

Rafif Tri Hartanto

SIB 2A

26

Jobsheet 11 PWL

Praktikum 1: Implementasi Eloquent 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\PWL_25\PWL_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 > ...  
1  <?php  
2  
3  use Illuminate\Database\Migrations\Migration;  
4  use Illuminate\Database\Schema\Blueprint;  
5  use Illuminate\Support\Facades\Schema;  
6  
7  return new class extends Migration  
8  {  
9      public function up(): void  
10     {  
11         Schema::table('m_user', function (Blueprint $table) {  
12             $table->string('image');  
13         });  
14     }  
15  
16     public function down(): void  
17     {  
18         Schema::table('m_user', function (Blueprint $table) {  
19             $table->dropColumn('image');  
20         });  
21     }  
22 };
```

4.

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan migrate --path=/database/migrations/2025_07_11_061324_add_image_to_m_user_table.php  
[INFO] Running migrations.  
2025_07_11_061324_add_image_to_m_user_table ..... 1,203ms DONE
```

5.

```
PWL_POS > app > Models > UserModel.php > ...
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\Relations\BelongsTo;
8  use App\Models\LevelModel;
9  use Tymon\JWTAuth\Contracts\JWTSubject;
10 use Illuminate\Foundation\Auth\User as Authenticatable;
11 use Illuminate\Database\Eloquent\Casts\Attribute;
12
13 class UserModel extends Authenticatable implements JWTSubject
14 {
15     use HasFactory;
16
17     public function getJWTIdentifier(){
18         return $this->getKey();
19     }
20
21     public function getJWTCustomClaims(){
22         return [];
23     }
24
25     protected $table = 'm_user';
26     protected $primaryKey = 'user_id';
27
28     protected $fillable = [
29         'username',
30         'password',
31         'nama',
32         'level_id',
33         'created_at',
```

6.

```
PWL_POS > app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Models\UserModel;
6  use App\Http\Controllers\Controller;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Validator;
9
10 class RegisterController extends Controller
11 {
12     public function __invoke(Request $request){
13         //set validation
14         $validator = Validator::make($request->all(), [
15             'username' => 'required',
16             'nama' => 'required',
17             'password' => 'required|min:5|confirmed',
18             'level_id' => 'required',
19             'image' => 'required',
20         ]);
21
22         //if validations fails
23         if ($validator->fails()) {
24             return response()->json($validator->errors(), 422);
25         }
26
27         //create user
28         $user = UserModel::create([
29             'username' => $request->username,
30             'nama' => $request->nama,
31             'password' => bcrypt($request->password),
32             'level_id' => $request->level_id,
33             'image' => $request->image
```

7.

```
23 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
24 Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('register1');
25 Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');
26 Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
27 Route::middleware('auth:api')->get('/user', function (Request $request) {
28     return $request->user();
29 });
```

8.

localhost/PWL_25/PWL_POS/public/api/register1 Save Share

POST localhost/PWL_25/PWL_POS/public/api/register1 Send

Params Authorization Headers (8) **Body** Scripts Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key		Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	username	Text	penggunaempat			
<input checked="" type="checkbox"/>	nama	Text	pengguna4			
<input checked="" type="checkbox"/>	password	Text	12345			
<input checked="" type="checkbox"/>	password_confirmation	Text	12345			
<input checked="" type="checkbox"/>	level_id	Text	2			
<input checked="" type="checkbox"/>	image	File	WhatsApp Image 2025-07-10 at...			

```
1 {
2     "success": true,
3     "user": {
4         "username": "penggunaempat",
5         "nama": "pengguna4",
6         "level_id": "2",
7         "image": "http://localhost/PWL_25/PWL_POS/public/storage/postshf31VEDxMkA9ootVADQ9timLB2hxK1XFGuKHBA3Z.
8         png",
9         "updated_at": "2025-07-11T06:34:05.000000Z",
10        "created_at": "2025-07-11T06:34:05.000000Z",
11        "user_id": 31
12    }
13 }
```

9.

```
28 $user = UserModel::create([
29     'username' => $request->username,
30     'nama' => $request->nama,
31     'password' => bcrypt($request->password),
32     'level_id' => $request->level_id,
33     'image' => $request->image->hashName() //edit
34 ]);
```

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

localhost/PWL_25/PWL_POS/public/api/register1

POST localhost/PWL_25/PWL_POS/public/api/register1

Params Authorization Headers (8) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	username	Text adamlima		
<input checked="" type="checkbox"/>	nama	Text adam5		
<input checked="" type="checkbox"/>	password	Text 12345		
<input checked="" type="checkbox"/>	password_confirmation	Text 12345		
<input checked="" type="checkbox"/>	level_id	Text 2		
<input checked="" type="checkbox"/>	image	File WhatsApp Image 2025-07-10 a...		

Body Cookies Headers (11) Test Results

201 Created • 675 ms • 666 B

JSON Preview Visualize

```
1 {
2   "success": true,
3   "user": {
4     "username": "adamlima",
5     "nama": "adam5",
```

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

```
PWL_POS > database > migrations > 2025_07_11_064307_add_image_to_m_barang_table.php > ..
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::table('m_barang', function (Blueprint $table) {
15             $table->string('image')->nullable();
16         });
17     }
18
19     /**
20      * Reverse the migrations.
21      */
22     public function down(): void
23     {
24         Schema::table('m_barang', function (Blueprint $table) {
25             $table->dropColumn('image');
26         });
27     }
28 };
```

3. Migrasi

```
PS C:\laragon\www\PWL_25\PWL_POS> php artisan migrate --path=/database/migrations/2025_07_11_064307_add_image_to_m_barang_table.php
```

```
INFO Running migrations.
```

```
2025_07_11_064307_add_image_to_m_barang_table ..... 73ms DONE
```

4. Modifikasi BarangModel

```
PWL_POS > app > Models > BarangModel.php > PHP Intelephense > BarangModel
```

```
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7  use Illuminate\Database\Eloquent\Relations\BelongsTo;
8  use Illuminate\Database\Eloquent\Casts\Attribute;
9
10 class BarangModel extends Model
11 {
12     use HasFactory;
13
14     protected $table = "m_barang";
15     protected $primaryKey = "barang_id";
16     protected $fillable = [
17         'kategori_id',
18         'barang_kode',
19         'barang_nama',
20         'harga_beli',
21         'harga_jual',
22         'image'
23     ];
24
25     public function kategori(): BelongsTo
26     {
27         return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
28     }
29
30     public function image(): Attribute {
31         return Attribute::make(
32             get: fn ($image) => url('/storage/barang/' . $image),
33         );
34     }
35 }
```

5. Modifikasi API/BarangController

```
10 class BarangController extends Controller
20
21 public function store(Request $request)
22 {
23     $validator = Validator::make($request->all(), [
24         'kategori_id' => 'required|exists:m_kategori,kategori_id',
25         'barang_kode' => 'required|unique:m_barang,barang_kode',
26         'barang_nama' => 'required',
27         'harga_beli' => 'required|numeric',
28         'harga_jual' => 'required|numeric',
29         'image' => 'nullable|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
30     ]);
31
32     if ($validator->fails()) {
33         return response()->json($validator->errors(), 422);
34     }
35
36     $imageName = null;
37     if ($request->hasFile('image')) {
38         $imageName = $request->image->hashName();
39         $request->image->storeAs('public/barang', $imageName);
40     }
41
42     $barang = BarangModel::create([
43         'kategori_id' => $request->kategori_id,
44         'barang_kode' => $request->barang_kode,
45         'barang_nama' => $request->barang_nama,
46         'harga_beli' => $request->harga_beli,
47         'harga_jual' => $request->harga_jual,
48         'image' => $imageName,
49     ]);
50
51     return response()->json([
```

6. Tes di postman Create data baru

localhost/PWL_25/PWL_POS/public/api/barangs

POST localhost/PWL_25/PWL_POS/public/api/barangs Send

Params Authorization Headers (8) Body Scripts Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key		Value	Description	
<input checked="" type="checkbox"/>	kategori_id	Text	1		
<input checked="" type="checkbox"/>	barang_kode	Text	ELC007		
<input checked="" type="checkbox"/>	barang_nama	Text	Speaker		
<input checked="" type="checkbox"/>	harga_beli	Text	25000		
<input checked="" type="checkbox"/>	harga_jual	Text	45000		
<input checked="" type="checkbox"/>	image	File	WhatsApp Image 2025-07-10 at...		

```

1  {
2      "success": true,
3      "data": {
4          "kategori_id": "1",
5          "barang_kode": "ELC007",
6          "barang_nama": "Speaker",
7          "harga_beli": "25000",
8          "harga_jual": "45000",
9          "image": "http://localhost/PWL_25/PWL_POS/public/storage/barang/KEhiNyKrudV14UZSvyExB2BjJWybn5iI14D1G3a3.png",
10         "updated_at": "2025-07-11T07:19:48.000000Z",
11         "created_at": "2025-07-11T07:19:48.000000Z",
12         "barang_id": 18
13     }
14 }

```

7. Lihat data menggunakan get

localhost/PWL_25/PWL_POS/public/api/barangs/18

GET localhost/PWL_25/PWL_POS/public/api/barangs/18

Params Authorization Headers (8) Body Scripts Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description	Bulk Edit
Body			

200 OK • 346 ms • 715 B

```

1  {
2      "success": true,
3      "data": {
4          "barang_id": 18,
5          "kategori_id": 1,
6          "barang_kode": "ELC007",
7          "barang_nama": "Speaker",
8          "harga_beli": 25000,
9          "harga_jual": 45000,
10         "created_at": "2025-07-11T07:19:48.000000Z",
11         "updated_at": "2025-07-11T07:19:48.000000Z",
12         "image": "http://localhost/PWL_25/PWL_POS/public/storage/barang/KEhiNyKrudV14UZSvyExB2BjJWybn5iI14D1G3a3.png"
13     }
14 }

```

8. Hasil di phpMyAdmin

- Sebelum

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

- Sesudah

barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at	image
9	5	FDM001	Coffee	20000	40000	NULL	NULL	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	KEhiNyKrudV14UZSvyExB2BjJWybn5ill4DIG3a3.png