

**LAPORAN JOBSHEET 11**  
**PEMROGRAMAN WEB LANJUT**  
**RESTFUL API 2**

Dosen pengajar: Dimas Wahyu Wibowo, ST., MT



Disusun Oleh:

Nama : M. Fatih Al Ghifary

NIM : 2341720194

Kelas : TI-2A

No : 16

**PRODI TEKNIK INFORMATIKA D-IV**  
**JURUSAN TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI MALANG**

**2025**

## 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*

```
D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 11\PWL_POS>php artisan make:migration add_image_to_m_user_table
[INFO] Migration [D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 11\PWL_POS\database\Migrations\2025_04_26_230634_add_image_to_m_user_table.php] created successfully.
```

3. Modifikasi file migration

```
<?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): void {
            $table->string('image');
        });
    }

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

4. Php artisan migrate

```
D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 11\PWL_POS>php artisan migrate --path=/database/migrations/2025_04_26_230634_add_image_to_m_user_table.php
[INFO] Running migrations.

2025_04_26_230634_add_image_to_m_user_table ..... 36ms DONE
```

5. Modifikasi models pada App/Models/UserModel.php

```
use Illuminate\Database\Eloquent\Casts\Attribute;
```

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

6. Modifikasi pada Controllers/Api/RegisterController

```
class RegisterController extends Controller
{
    0 references | 0 overrides
    public function __invoke(Request $request): JsonResponse|mixed
    {
        //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 validations fails
        if ($validator->fails()) {
            return response()->json($validator->errors(), 422);
        }

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

        //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);
    }
}
```

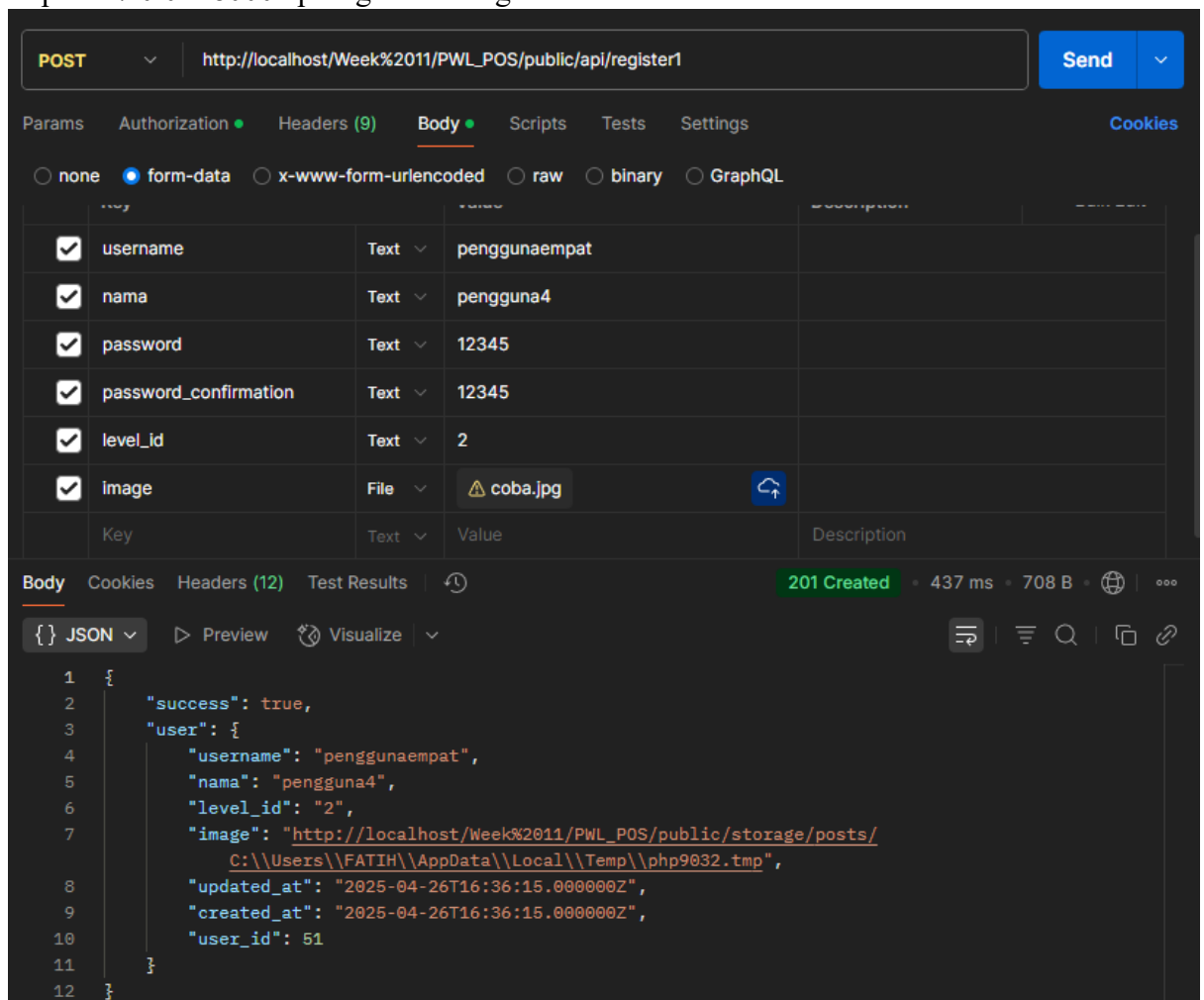
- Anda dapat menambahkan detail untuk spesifikasi image pada validator

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

- Tambahkan register1 pada routes/api.php

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

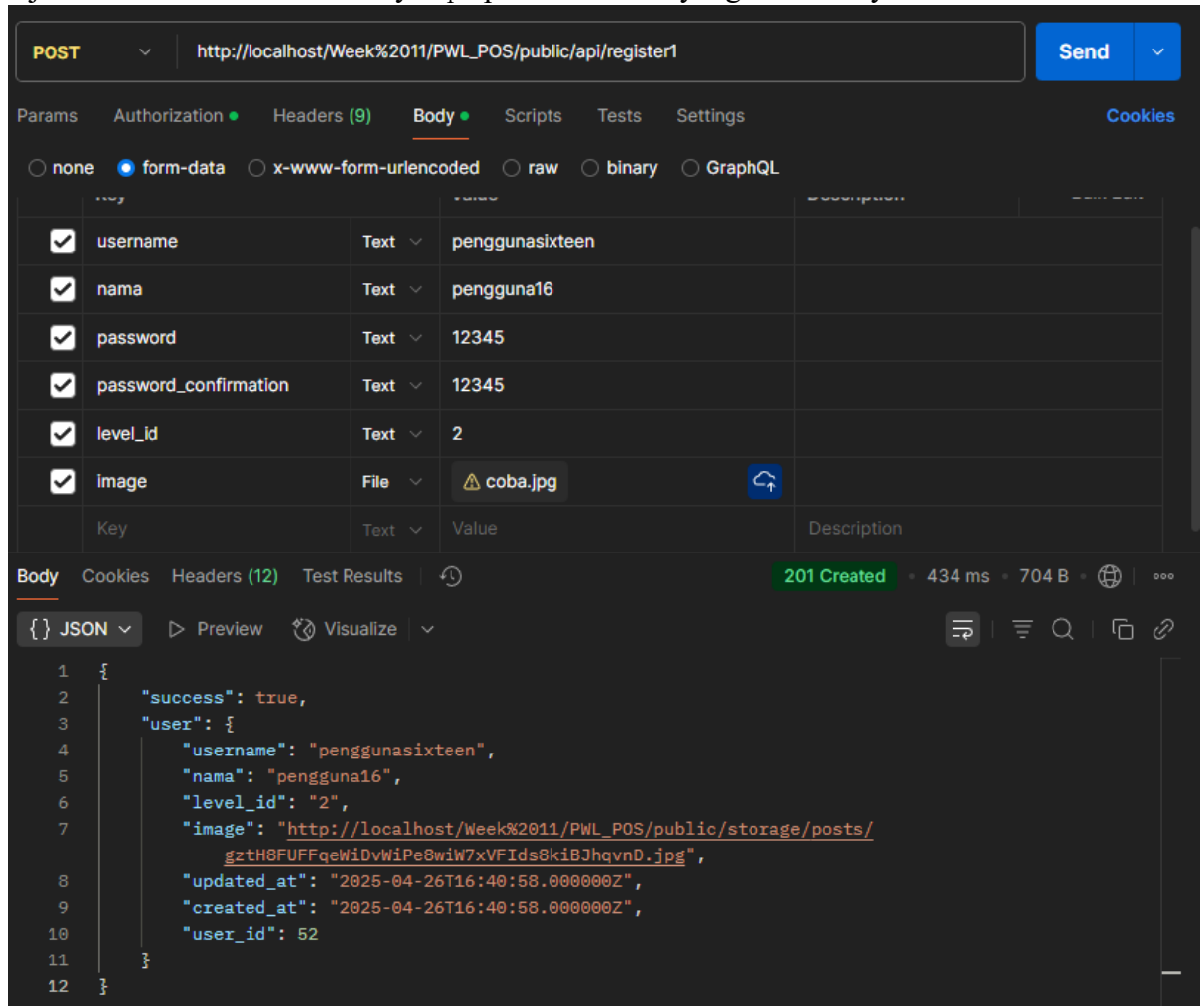
- Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar  
http://127.0.0.1:8000/api/register1 dengan method POST dan klik send



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

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

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



Bedanya ketika menggunakan `hashName()` maka nama dari file yang kita send akan ter-hash atau diacak bisa dikatakan. Sedangkan kalau biasa, mendirect ke penyimpanan/direktori dimana kita menyimpan fotonya.

12. Commit

<https://github.com/fateehhh/PROWEBLNJT/commit/5f1a3b8fdb1b50ce84e05e24f626ad876ef5a8b>

## 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.

- Buat Migration

```
D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 11\PWL_POS>php artisan make:migration add_image_to_m_barang_table
INFO Migration [D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 11\PWL_POS\database\migrations\2025_04_26_234615_add_image_to_m_barang_table.php] created successfully.
```

- Modifikasi isi file Migration

```
<?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): void {
            $table->string('image')->after('barang_nama')->nullable();
        });
    }

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

- Php artisan migrate --path=blablabla (agar migrate secara spesifik)

```
D:\KULIAH\SEMESTER-4\file\laragon\www\PROWEBLNJT\Week 11\PWL_POS>php artisan migrate --path=/database/migrations/2025_04_26_234615_add_image_to_m_barang_table.php

INFO Running migrations.

2025_04_26_234615_add_image_to_m_barang_table ..... 14ms DONE
```

- Ubah BarangModel.php dengan menambahkan 'image' ke \$fillable dan menambahkan function image

```
0 references
protected $table = 'm_barang';
0 references
protected $primaryKey = 'barang_id';
0 references
protected $fillable = ['barang_id', 'kategori_id', 'barang_kode', 'barang_nama', 'image', 'harga_beli', 'harga_jual'];

0 references | 0 overrides
public function kategori(): BelongsTo
{
    return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
}

0 references | 0 overrides
protected function image(): Attribute
{
    return Attribute::make(
        get: fn($image): string => asset(path: 'storage/barang/' . $image),
    );
}
```

- Modifikasi pada api/BarangController di dalam function store

```

public function store(Request $request): JsonResponse|mixed
{
    $validator = Validator::make($request->all(), [
        'kategori_id' => 'required',
        'barang_kode' => 'required',
        'barang_nama' => 'required',
        'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        'harga_beli' => 'required',
        'harga_jual' => 'required',
    ]);

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

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

    if ($barang) {
        return response()->json([
            'success' => true,
            'barang' => $barang,
        ], 201);
    }

    return response()->json([
        'success' => false,
    ], 409);
}

```

- Percobaan

**POST** `http://localhost/Week%2011/PWL_POS/public/api/barangs` **Send**

Params Authorization Headers (9) **Body** Scripts Tests Settings Cookies

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

	Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	kategori_id	Text 13		
<input checked="" type="checkbox"/>	barang_kode	Text BRG019		
<input checked="" type="checkbox"/>	barang_nama	Text Moisturizer		
<input checked="" type="checkbox"/>	image	File moisturizer.jpg		
<input checked="" type="checkbox"/>	harga_beli	Text 19500		
<input checked="" type="checkbox"/>	harga_jual	Text 27000		

**Body** Cookies Headers (12) Test Results 201 Created • 229 ms • 757 B

**JSON** Preview Visualize

```

1 {
2   "success": true,
3   "barang": {
4     "kategori_id": "13",
5     "barang_kode": "BRG019",
6     "barang_nama": "Moisturizer",
7     "image": "http://localhost/Week%2011/PWL_POS/public/storage/barang/
8       mu20GabbFYWuuTKrVuzyR0sgTCgTZLf6cWmMxAN6.jpg",
9     "harga_beli": "19500",
10    "harga_jual": "27000",
11    "updated_at": "2025-04-26T17:00:16.000000Z",
12    "created_at": "2025-04-26T17:00:16.000000Z",
13    "barang_id": 31
14  }
15 }

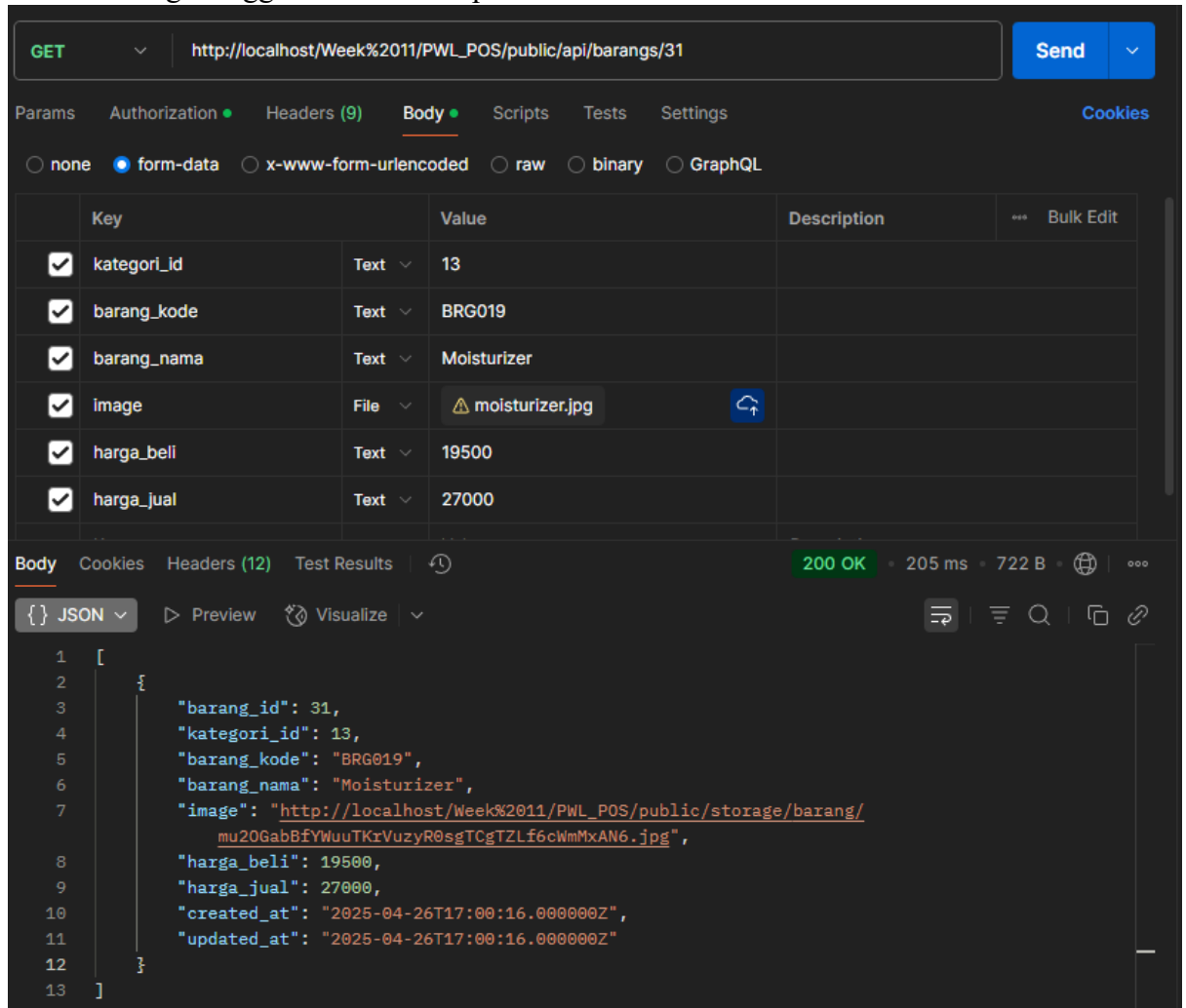
```

Edit Copy Delete 31 13 BRG019 Moisturizer mu20GabbFYWuuTKrVuzyR0sgTCgTZLf6cWmMxAN6.jpg 19500 27000 2025-04-27 00:00:16 2025-04-27 00:00:16

tersimpan di database dengan nama image yang sudah ter-hush



- Detail Barang menggunakan GET di postman



- Commit  
<https://github.com/fateehhh/PROWEBLNJT/commit/2c991acfb19ff66eba9f72011783062a5115444a>