



Nama : Nimas Septiandini  
NIM : 2341760087  
Kelas : SIB-2B

Mata Kuliah : Pemrograman Web Lanjut (PWL)  
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis  
Semester : 4 (empat) / 6 (enam)  
Pertemuan ke- : 11 (sebelas)

## **JOBSHEET 11**

### **RESTFUL API 2**

Sebelumnya kita sudah membahas mengenai *RESTFUL API dan JSON Web Token (JWT)* pada Laravel. Dimana kita telah membuat RESTful API Register, RESTful API Login, RESTful API Logout, dan implementasi CRUD dalam RESTful API pada halaman web. Pada pertemuan kali ini, kita akan mempelajari penerapan RESTFUL API lanjutan di dalam project Laravel.

Sebelum kita masuk materi, kita buat dulu project baru yang akan kita gunakan untuk membangun aplikasi sederhana dengan topik *Point of Sales (PoS)*, sesuai dengan **Studi Kasus PWL.pdf**. Jadi kita bikin project Laravel 10 dengan nama **PWL\_POS**.

*Project PWL\_POS* akan kita gunakan sampai pertemuan 12 nanti, sebagai project yang akan kita pelajari

### **A. ELOQUENT ACCESSOR**

Laravel memiliki fitur yang bernama mutator, accessor dan casting, fitur-fitur ini digunakan untuk melakukan manipulasi data di dalam attribute database dengan sangat mudah. Contohnya pada saat insert data dengan enkripsi ke dalam database dan melakukan deskripsi saat menampilkan dari database secara otomatis.

Accessor dapat mengubah nilai saat attribute atau field eloquent diakses. Untuk mendefinisikan Accessor dapat membuat method di dalam model untuk menentukan attribute yang akan diakses. Nama method yang dibuat harus sama dengan nama attribute yang akan di format: contoh :



Attribute/field pada tabel first\_name maka methodnya firstName()

protected function firstName(): Attribute

```
{ //...  
}
```

Jika membuat attribute/field image yang ada di table m\_user kita akan memberikan nilai full path dari direktori dimana file gambar tersebut disimpan. contohnya pada UserModel ditambahkan

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

Dengan begitu dapat melakukan import Eloquent Attribute dengan

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

Lalu method baru dengan nama image() melakukan return path nama file image itu berada

```
get: fn ($image) => url('/storage/posts/' . $image),
```

Hasil akhir memanggil attribute image

```
domain.com/storage/posts/nama_file_image.png
```

## 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**

```
PS C:\laragon\www\PWL_POS> php artisan make:migration add_image_to_m_user_table  
[INFO] Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_04_30_040822_add_image_to_m_user_table.php] created successfully.
```

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) {
            $table->string('image');
        });
    }

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

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

**INFO** Running migrations.

2025\_04\_28\_065129\_add\_image\_to\_m\_user\_table ..... 122ms **DONE**

5. Lalu lakukan modifikasi models pada App/Models/UserModel.php



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

---

```
<?php
namespace App\Models;
    use Illuminate\Database\Eloquent\Model; use
Tymon\JWTAuth\Contracts\JWTSubject; use
Illuminate\Database\Eloquent\Casts\Attribute; use
Illuminate\Foundation\Auth\User as Authenticatable;
```



```
class UserModel extends Authenticatable implements JWTSubject {
    public function getJWTIdentifier(){           return $this->
    >getKey();
    }      public function
    getJWTCustomClaims(){           return [];
    }      protected $table = 'm_user';
    protected $primaryKey = 'user_id';
    protected $fillable = [
        'username',
        'nama',
        'password',
        'level_id',
        'image'//tambahan
    ];      public function
    level()
    {
        return $this->belongsTo(LevelModel::class,
        'level_id', 'level_id');    }
    protected function image(): Attribute
    {
        return Attribute::make(
            get: fn ($image) => url('/storage/posts/' . $image),
        );
    }
}
```

6. Lakukan modifikasi pada Controllers/Api/RegisterController

```
<?php
namespace App\Http\Controllers\Api;
    use App\Models\UserModel; use
    App\Http\Controllers\Controller;
    use Illuminate\Http\Request;
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI  
**POLITEKNIK NEGERI MALANG**  
**JURUSAN TEKNOLOGI INFORMASI**  
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141  
Telp. (0341) 404424 – 404425, Fax (0341) 404420  
<http://www.polinema.ac.id>

---



```
use Illuminate\Support\Facades\Validator;
class RegisterController extends
Controller
{
    public function __invoke(Request
$request)
    {
        //set validation
        $validator = Validator::make($request->all(), [
            'username' => 'required',
            'nama' => 'required',
            'password' => 'required|min:5|confirmed',
            'level_id' => 'required',
            'image' => 'required'

        ]);

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

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



7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

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

8. Ubah atau tambahkan register1 pada routes/api.php

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

8. 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

RESTful API basics: CRUD, test & variable / <http://127.0.0.1:8000/api/register1>

POST <http://127.0.0.1:8000/api/register1> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

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

Key	Value	Content-Type	Description
<input checked="" type="checkbox"/> username	Text penggunaempat	Auto	
<input checked="" type="checkbox"/> nama	Text pengguna4	Auto	
<input checked="" type="checkbox"/> password	Text 12345	Auto	
<input checked="" type="checkbox"/> password_confirmation	Text 12345	Auto	
<input checked="" type="checkbox"/> level_id	Text 2	Auto	
<input checked="" type="checkbox"/> image	File Screenshot 2023-04-28 095851.png	Auto	

body Cookies Headers (10) Test Results Status: 201 Created Time: 591 ms Size: 691 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "user": {
4     "username": "penggunaempat",
5     "nama": "pengguna4",
6     "password": "$2y$12$9sd1lI/uJfy33ABuJ9s50tkYe3rlG9ci.SyNvbys5CYJgvh0vYlm",
7     "level_id": "2",
8     "image": "http://127.0.0.1:8000/storage/posts/c:/Users/Asus Zenbook 14 OLED/AppData/Local/Temp/php57C5.tmp",
9     "updated_at": "2024-04-30T07:16:25.000000Z",
10    "created_at": "2024-04-30T07:16:25.000000Z",
11    "user_id": 22
12  }
13 }
```





HTTP <http://127.0.0.1:8000/api/register1> Save Share

POST <http://127.0.0.1:8000/api/register1> Send

Params Auth Headers (9) Body Scripts Settings Cookies

form-data

Key	Value	Description
<input checked="" type="checkbox"/> nama	Text pengguna44	
<input checked="" type="checkbox"/> password	Text 12345	
<input checked="" type="checkbox"/> password_confirm...	Text 12345	
<input checked="" type="checkbox"/> level_id	Text 2	
<input checked="" type="checkbox"/> image	File What The Girl Fr...	

Body 201 Created • 1.10 s • 680 B

{ JSON Preview Visualize

```
1 {
2   "success": true,
3   "user": {
4     "username": "penggunaempatempat",
5     "nama": "pengguna44",
6     "password": "$2y$12$Hg0s7D4tTMY0q3fS1l4Bk.FFqycIwFUrKoxbDBU16RFn85hg.wiDa",
7     "level_id": "2",
8     "image": "http://127.0.0.1:8000/storage/posts/C:\\Users\\NXST\\AppData\\Local\\Temp\\php41B8.tmp",
9     "updated_at": "2025-04-28T07:26:50.000000Z",
10    "created_at": "2025-04-28T07:26:50.000000Z",
11    "user_id": 28
  }
```

9. Pada Controllers/Api/RegisterController bagian create user ganti dengan

```
'image' => $image->hashName(),
```

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



HTTP <http://127.0.0.1:8000/api/register1> Save Share

POST <http://127.0.0.1:8000/api/register1> Send

Params Auth Headers (9) **Body** Scripts Settings Cookies

form-data

<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_confirm...	Text	12345
<input checked="" type="checkbox"/>	level_id	Text	2
<input checked="" type="checkbox"/>	image	File	images (2).jpeg

Body 201 Created 6.22 s 669 B

{ JSON Preview Visualize

```
1 {
2   "success": true,
3   "user": {
4     "username": "penggunaempat",
5     "nama": "pengguna4",
6     "password": "$2y$12$hZ.6Vzd1SiTSrpk2g18n20E2SEcKBq/e2MTuE4.
7       gFr1F6D31WA0SC",
8     "level_id": "2",
9     "image": "http://127.0.0.1:8000/storage/posts/
10       zItmapSIQM9nAL2EHUM7dLDaTfzzZ7iqCidIQn4Y.jpg",
11     "updated_at": "2025-04-28T07:23:05.000000Z",
12     "created_at": "2025-04-28T07:23:05.000000Z",
13     "user_id": 27
14   }
15 }
```



## 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.

### 1. Get Barang

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** <http://127.0.0.1:8000/api/barangs>
- Send Button:** A blue button labeled "Send" with a dropdown arrow.
- Params:** A tab labeled "form-data" with a dropdown arrow.
- Table:** A table with 4 columns: Key, Value, Description, and Bulk Edit. The first row has "Key" and "Text" in the first two columns, and "Value" and "Description" in the last two columns.
- Body:** A tab labeled "Body" with a dropdown arrow.
- Status:** 200 OK, 800 ms, 2.9 KB.
- JSON Response:** A JSON array containing one object with the following fields:

```
1 [
2   {
3     "barang_id": 1,
4     "kategori_id": 2,
5     "barang_kode": "B001",
6     "barang_nama": "Laptop",
7     "harga_beli": 5000000,
8     "harga_jual": 6000000,
9     "created_at": null,
10    "updated_at": null,
11    "image": null
12  },
13 ]
```



## 2. Melakukan input data barang

The screenshot shows a REST client interface with the following details:

- URL:** `http://127.0.0.1:8000/api/barangs`
- Method:** `POST`
- Body Type:** `form-data`
- Form Data Fields:**

Key	Value
<input checked="" type="checkbox"/> kategori_id	1
<input checked="" type="checkbox"/> barang_kode	B016
<input checked="" type="checkbox"/> barang_nama	Lampu Philips
<input checked="" type="checkbox"/> harga_beli	45000
<input checked="" type="checkbox"/> harga_jual	50000
<input checked="" type="checkbox"/> image	LampuPhilips.jpg
- Response:** `201 Created` (1.66 s, 593 B)
- Response Body (JSON):**

```
{  "barang_kode": "B016",  "barang_nama": "Lampu Philips",  "kategori_id": "1",  "harga_beli": "45000",  "harga_jual": "50000",  "image": "http://127.0.0.1:8000/storage/barang/1745850724.jpg",  "updated_at": "2025-04-28T14:32:05.000000Z",  "created_at": "2025-04-28T14:32:05.000000Z",  "barang_id": 18}
```



### 3. Melihat data barang

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/api/barangs`. The request is successful, returning a 200 OK status. The response body is a JSON array containing one object representing a product.

Key	Value	Description
<input checked="" type="checkbox"/> kategori_id	Text	1
<input checked="" type="checkbox"/> barang_kode	Text	B016
<input checked="" type="checkbox"/> barang_nama	Text	Lampu Philips
<input checked="" type="checkbox"/> harga_beli	Text	45000
<input checked="" type="checkbox"/> harga_jual	Text	50000
<input checked="" type="checkbox"/> image	File	LampuPhilips.jpg

```
{
  "barang_id": 18,
  "kategori_id": 1,
  "barang_kode": "B016",
  "barang_nama": "Lampu Philips",
  "harga_beli": 45000,
  "harga_jual": 50000,
  "created_at": "2025-04-28T14:32:05.000000Z",
  "updated_at": "2025-04-28T14:32:05.000000Z",
  "image": "http://127.0.0.1:8000/storage/barang/1745850724.jpg"
}
```

\*\*\* Sekian, dan selamat belajar \*\*\*