



Jurusan Teknologi Informasi Politeknik Negeri Malang  
JOBSHEET 11 RESTFUL API 2  
Mata Kuliah Pemrograman Web Lanjut (PWL)

Nama	:	Candra Ahmad Dani
Nim	:	2341720187
Kelas	:	TI-2A

### 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

PWL\_2025 > week10 > PWL\_POS > database > migrations > 2025\_04\_13\_052550\_add\_foto\_m\_user\_table.php

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

```
<?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('foto')->nullable();
        });
    }

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

4. Lakukan jalankan update migrasi dengan cara:  
php artisan migrate



Jurusan Teknologi Informasi Politeknik Negeri Malang  
JOBSHEET 11 RESTFUL API 2  
Mata Kuliah Pemrograman Web Lanjut (PWL)

5. Lalu lakukan modifikasi models pada 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 Illuminate\Foundation\Auth\User as Authenticatable; // implementasi class Authenticatable
9 use Tymon\JWTAuth\Contracts\JWTSubject;
10
11 class UserModel extends Authenticatable implements JWTSubject
12 {
13     public function getJWTIdentifier()
14     {
15         return $this->getKey();
16     }
17     public function getJWTCustomClaims()
18     {
19         return [];
20     }
21     use HasFactory;
22
23     protected $table = 'm_user'; //Mendefinisikan nama tabel yang digunakan oleh model
24     protected $primaryKey = 'user_id'; //Mendefinisikan primary key dari tabel yang digunakan
25
26     protected $fillable = [
27         'username',
28         'nama',
29         'password',
30         'level_id',
31         'foto'
32     ];
33
34     protected $hidden = ['password']; // jangan ditampilkan saat di select
35     protected $casts = ['password' => 'hashed']; // casting password agar otomatis di hash
36
37     public function level(): BelongsTo
38     {
39         return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
40     }
41
42     // Mendapatkan nama role
43     public function getRoleName(): string
44     {
45         return $this->level->level_nama;
46     }
47
48     // cek apakah user memiliki role tertentu
49     public function hasRole($role): bool
50     {
51         return $this->level->level_kode == $role;
52     }
53
54     // mendapatkan kode role
55     public function getRole()
56     {
57         return $this->level->level_kode;
58     }
59 }
```



Jurusan Teknologi Informasi Politeknik Negeri Malang  
JOBSHEET 11 RESTFUL API 2  
Mata Kuliah Pemrograman Web Lanjut (PWL)

6. Lakukan modifikasi pada Controllers/Api/RegisterController

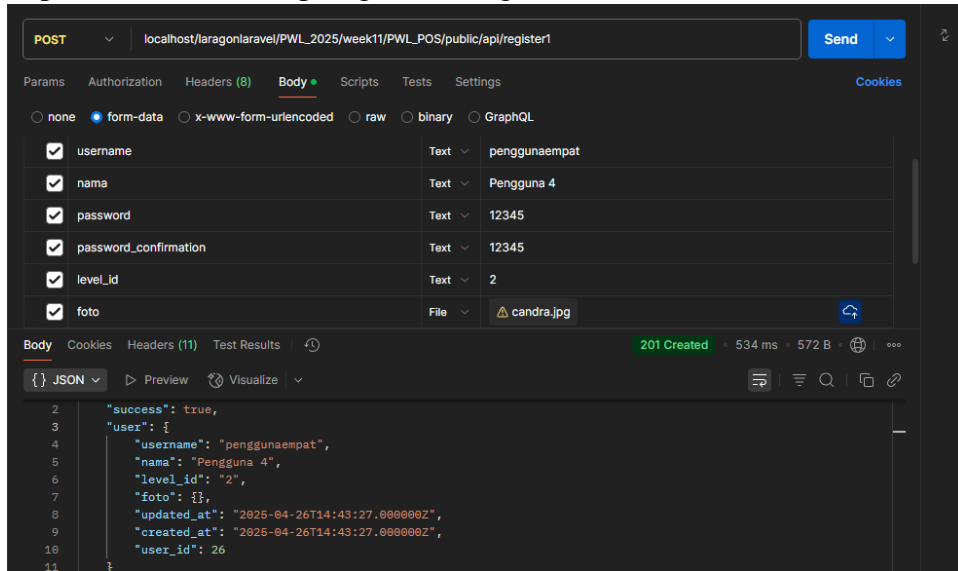
```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Models\UserModel;
6  use App\Http\Controllers\Controller;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Validator;
9
10 class RegisterController extends Controller
11 {
12     public function __invoke(Request $request)
13     {
14         //set validator
15         $validator = Validator::make($request->all(), [
16             'username' => 'required',
17             'nama' => 'required',
18             'password' => 'required|min:5|confirmed',
19             'level_id' => 'required',
20             'foto' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
21         ]);
22         //if validations fails
23         if ($validator->fails()) {
24             return response()->json($validator->errors(), 422);
25         }
26         //create user
27         $user = UserModel::create([
28             'username' => $request->username,
29             'nama' => $request->nama,
30             'password' => bcrypt($request->password),
31             'level_id' => $request->level_id,
32             'foto' => $request->foto
33         ]);
34         //return response JSON user is created
35         if ($user) {
36             return response()->json([
37                 'success' => true,
38                 'user' => $user,
39             ], 201);
40         }
41         //return response JSON insert failed
42         return response()->json([
43             'success' => false,
44             ], 409);
45     }
46 }
```

7. Anda dapat menambahkan detail untuk spesifikasi image pada validator 8. Ubah atau tambahkan register1 pada routes/api.php
8. Simpan dan akses pada aplikasi Postman, atur pada Body isi manual Key dan Valuenya pada Key image tambahkan value File dan upload gambar



Jurusan Teknologi Informasi Politeknik Negeri Malang  
JOBSHEET 11 RESTFUL API 2  
Mata Kuliah Pemrograman Web Lanjut (PWL)

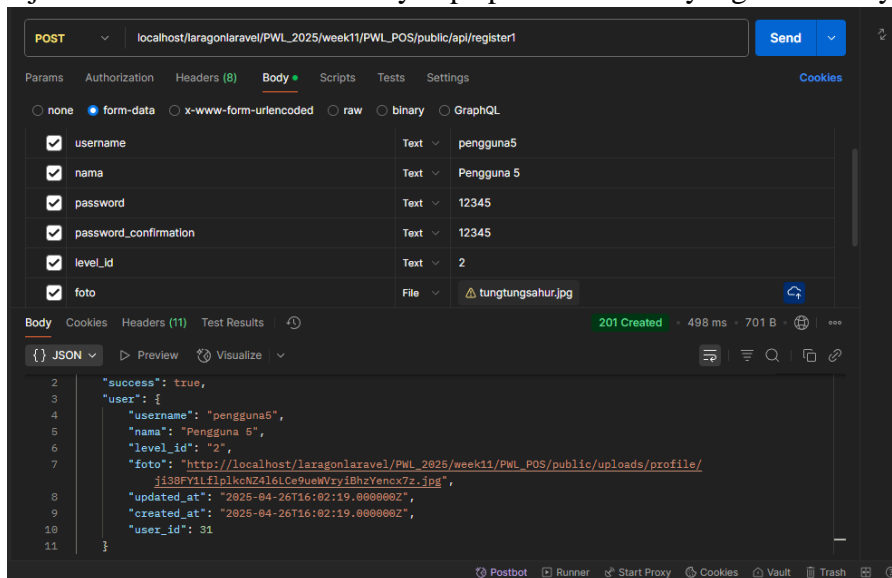
http://127.0.0.1:8000/api/register1 dengan method POST dan klik send



9. 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,
    'foto' => $request->foto->hashName(),
]);
```

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



perbedaannya nama foto disimpan menggunakan hash



## Jurusan Teknologi Informasi Politeknik Negeri Malang

### JOB SHEET 11 RESTFUL API 2

#### Mata Kuliah Pemrograman Web Lanjut (PWL)

### 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.

The screenshot displays two Postman requests in a dark theme. The top request is a POST to `localhost/laragonlaravel/PWL_2025/week11/PWL_POS/public/api/barangs` with a status of **201 Created**. The bottom request is a GET to `localhost/laragonlaravel/PWL_2025/week11/PWL_POS/public/api/barangs/16` with a status of **200 OK**.

**POST Request Details:**

- URL: `localhost/laragonlaravel/PWL_2025/week11/PWL_POS/public/api/barangs`
- Method: POST
- Body (form-data):
  - `kategori_id`: 1
  - `barang_kode`: MKN-0004
  - `barang_nama`: Ikan Bakar
  - `harga_beli`: 10000
  - `harga_jual`: 15000
  - `image`: tungtungsahur.jpg
- Status: 201 Created, 365 ms, 759 B
- Response (JSON):

```
{  "barang": {    "kategori_id": "1",    "barang_kode": "MKN-0004",    "barang_nama": "Ikan Bakar",    "harga_beli": "10000",    "harga_jual": "15000",    "image": "http://localhost/laragonlaravel/PWL_2025/week11/PWL_POS/public/storage/barang/111LK12XZMyhcEMJoxbENxHbuU1MYQWwzx1KGdzK.jpg",    "updated_at": "2025-04-26T17:31:48.000000Z",    "created_at": "2025-04-26T17:31:48.000000Z",    "barang_id": 16  }
```

**GET Request Details:**

- URL: `localhost/laragonlaravel/PWL_2025/week11/PWL_POS/public/api/barangs/16`
- Method: GET
- Status: 200 OK, 244 ms, 722 B
- Response (JSON):

```
{  "barang_id": 16,  "kategori_id": 1,  "barang_kode": "MKN-0004",  "barang_nama": "Ikan Bakar",  "harga_beli": 10000,  "harga_jual": 15000,  "created_at": "2025-04-26T17:31:48.000000Z",  "updated_at": "2025-04-26T17:31:48.000000Z",  "image": "http://localhost/laragonlaravel/PWL_2025/week11/PWL_POS/public/storage/barang/111LK12XZMyhcEMJoxbENxHbuU1MYQWwzx1KGdzK.jpg"}
```