Rafif Tri Hartanto

SIB 2A

26

Jobsheet 11 PWL

Praktikum 1: Implementasi Elonquent Accessor

1. REST API

```
2.
     PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:migration add_image_to_m_user_table
      INFO Migration [C:\laragon\www\PML_25\PML_POS\database\migrations/2025_07_11_061324_add_image_to_m_user_table.php] created successfully
3.
   PWL_POS > database > migrations > 🦬 2025_07_11_061324_add_image_to_m_user_table.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_user', function (Blueprint $table) {
                          $table->string('image');
                     });
                public function down(): void
                     Schema::table('m user', function (Blueprint $table) {
                          $table->dropColumn('image');
                     });
     22
```

4.

```
WL_POS > app > Models > 💝 UserModel.php >
     use Illuminate\Database\Eloquent\Factories\HasFactory;
     use Illuminate\Database\Eloquent\Relations\BelongsTo;
     use App\Models\LevelModel;
     use Tymon\JWTAuth\Contracts\JWTSubject;
     use Illuminate\Foundation\Auth\User as Authenticatable;
    use Illuminate\Database\Eloquent\Casts\Attribute;
         use HasFactory;
         public function getJWTIdentifier(){
             return $this->getKey();
         public function getJWTCustomClaims(){
         protected $table = 'm_user';
         protected $primaryKey = 'user_id';
         protected $fillable = [
             'password',
             'nama',
'level_id',
```

6.

```
PWL_POS > app > Http > Controllers > Api > 😁 RegisterController.php > PHP Intelephense > 😘 RegisterController
      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){
               $validator = Validator::make($request->all(), [
                   'nama' => '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,
 33
                   'image' => $request->image
```

```
7.
           Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
           Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('register1');
     24
           Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');
           Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
           Route::middleware('auth:api')->get('/user', function (Request $request) {
               return $request->user();
8.
      Iocalhost/PWL_25/PWL_POS/public/api/register1
                                                                                               □ Save ∨
                                                                                                           Share
               V localhost/PWL_25/PWL_POS/public/api/register1
      Params
             Authorization Headers (8) Body •
                                            Scripts Settings
                                                                                                          Cookies
      ○ none  of form-data  ○ x-www-form-urlencoded  ○ raw  ○ binary  ○ GraphQL
           Key
                                             Value
                                                                               Description
                                                                                                      ON Bulk Edit
        username
                                             penggunaempat
                                      Text ~
        nama
                                             pengguna4
        password
                                      Text ∨
                                             12345
        password_confirmation
                                             12345
                                      Text ~
        level_id
                                      Text ∨
        image

⚠ WhatsApp Image 2025-07-10 at...

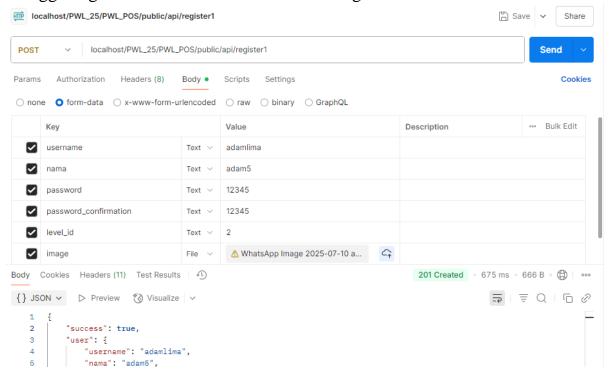
                                      File ∨
     1
            "success": true,
     2
     3
            "user": {
                "username": "penggunaempat",
     4
     5
                "nama": "pengguna4",
                "level_id": "2",
     6
                "image": "http://localhost/PWL_25/PWL_POS/public/storage/postshf31VEDxMkA9ootVADQ9timLB2hxKlXFGuKHBA3Z.
                   png",
                "updated_at": "2025-07-11T06:34:05.000000Z",
     8
                "created_at": "2025-07-11T06:34:05.0000000Z",
                "user_id": 31
    10
    11
    12 }
9.
                        $user = UserModel::create([
                              'username' => $request->username,
                              'nama' => $request->nama,
                              'password' => bcrypt($request->password),
                              'level_id' => $request->level_id,
```

'image' => \$request->image->hashName() //edit

33

1);

10. Fungsi hashName() dipakai untuk membuat nama acak pada file yang diunggah agar tidak mudah ditebak dan meningkatkan keamanan data.



Tugas

1. Buat migrasi add image m_barang

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan make:migration add_image_to_m_barang_table

INFO Migration [C:\laragon\www\PWL_25\PWL_POS\database\migrations/2025_07_11_064307_add_image_to_m_barang_table.php] created successfully.
```

2. Modifikasi migrasi add image m_barang

3. Migrasi

4. Modifikasi BarangModel

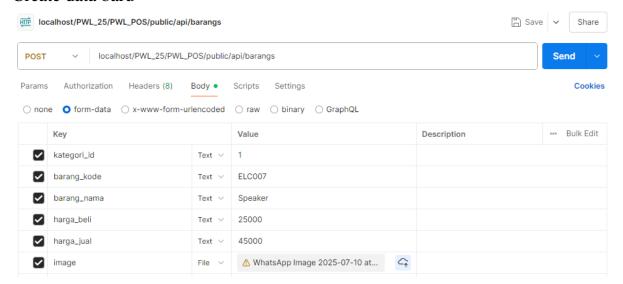
```
PWL_POS > app > Models > 🦬 BarangModel.php > PHP Intelephense > 😭 BarangModel
      namespace App\Models;
      use Illuminate\Database\Eloquent\Factories\HasFactory;
      use Illuminate\Database\Eloquent\Model;
      use Illuminate\Database\Eloquent\Relations\BelongsTo;
  8
     use Illuminate\Database\Eloquent\Casts\Attribute;
      class BarangModel extends Model
          use HasFactory;
          protected $table = "m_barang";
          protected $primaryKey = "barang_id";
          protected $fillable = [
               'kategori_id',
               'barang_kode',
               'barang_nama',
               'harga_beli',
               'harga_jual',
               'image'
 25
          public function kategori(): BelongsTo
               return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
          public function image(): Attribute {
              return Attribute::make(
                   get: fn ($image) => url('/storage/barang/' . $image),
```

5. Modifikasi API/BarangController

```
class BarangController extends Controller
   public function store(Request $request)
       $validator = Validator::make($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:2048'
        if ($validator->fails()) {
            return response()->json($validator->errors(), 422);
       $imageName = null;
        if ($request->hasFile('image')) {
            $imageName = $request->image->hashName();
            $request->image->storeAs('public/barang', $imageName);
       $barang = BarangModel::create([
            'kategori id' => $request->kategori id,
            'barang kode' => $request->barang kode,
            'barang_nama' => $request->barang_nama,
            'harga_beli'
                         => $request->harga beli,
            'harga_jual' => $request->harga_jual,
            'image'
                          => $imageName,
        ]);
       return response()->json([
```

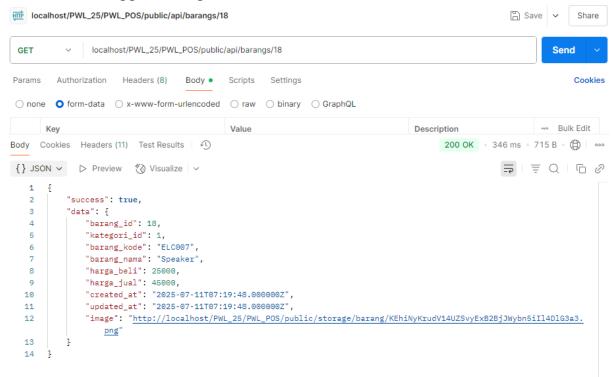
6. Tes di postman

Create data baru



```
{} JSON ✓ ▷ Preview 🍪 Visualize ✓
                                                                                                   = 1 = Q 1 G
   1
           "success": true,
           "data": {
              "kategori_id": "1",
              "barang_kode": "ELC007",
   5
              "barang_nama": "Speaker",
              "harga_beli": "25000",
              "harga_jual": "45000",
   8
              "image": "http://localhost/PWL_25/PWL_POS/public/storage/barang/KEhiNyKrudV14UZSvyExB2BjJWybn5iIl4DlG3a3.
                 png",
              "updated_at": "2025-07-11T07:19:48.000000Z",
  10
              "created_at": "2025-07-11T07:19:48.0000002",
  11
  12
              "barang_id": 18
  13
  14 }
```

7. Lihat data menggunakan get



8. Hasil di phpMyAdmin

• Sebelum

←T	_→		~	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at	image
	Edit	≩ е́ Сору	Delete	3	2	FSH001	T-shirt	100000	200000	NULL	NULL	NULL
		∄ е́ Сору	Delete	4	2	FSH002	Jeans	200000	400000	NULL	NULL	NULL
	Ø Edit	≩ е́ Сору	Delete	5	3	BOK001	Novel	50000	100000	NULL	NULL	NULL
	Edit	∄ е́ Сору	Delete	6	3	BOK002	Comic Book	40000	80000	NULL	NULL	NULL
	Ø Edit	≩ е́ Сору	Delete	7	4	HTH001	Shampoo	25000	300	NULL	NULL	NULL
	Edit	≩ е́ Сору	Delete	8	4	HTH002	Toothpaste	10000	20000	NULL	NULL	NULL
		≩ е́ Сору	Delete	9	5	FDM001	Coffee	20000	40000	NULL	NULL	NULL
	<i></i>	≩ е́ Сору	Delete	10	5	FDM002	Tea	15000	30000	NULL	NULL	NULL
	Ø Edit	≩ е́ Сору	Delete	11	1	ELC003	Charger	60000	90000	2025-07-03 05:28:07	2025-07-03 05:31:00	NULL
	<i> </i>	≩ е́ Сору	Delete	12	1	ELC004	Headphone	25000	50000	2025-07-09 04:56:33	NULL	NULL
	<i> ⊗</i> Edit	≩ в Сору	Delete	13	2	FSH003	Jacket	20000	40000	2025-07-09 04:56:33	NULL	NULL
	<i> ⊗</i> Edit	≩ сору	Delete	14	3	BOK003	Majalah	15000	20000	2025-07-09 04:56:33	NULL	NULL
	Edit	≩ в Сору	Delete	15	4	HTH003	Sabun	5000	8000	2025-07-09 04:56:33	NULL	NULL
	Ø Edit	≩ сору	Delete	16	5	FDM003	Soda	6000	9000	2025-07-09 04:56:33	NULL	NULL
	Edit	≩ в Сору	Delete	17	1	ELC006	Monitor	20000	40000	2025-07-11 06:55:08	2025-07-11 06:55:08	NULL

• Sesudah

barang_id 9	-	barang_kode FDM001	barang_nama Coffee	harga_beli 20000	harga_jual 40000	created_at NULL	updated_at NULL	image NULL
10	5	FDM002	Tea	15000	30000	NULL	NULL	NULL
11	1	ELC003	Charger	60000	90000	2025-07-03 05:28:07	2025-07-03 05:31:00	NULL
12	1	ELC004	Headphone	25000	50000	2025-07-09 04:56:33	NULL	NULL
13	2	FSH003	Jacket	20000	40000	2025-07-09 04:56:33	NULL	NULL
14	3	BOK003	Majalah	15000	20000	2025-07-09 04:56:33	NULL	NULL
15	4	HTH003	Sabun	5000	8000	2025-07-09 04:56:33	NULL	NULL
16	5	FDM003	Soda	6000	9000	2025-07-09 04:56:33	NULL	NULL
17	1	ELC006	Monitor	20000	40000	2025-07-11 06:55:08	2025-07-11 06:55:08	NULL
18	1	ELC007	Speaker	25000	45000	2025-07-11 07:19:48	2025-07-11 07:19:48	KEhiNyKrudV14UZSvyExB2BjJWybn5ill4DlG3a3.png