LAPORAN JOBSHEET 11

RESTFUL API 2

MATA KULIAH PEMROGRAMAN WEB LANJUT

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.

Disusun oleh :

Dahniar Davina SIB-2A / 2341760023

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS

POLITEKNIK NEGERI MALANG

2025

A. 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_2025\Minggu11\Jobsheet11> php artisan make:migration add_im
age_to_m_user_table

  INFO  Migration [C:\laragon\www\PWL_2025\Minggu11\Jobsheet11\database\migratio
ns/2025_04_30_085209_add_image_to_m_user_table.php] created successfully.
```

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

database > migrations > 2025_04_30_085209_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 :

```
php artisan migrate
```

```
PS C:\laragon\www\PWL_2025\Minggu11\Jobsheet11> php artisan migrate

  INFO  Running migrations.

  2025_04_30_085209_add_image_to_m_user_table ..............................  105ms DONE
```

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

app > Models > UserModel.php > UserModel > $fillable

```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
    ];
```

6. Lakukan modifikasi pada Controllers/Api/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);
        }
```

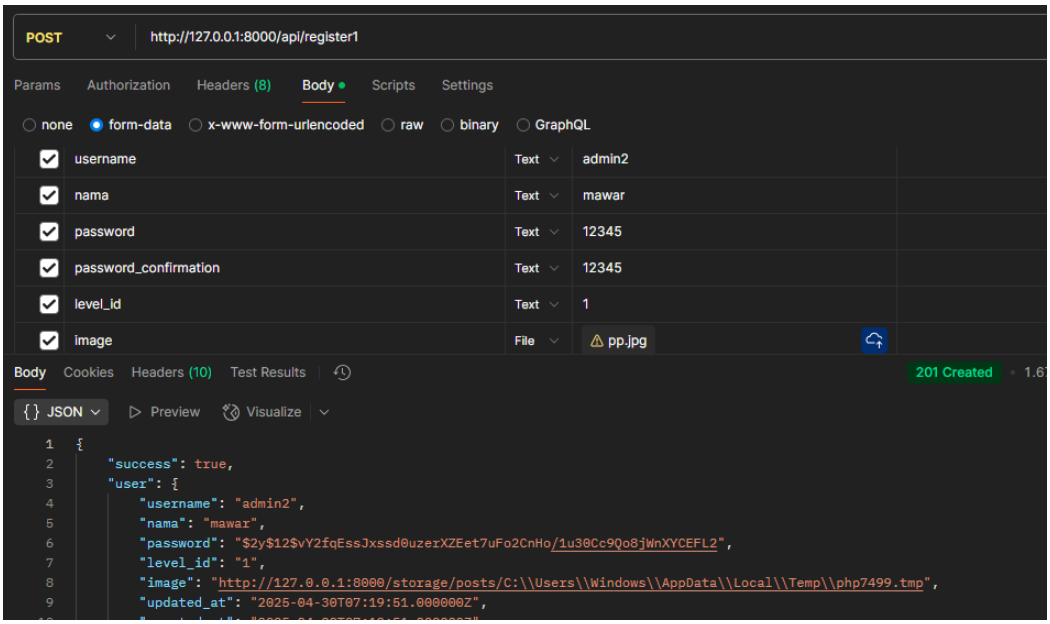
7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

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

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

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

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

11. Uji coba dan screenshot hasilnya apa perbedaan dari yang sebelumnya
    - Jika menggunakan 'image' => $request->image maka file langsung mengambil data yang di-upload tanpa diproses apa pun. Yang tersimpan ke database bukan nama file (contohnya gambar.jpg), tapi object file pathnya.
    - Sedangkan Jika menggunakan 'image' => $request->image->hashName() maka akan mengambil file → simpan dulu ke storage (storage/app/public/users) → baru simpan nama filenya ke database. File gambarnya akan masuk ke folder storage/app/public/users. Di database, hanya akan menyimpan nama file-nya aja, contoh ProfilePicture.jpg, jadi lebih ringan dan aman.

B. 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 diinputkan.

- M_barang

1. Buat migration pada table m_barang

```
  INFO  Migration [C:\laragon\www\PWL_2025\Minggu11\Jobsheet11\database\migrations/2025_04_30_145930_add_image_to_m_barang_table.php] created successfully.
```

2. Ubah isi filenya

database > migrations > 2025_04_30_145930_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');
        });
```

3. Modifikasi isi file pada BarangModel.php

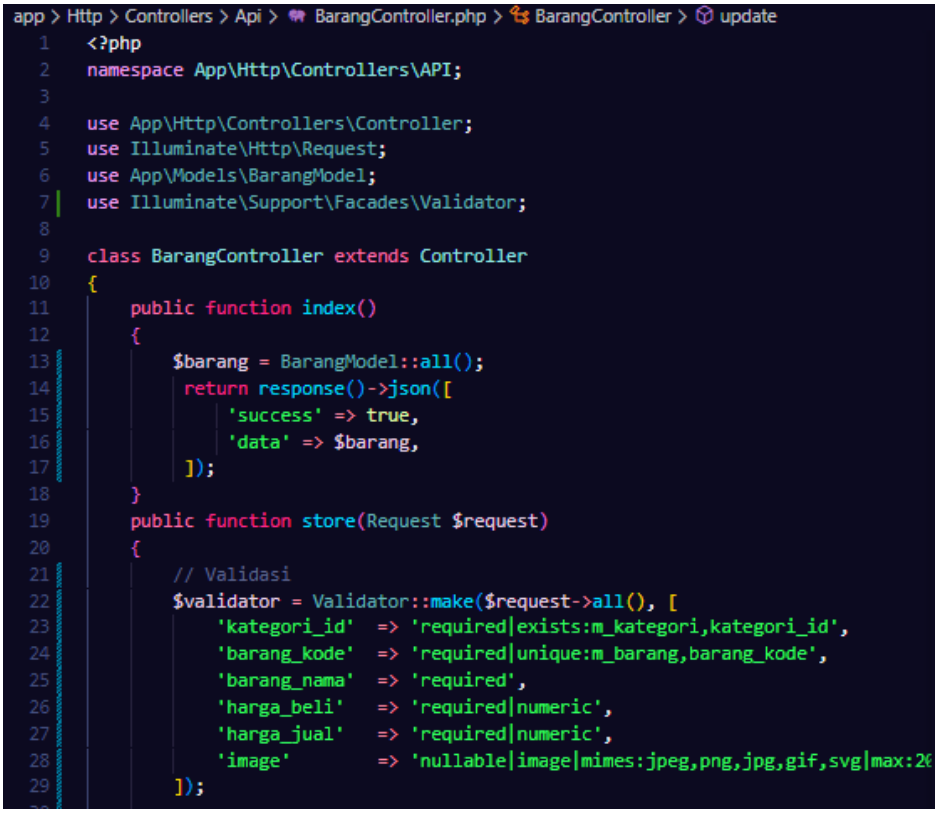
app > Models > BarangModel.php > BarangModel > getJWTIdentifier

```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;
use Tymon\JWTAuth\Contracts\JWTSubject;

class BarangModel extends Model
{
    public function getJWTIdentifier(){
        return $this->getKey();
    }
    public function getJWTCustomClaims(){ return [];
    }
    use HasFactory;
```

4. Modifikasi isi file pada Api/BarangController.php

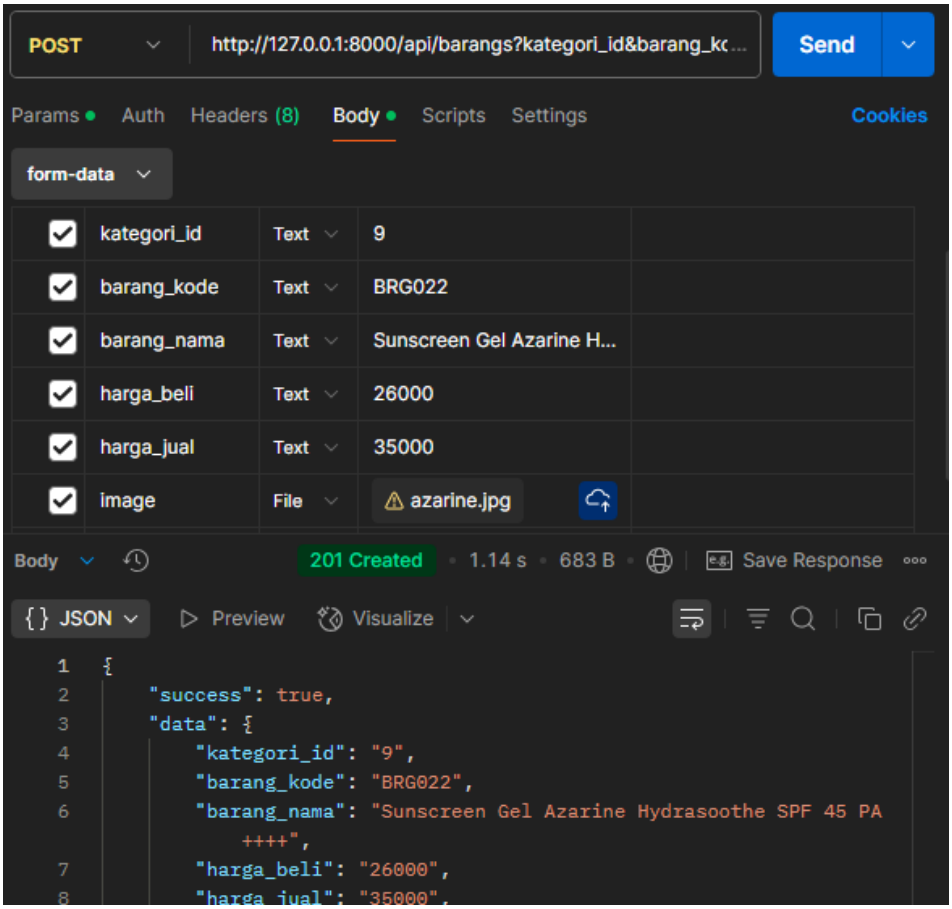


```
app > Http > Controllers > Api > BarangController.php > BarangController > update
<?php
namespace App\Http\Controllers\API;

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()
    {
        $barang = BarangModel::all();
        return response()->json([
            'success' => true,
            'data' => $barang,
        ]);
    }
    public function store(Request $request)
    {
        // Validasi
        $validator = Validator::make($request->all(), [
            'kategori_id'  => 'required|exists:m_kategori,kategori_id',
            'barang_kode'  => 'required|unique:m_barang,barang_kode',
            'barang_nama'  => 'required',
            'harga_beli'   => 'required|numeric',
            'harga_jual'   => 'required|numeric',
            'image'        => 'nullable|image|mimes:jpeg,png,jpg,gif,svg|max:2
        ]);
```

5. Proses pada Postman



➢ M_transaksi

1. Buat Migration pada table t_penjualan

```
PS C:\laragon\www\PWL_2025\Minggu11\Jobsheet11> php artisan make:migration add_image_to_t_penjualan_table

  INFO  Migration [C:\laragon\www\PWL_2025\Minggu11\Jobsheet11\database\migrations/2025_05_03_214345_add_image_to_t_penjualan_table.php] created successfully.
```

2. Ubah isi filenya

```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('t_penjualan', function (Blueprint $table) {
            $table->string('image')->nullable()->after('penjualan_tanggal');
        });
    }

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

3. Buat API/PenjualanController

```
PS C:\laragon\www\PWL_2025\Minggu11\Jobsheet11> php artisan make:migration add_image_

ke:controller Api/PenjualanController

  INFO  Controller [C:\laragon\www\PWL_2025\Minggu11\Jobsheet11\app\Http\Controllers\Api\PenjualanController.php] created successfully.
```

4. Tambahkan isi filenya

```php
<?php

namespace App\Http\Controllers\Api;

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

class PenjualanController extends Controller
{
    public function index()
    {
        $data = PenjualanModel::with(['user', 'details.barang'])->get();
        return response()->json($data);
    }

    public function show($transaksi)
    {
        $penjualan = PenjualanModel::with(['user', 'details.barang'])->findOrFail($transaksi);
        return response()->json($penjualan);
    }

    public function store(Request $request)
    {
        $v = Validator::make($request->all(), [
            'user_id'           => 'required|exists:m_user,user_id',
            'pembeli'           => 'required|string|max:100',
            'penjualan_kode'    => 'required|string|unique:t_penjualan,penjualan_kode',
```

5. Modifikasi isi file pada PenjualanModel.php

app > Models > PenjualanModel.php > PenjualanModel > image

```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\HasMany;
use App\Models\UserModel;
use App\Models\DetailPenjualanModel;
use Illuminate\Database\Eloquent\Casts\Attribute;
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Foundation\Auth\User as Authenticatable;

class PenjualanModel extends Model
{
    public function getJWTIdentifier(){
        return $this->getKey();
    }

    public function getJWTCustomClaims(){
        return [];
    }
```

6. Modifikasi tambahan pada route/api

```php
use App\Http\Controllers\Api\PenjualanController;
```

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

7. Proses pada Postman