

Nama : Ereen Lourenza Natalia Mamahi  
Kelas / no : SIB3C / 09  
NIM : 2141762044

## Jobsheet 11

### Restful API 2

#### Praktikum 1 – Implementasi Eloquent Accessor

1

Memodifikasi table m\_user dengan menambahkan column image menggunakan migration

C:\laragon\www\Pemrograman\_Web\_Lanjut\Pertemuan 3\PWL\_POS>php artisan make:migration add\_image\_to\_m\_user\_table

INFO

 Migration [C:\laragon\www\Pemrograman\_Web\_Lanjut\Pertemuan 3\PWL\_POS\database\migrations\2024\_05\_20\_080602\_add\_image\_to\_m\_us  
er\_table.php] created successfully.

🔗

 2024\_05\_20\_080602\_add\_image\_to\_m\_user\_table.php    U

2

Memodifikasi isi migration

```
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');
    });
}
```

3

Melakukan update migration

C:\laragon\www\Pemrograman\_Web\_Lanjut\Pertemuan 3\PWL\_POS>php artisan migrate

INFO

 Running migrations.

2024\_05\_20\_080602\_add\_image\_to\_m\_user\_table ..... 115ms **DONE**

Hasil update migration

status	profile_img	created_at	updated_at	image
1	NULL	NULL	NULL	

4

Melakukan modifikasi di UserModel

```
class UserModel extends Authenticatable implements JWTSubject
{
    /**
     * protected $fillable = ['level_id','username','nama','password', 'status', 'profile_img', 'image'];
     */

    public function level(): BelongsTo{
        return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
    }

    // public function stok(): HasMany{
    //     return $this->hasMany(StokModel::class, 'stok_id', 'stok_id');
    // }

    protected function image(): Attribute{
        return Attribute::make(
            get: fn ($image) => url('/storage/posts/' . $image)
        );
    }

    protected $hidden = [
        'password'
    ];
}
```

## 5 Modifikasi Api/RegisterController

```
//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',
]);

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

## 6 Menambahkan route register1

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

## 7 Akses Postman dengan method POST (127.0.0.1:8000/api/register1)

The screenshot shows the Postman interface with a POST request to `127.0.0.1:8000/api/register1`. The body is set to `form-data` with the following fields:

Key	Value	Description
username	penggunaempat	
nama	pengguna4	
password	12345	
password_confirmation	12345	
level_id	2	
image	File	

The response is a JSON object:

```
{
  "success": true,
  "user": {
    "username": "penggunaempat",
    "nama": "pengguna4",
    "level_id": "2",
    "image": "http://127.0.0.1:8000/storage/posts/C:\\xampp\\tmp\\phpB68B.tmp",
    "updated_at": "2024-05-20T08:23:21.000000Z",
    "created_at": "2024-05-20T08:23:21.000000Z",
    "user_id": 9
  }
}
```

8	<p>Mengganti create user image di RegisterController</p> <pre>//create user \$user = UserModel::create([     'username' =&gt; \$request-&gt;username,     'nama' =&gt; \$request-&gt;nama,     'password' =&gt; bcrypt(\$request-&gt;password),     'level_id' =&gt; \$request-&gt;level_id,     'image' =&gt; \$request-&gt;image-&gt;hashName(), ]);</pre>
9	<p>Hasil</p> <pre>{   "success": true,   "user": {     "username": "penggunalima",     "nama": "pengguna5",     "level_id": "2",     "image": "http://127.0.0.1:8000/storage/posts/NS2RLIFHfqahXrFp8HpL0DKI9Wc5KZEBE72SGrvb.jpg",     "updated_at": "2024-05-20T08:31:17.000000Z",     "created_at": "2024-05-20T08:31:17.000000Z",     "user_id": 10   } }</pre> <p>Perbedaannya dari penamaan file menjadi hash dimana nama menjadi rahasia</p>

## TUGAS – Upload File / Gambar

m_barang	
1	<p>Menambahkan column image di table m_barang menggunakan migration</p> <pre>C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PWL_POS&gt;php artisan make:migration add_image_to_m_barang_table</pre> <pre>INFO Migration [C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PWL_POS\database\migrations\2024_05_20_083906_add_image_to_m_barang_table.php] created successfully.</pre> <pre>2024_05_20_084137_add_image_to_m_barang_table.php U</pre>
2	<p>Memodifikasi isi migration</p> <pre>public function up(): void {     Schema::table('m_barang', function (Blueprint \$table) {         \$table-&gt;string('image');     }); }  /**  * Reverse the migrations.  */ public function down(): void {     Schema::table('m_barang', function (Blueprint \$table) {         \$table-&gt;dropColumn('image');     }); }</pre>
3	<p>Melakukan update migration</p> <pre>C:\laragon\www\Pemrograman_Web_Lanjut\Pertemuan 3\PWL_POS&gt;php artisan migrate</pre> <pre>INFO Running migrations.  2024_05_20_084137_add_image_to_m_barang_table 257ms DONE</pre>



## 6 Akses Postman dengan method POST (127.0.0.1:8000/api/barangs)

The screenshot shows the Postman interface for a POST request. The URL is `127.0.0.1:8000/api/barangs`. The request body is set to `form-data` with the following fields:

Key	Value	Description
<input checked="" type="checkbox"/> barang_kode	Text HDP	
<input checked="" type="checkbox"/> barang_nama	Text Handphone	
<input checked="" type="checkbox"/> harga_beli	Text 1800000	
<input checked="" type="checkbox"/> harga_jual	Text 2000000	
<input checked="" type="checkbox"/> kategori_id	Text 3	
<input checked="" type="checkbox"/> image	File Artwor...	
Key	Text Value	Description

The response body is shown in the console, displaying a JSON object:

```
{
  "success": true,
  "barang": {
    "barang_kode": "HDP",
    "barang_nama": "Handphone",
    "harga_beli": "1800000",
    "harga_jual": "2000000",
    "kategori_id": "3",
    "image": "http://127.0.0.1:8000/storage/posts/G0hcS9s1LwZ6ArMgkLVZwDVbDUe4S3UqratFPIBk.jpg",
    "updated_at": "2024-05-20T09:48:33.000000Z",
    "created_at": "2024-05-20T09:48:33.000000Z",
    "barang_id": 12
  }
}
```

## 7 Memanggil data menggunakan GET

The screenshot shows the Postman interface for a GET request. The URL is `127.0.0.1:8000/api/barangs/12`. The response body is shown in the console, displaying a JSON object:

```
{
  "barang_id": 12,
  "kategori_id": 3,
  "barang_kode": "HDP",
  "barang_nama": "Handphone",
  "harga_beli": 1800000,
  "harga_jual": 2000000,
  "created_at": "2024-05-20T09:48:33.000000Z",
  "updated_at": "2024-05-20T09:48:33.000000Z",
  "image": "http://127.0.0.1:8000/storage/posts/G0hcS9s1LwZ6ArMgkLVZwDVbDUe4S3UqratFPIBk.jpg"
}
```