



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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\laragon\www\PMW_2025\Week11\PMW_POS> php artisan make:migration add_image_to_m_user_table  
INFO Migration [C:\laragon\www\PMW_2025\Week11\PMW_POS\database\migrations\2025_04_28_110222_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

```
PS C:\laragon\www\PNM_2025\Week11\PNM_POS> php artisan migrate
INFO Running migrations.
2025_04_14_135121_add_profile_picture_to_m_user_table ..... 6ms DONE
2025_04_22_043137_add_stok_to_barang_table ..... 50ms DONE
2025_04_28_110222_add_image_to_m_user_table ..... 49ms DONE
```

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

```
<?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;
```



```
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

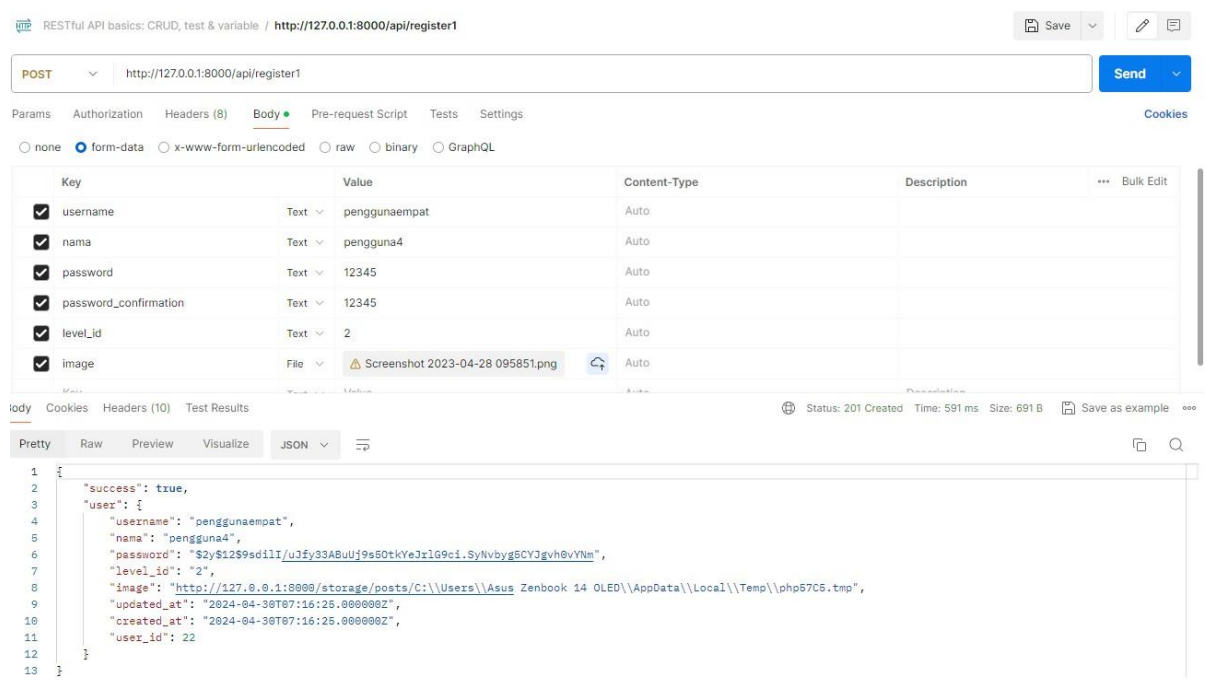
```
'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
```

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

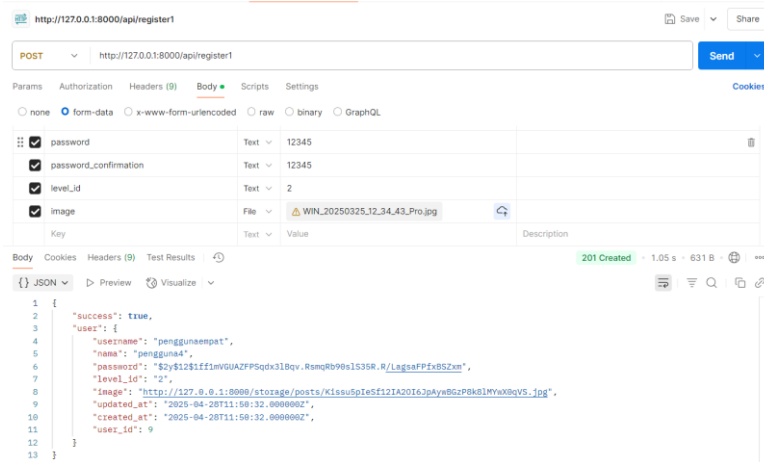


9. Pada Controllers/Api/RegisterController bagian create user ganti dengan

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

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

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

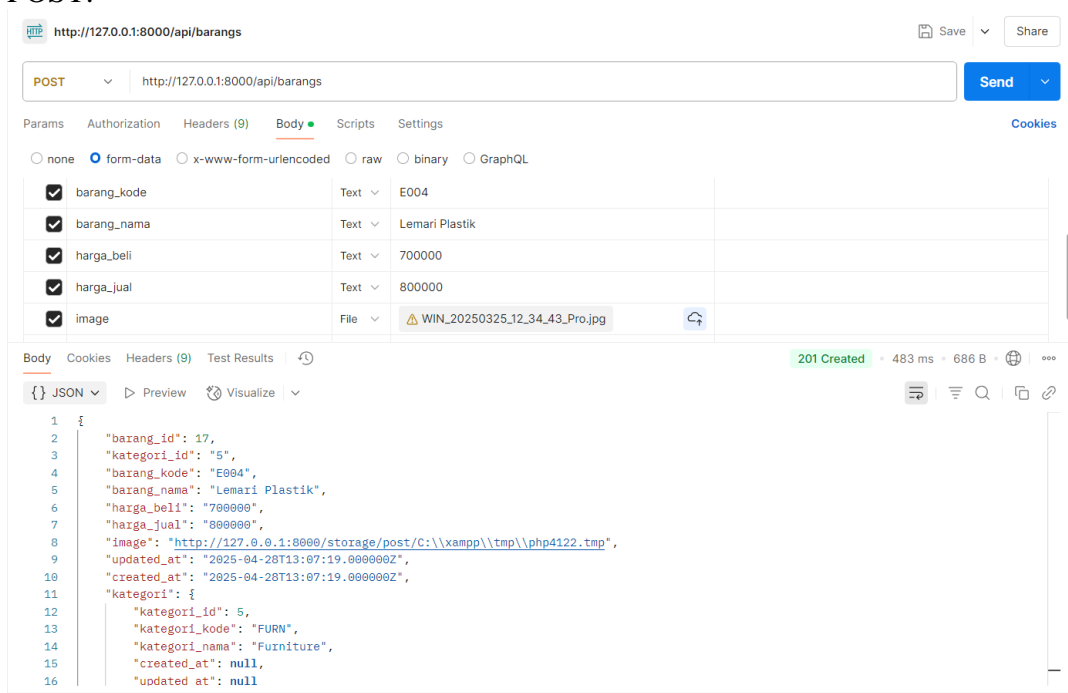


TUGAS

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. M_barang

POST:





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>

GET:

http://127.0.0.1:8000/api/barangs/17

GET http://127.0.0.1:8000/api/barangs/17

Params Authorization Headers (9) Body Scripts Settings Cookies

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

<input type="checkbox"/> barang_nama	Text	Lemari Plastik
<input type="checkbox"/> harga_beli	Text	700000
<input type="checkbox"/> harga_jual	Text	800000
<input type="checkbox"/> image	File	WIN_20250325_12_34_43_Pro.jpg
Key	Text	Value
		Description

Body Cookies Headers (9) Test Results

200 OK • 440 ms • 684 B

JSON Preview Visualize

```
1 {
2   "barang_id": 17,
3   "kategori_id": 5,
4   "barang_kode": "E004",
5   "barang_nama": "Lemari Plastik",
6   "harga_beli": 700000,
7   "harga_jual": 800000,
8   "stok": 0,
9   "created_at": "2025-04-28T13:07:19.000000Z",
10  "updated_at": "2025-04-28T13:07:19.000000Z",
11  "image": "http://127.0.0.1:8000/storage/post/C:\\xampp\\tmp\\php4122.tmp",
12  "kategori": {
13    "kategori_id": 5,
14    "kategori_kode": "FURN",
15    "kategori_nama": "Furniture",
16    "created_at": null,
```

DELETE:

http://127.0.0.1:8000/api/barangs/17

DELETE http://127.0.0.1:8000/api/barangs/17

Params Authorization Headers (9) Body Scripts Settings Cookies

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

<input type="checkbox"/> barang_nama	Text	Lemari Plastik
<input type="checkbox"/> harga_beli	Text	700000
<input type="checkbox"/> harga_jual	Text	800000
<input type="checkbox"/> image	File	WIN_20250325_12_34_43_Pro.jpg
Key	Text	Value
		Description

Body Cookies Headers (9) Test Results

200 OK • 503 ms • 313 B

JSON Preview Visualize

```
1 {
2   "success": true,
3   "message": "Data terhapus"
4 }
```




2. Transaksi

POST:

http://127.0.0.1:8000/api/transaksis

POST http://127.0.0.1:8000/api/transaksis

Params Authorization Headers (9) Body Scripts Settings Cookies

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

Key	Value	Description
penjualan_id	11	
user_id	1	
pembeli	Pelanggan 11	
penjualan_kode	ORD011	

Body Cookies Headers (9) Test Results

201 Created • 1.48 s • 472 B

```
{
  "user_id": "1",
  "pembeli": "Pelanggan 11",
  "penjualan_kode": "ORD011",
  "penjualan_tanggal": "2025-04-29 11:57:21",
  "image": "http://127.0.0.1:8000/storage/transaksi/1745846949.jpg",
  "penjualan_id": 11
}
```

GET:

http://127.0.0.1:8000/api/transaksis/11

GET http://127.0.0.1:8000/api/transaksis/11

Params Authorization Headers (7) Body Scripts Settings Cookies

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

pembeli	Pelanggan 11	
penjualan_kode	ORD011	
penjualan_tanggal	2025-04-29 11:57:21	
image	WIN_20250325_12_34_43_Pro.jpg	
Key	Value	Description

Body Cookies Headers (9) Test Results

200 OK • 334 ms • 526 B

```
{
  "penjualan_id": 11,
  "user_id": 1,
  "pembeli": "Pelanggan 11",
  "penjualan_kode": "ORD011",
  "penjualan_tanggal": "2025-04-29 11:57:21",
  "image": "http://127.0.0.1:8000/storage/transaksi/1745846949.jpg",
  "created_at": null,
  "updated_at": null,
  "user": null,
  "details": []
}
```



DELETE:

Getting started | DEL <http://127.0.0.1:8000/api> | DEL <http://127.0.0.1:8000/api> | No environment

<http://127.0.0.1:8000/api/transaksis/11> Save Share

DELETE <http://127.0.0.1:8000/api/transaksis/11> Send

Params Authorization Headers (7) **Body** Scripts Settings Cookies

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

<input type="checkbox"/> pembeli	Text	Pelanggan 11	
<input type="checkbox"/> penjualan_kode	Text	ORD011	
<input type="checkbox"/> penjualan_tanggal	Text	2025-04-29 11:57:21	
<input type="checkbox"/> image	File	WIN_20250325_12_34_43_Pro.jpg	
Key	Text	Value	Description

Body Cookies Headers (9) Test Results

200 OK 416 ms 317 B

JSON Preview Visualize

```
1 {
2   "success": true,
3   "message": "Transaksi dihapus"
4 }
```

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