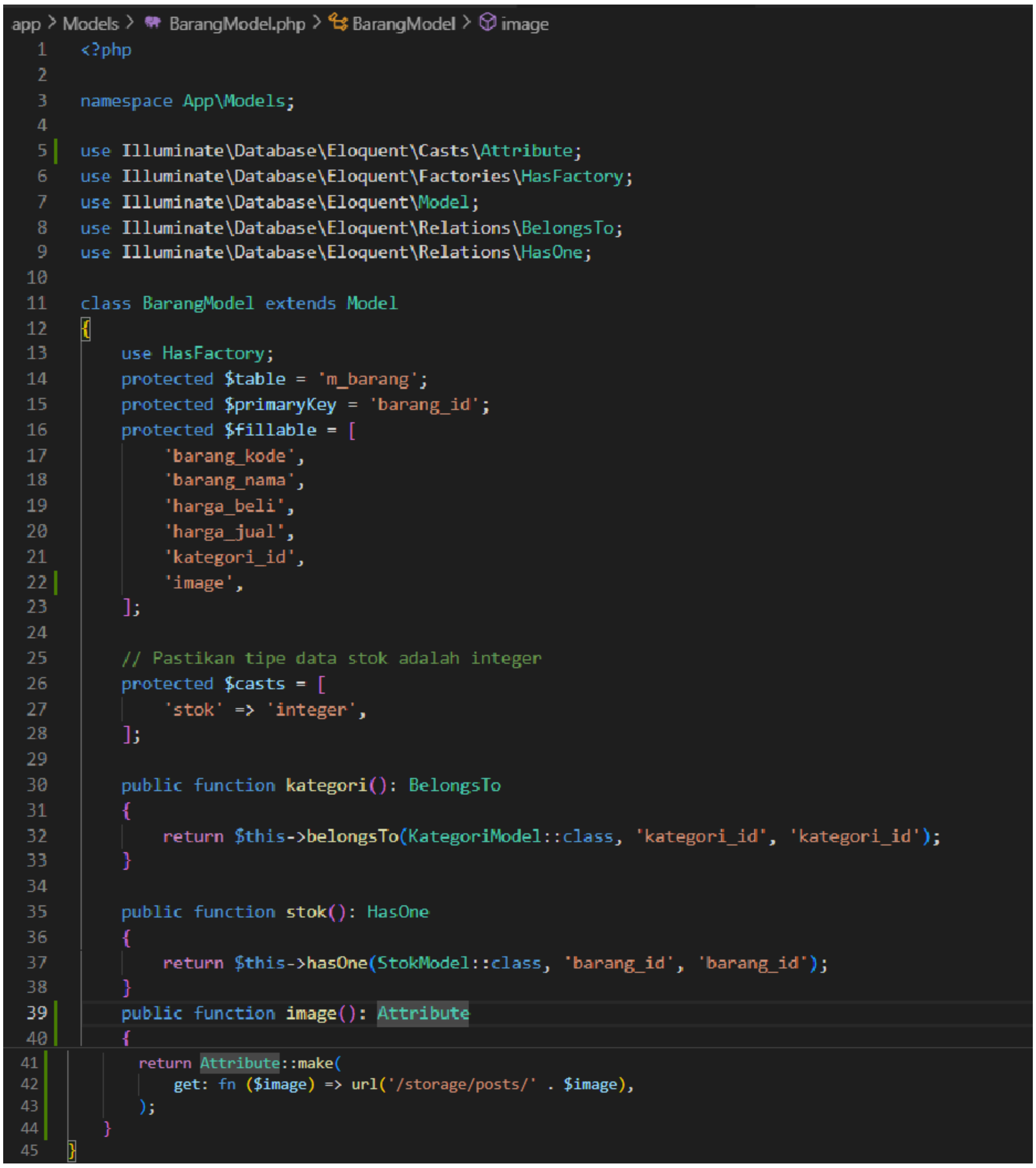
- Lakukan jalankan update migrasi dengan cara:

  php artisan migrate



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

  INFO  Running migrations.

  2024_05_14_072104_add_image_to_m_barang_table ..................................................................... 109ms DONE
```

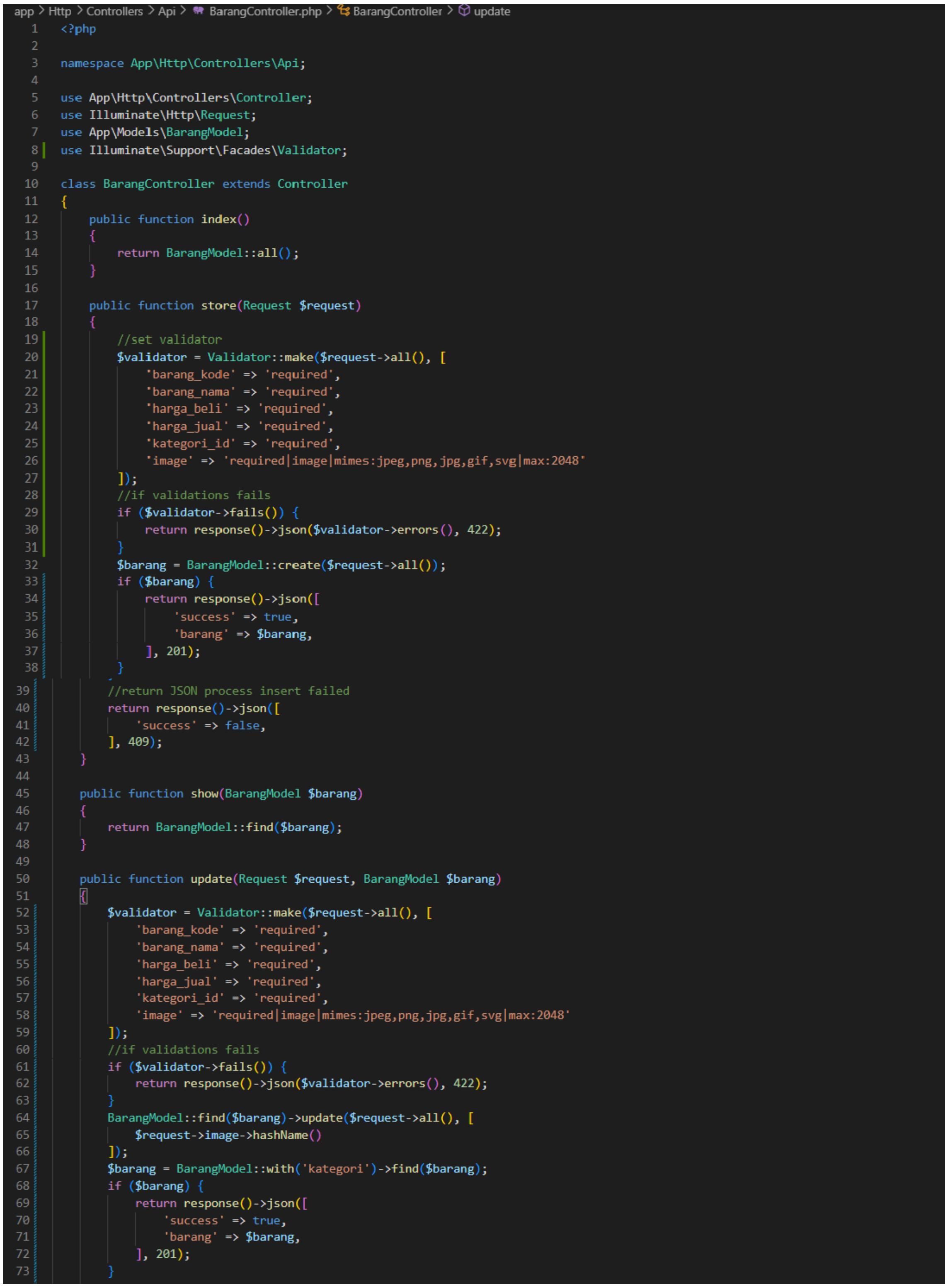
- Lalu lakukan modifikasi models pada App/Models/BarangModel.php



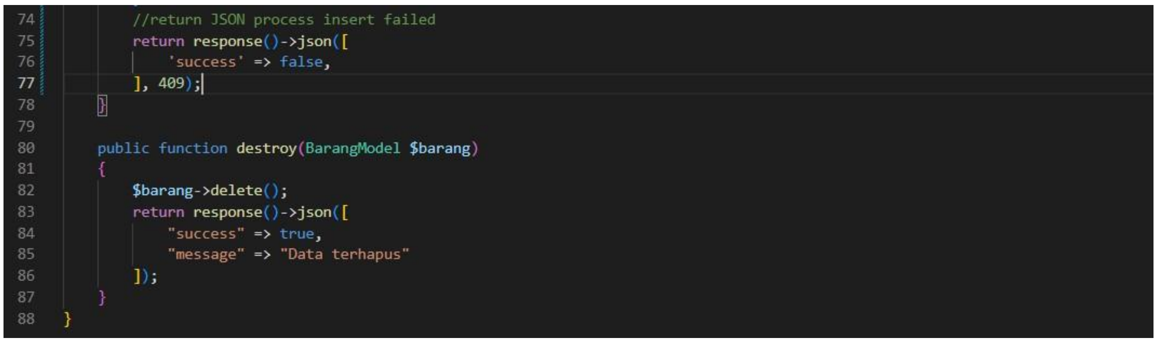
app > Models > BarangModel.php > BarangModel > image

```php
<?php

namespace App\Models;

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

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',
    ];

    // Pastikan tipe data stok adalah integer
    protected $casts = [
        'stok' => 'integer',
    ];

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

    public function stok(): HasOne
    {
        return $this->hasOne(StokModel::class, 'barang_id', 'barang_id');
    }
    public function image(): Attribute
    {
        return Attribute::make(
            get: fn ($image) => url('/storage/posts/' . $image),
        );
    }
}
```

- Lakukan modifikasi pada Controllers/Api/BarangController

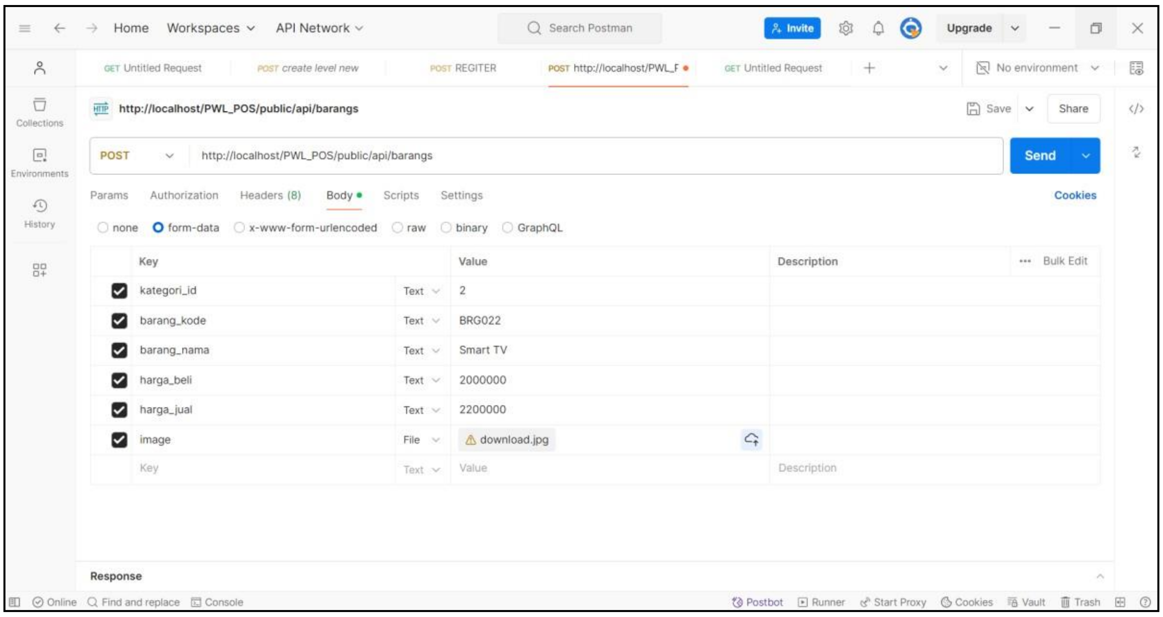

app > Http > Controllers > Api > BarangController.php > BarangController > update

```php
<?php

namespace App\Http\Controllers\Api;

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

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

    public function store(Request $request)
    {
        //set validator
        $validator = Validator::make($request->all(), [
            'barang_kode' => 'required',
            'barang_nama' => 'required',
            'harga_beli' => 'required',
            'harga_jual' => 'required',
            'kategori_id' => 'required',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048'
        ]);
        //if validations fails
        if ($validator->fails()) {
            return response()->json($validator->errors(), 422);
        }
        $barang = BarangModel::create($request->all());
        if ($barang) {
            return response()->json([
                'success' => true,
                'barang' => $barang,
            ], 201);
        }
        //return JSON process insert failed
        return response()->json([
            'success' => false,
        ], 409);
    }

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

    public function update(Request $request, BarangModel $barang)
    {
        $validator = Validator::make($request->all(), [
            'barang_kode' => 'required',
            'barang_nama' => 'required',
            'harga_beli' => 'required',
            'harga_jual' => 'required',
            'kategori_id' => 'required',
            'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048'
        ]);
        //if validations fails
        if ($validator->fails()) {
            return response()->json($validator->errors(), 422);
        }
        BarangModel::find($barang)->update($request->all(), [
            $request->image->hashName()
        ]);
        $barang = BarangModel::with('kategori')->find($barang);
        if ($barang) {
            return response()->json([
                'success' => true,
                'barang' => $barang,
            ], 201);
        }
```



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

    public function destroy(BarangModel $barang)
    {
        $barang->delete();
        return response()->json([
            "success" => true,
            "message" => "Data terhapus"
        ]);
    }
}
```

- Create Barang



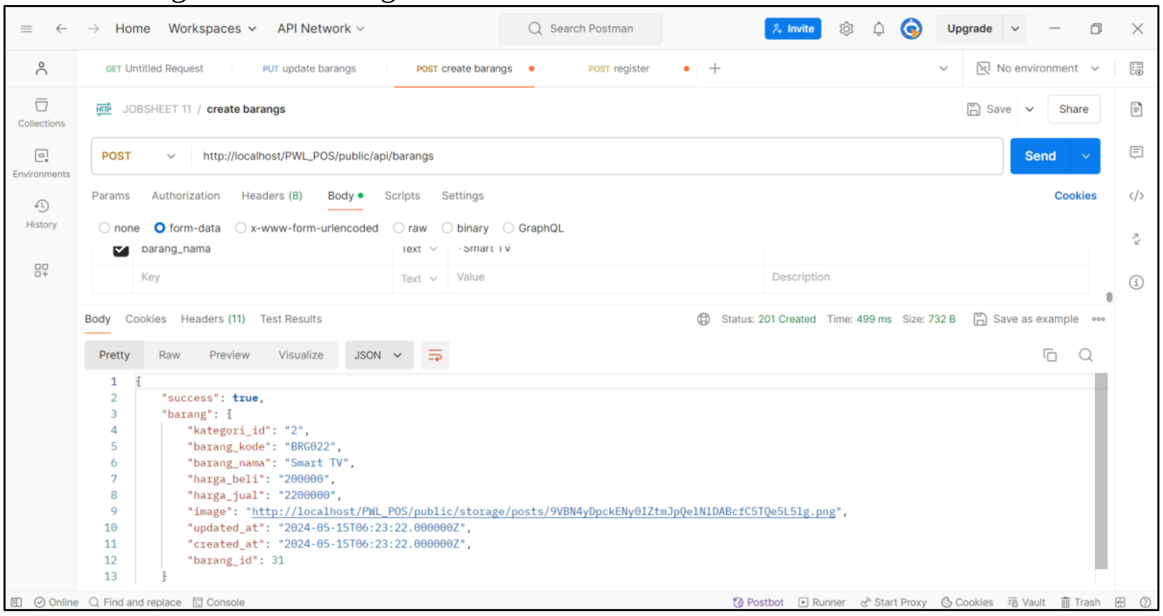
http://localhost/PWL_POS/public/api/barangs

POST http://localhost/PWL_POS/public/api/barangs

Body: form-data

Key		Value	Description
kategori_id	Text	2	
barang_kode	Text	BRG022	
barang_nama	Text	Smart TV	
harga_beli	Text	2000000	
harga_jual	Text	2200000	
image	File	download.jpg	

- Hasil keterangan create barangs sukses atau tidak



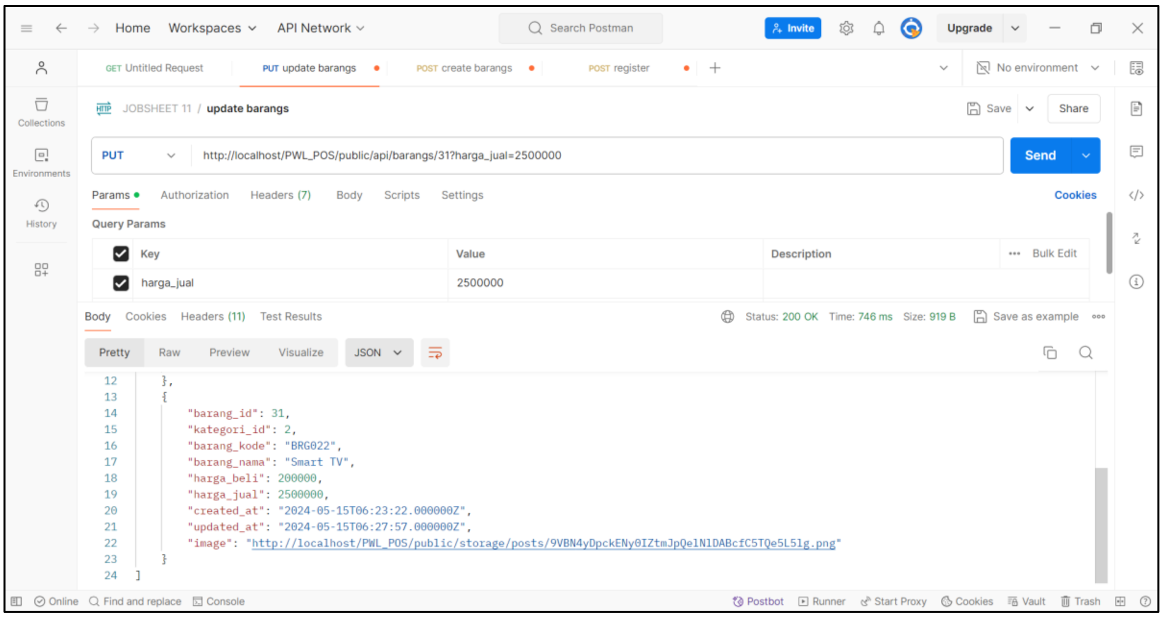
JOBSHEET 11 / create barangs

POST http://localhost/PWL_POS/public/api/barangs

Status: 201 Created Time: 499 ms Size: 732 B

```
{
    "success": true,
    "barang": {
        "kategori_id": "2",
        "barang_kode": "BRG022",
        "barang_nama": "Smart TV",
        "harga_beli": "200000",
        "harga_jual": "2200000",
        "image": "http://localhost/PWL_POS/public/storage/posts/9VBN4yDpckENy0IZtmJpQelNlDABcfC5TQe5L5lg.png",
        "updated_at": "2024-05-15T06:23:22.000000Z",
        "created_at": "2024-05-15T06:23:22.000000Z",
        "barang_id": 31
    }
```

- Update data barang



JOBSHEET 11 / update barangs

PUT http://localhost/PWL_POS/public/api/barangs/31?harga_jual=2500000

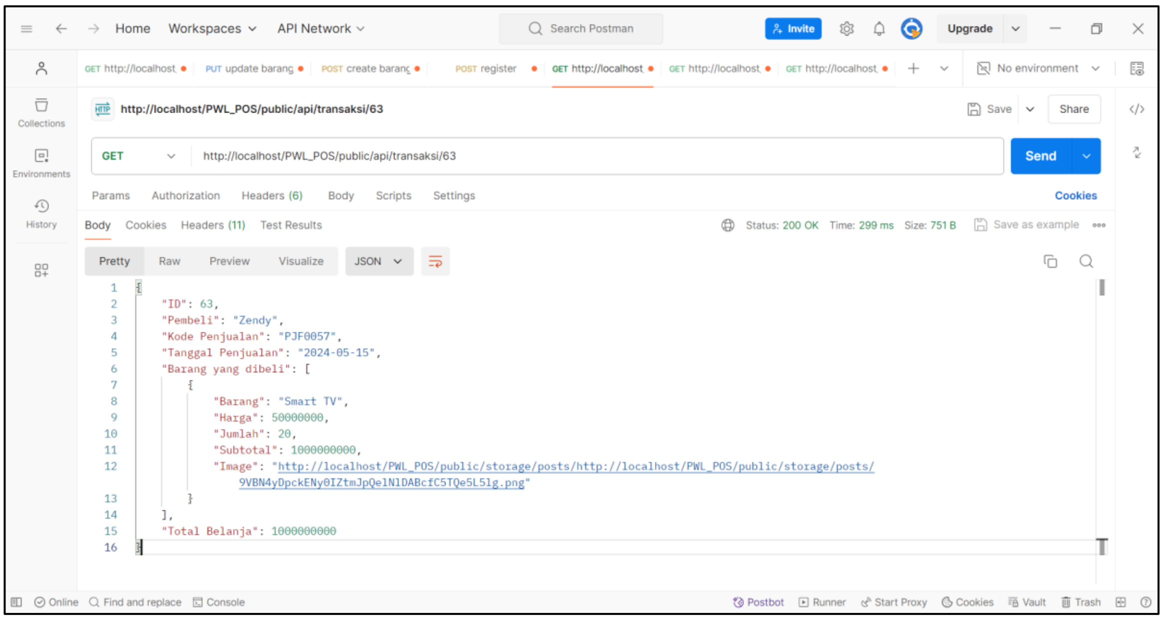
Query Params

Key	Value	Description
harga_jual	2500000	

Status: 200 OK Time: 746 ms Size: 919 B

```
    },
    {
        "barang_id": 31,
        "kategori_id": 2,
        "barang_kode": "BRG022",
        "barang_nama": "Smart TV",
        "harga_beli": 200000,
        "harga_jual": 2500000,
        "created_at": "2024-05-15T06:23:22.000000Z",
        "updated_at": "2024-05-15T06:27:57.000000Z",
        "image": "http://localhost/PWL_POS/public/storage/posts/9VBN4yDpckENy0IZtmJpQelN1DABcfC5TQe5L51g.png"
    }
]
```

- Transaksi



GET http://localhost/PWL_POS/public/api/transaksi/63

Status: 200 OK Time: 299 ms Size: 751 B

```
{
    "ID": 63,
    "Pembeli": "Zendy",
    "Kode Penjualan": "PJF0057",
    "Tanggal Penjualan": "2024-05-15",
    "Barang yang dibeli": [
        {
            "Barang": "Smart TV",
            "Harga": 50000000,
            "Jumlah": 20,
            "Subtotal": 1000000000,
            "Image": "http://localhost/PWL_POS/public/storage/posts/http://localhost/PWL_POS/public/storage/posts/
                9VBN4yDpckENy0IZtmJpQelN1DABcfC5TQe5L51g.png"
        }
    ],
    "Total Belanja": 1000000000
}
```

**** Sekian, dan Terima Kasih****