

## **LAPORAN PRAKTIKUM**

### **JOBSHEET XI: RESTFUL API (*APPLICATION PROGRAMMING INTERFACE*) BAGIAN 2**

#### **MATA KULIAH PEMROGRAMAN WEB LANJUT**

Dosen Pengampu : Dimas Wahyu Wibowo, S.T., M.T.



**Disusun oleh :**

Nama : Vita Eka Saraswati

NIM : 2341760082

No. Absen : 29

**JURUSAN TEKNOLOGI INFORMASI**

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**POLITEKNIK NEGERI MALANG**

**2025**



Nama : Vita Eka Saraswati  
NIM/Kelas : 2341760082 / SIB 2A  
No. Absen : 29  
Mata Kuliah : Pemrograman Web Lanjut (PWL)

## 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 ditable 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 F:\laragon\www\Semester 4\PWL25\Wingui11\PWL_POS> php artisan make:migration add_image_to_m_user_table  
INFO: Migration [F:\laragon\www\Semester 4\PWL25\Wingui11\PWL_POS\database\migrations\2025_04_30_085940_add_image_to_m_user_table.php] created successfully.
```

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



```
2025_04_30_085940_add_image_to_m_user_table.php U X
Minggu11 > PWL_POS > database > migrations > 2025_04_30_085940_add_image_to_m_user_table.php > _
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::table('m_user', function (Blueprint $table): void {
15             $table->string('image');
16         });
17     }
18
19     /**
20      * Reverse the migrations.
21      */
22     public function down(): void
23     {
24         Schema::table('m_user', function (Blueprint $table): void {
25             $table->dropColumn('image');
26         });
27     }
28 };
```

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

```
PS F:\laragon\www\Semester 4\PWL25\Minggu11\PWL_POS> php artisan migrate
INFO Running migrations.
2025_04_30_085940_add_image_to_m_user_table ..... 496ms DONE
```

Showing rows 0 - 12 (13 total, Query took 0.0007 seconds.)

SELECT \* FROM "m\_user"

Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all Number of rows: 25 Filter rows: Search this table Sort by: None

Extra options

	user_id	level_id	profile_photo	username	nama	password	created_at	updated_at	image
<input type="checkbox"/>	1	1	profiles/profile_1/1748725361.jpg	admin	Administrator	\$2y\$12\$HhYkKkL4dU0.aV7Nc1edwBv6R0HPuPgywKOL...	2025-04-27 10:42:41	2025-04-27 10:42:41	

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 App\Models\LevelModel;
9  use Tymon\JWTAuth\Contracts\JWTSubject;
10 use Illuminate\Database\Eloquent\Casts\Attribute;
11 use Illuminate\Foundation\Auth\User as Authenticatable; //implementasi class Authenticatable
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28  protected $fillable = ['username','password','nama','level_id','created_at','updated_at','profile_photo','image']; // Mendefinisikan field yang dapat diisi saat melakukan insert-references
```

```
0 references | 0 overrides
38  protected function image(): Attribute
39  {
40      {
41          return Attribute::make(
42              get: fn ($image): string|UrlGenerator => url(path: './storage/posts/' . $image),
43          );
44      }
45
46      //Mendapatkan nama role
```



6. Lakukan modifikasi pada Controllers/Api/RegisterController

```
1 use Illuminate\Http\Request;
2 use Illuminate\Support\Facades\Validator;
3
4 // references | 0 implementations
5 class RegisterController extends Controller
6 {
7     // references | 0 overrides
8     public function __invoke(Request $request): JsonResponse|mixed
9     {
10         // Set validation
11         $validator = Validator::make($request->all(), [
12             'username' => 'required',
13             'nama' => 'required',
14             'password' => 'required|min:5|confirmed',
15             'level_id' => 'required',
16             'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
17         ]);
18
19         // If validation fails
20         if ($validator->fails()) {
21             return response()->json($validator->errors(), 422);
22         }
23
24         // Create user
25         $user = UserModel::create([
26             'username' => $request->username,
27             'nama' => $request->nama,
28             'password' => bcrypt(value: $request->password),
29             'level_id' => $request->level_id,
30             'image' => $request->image
31         ]);
32
33         // Return response JSON if user is created
34         if ($user) {
35             return response()->json([
36                 'success' => true,
37                 'user' => $user,
38             ], 201);
39         }
40
41         // Return JSON if process insert failed
42         return response()->json([
43             'success' => false,
44         ], 409);
45     }
46 }
```

7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

```
12 public function __invoke(Request $request): JsonResponse|mixed
13 {
14     // Set validation
15     $validator = Validator::make($request->all(), [
16         'username' => 'required',
17         'nama' => 'required',
18         'password' => 'required|min:5|confirmed',
19         'level_id' => 'required',
20         'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
21     ]);
22 }
```

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

```
25 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
26 Route::post('/register1', App\Http\Controllers\Api\RegisterController::class)->name('register1');
27 Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');
28 Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
29
```

9. 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



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>

POST localhost/Semester%204/PWL25/Minggu11/PWL\_POS/public/api/register1? Send

Params Authorization Headers (8) Body Scripts Settings Cookies

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

<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_confirmation	Text	12345
<input checked="" type="checkbox"/> level_id	Text	2
<input checked="" type="checkbox"/> image	File	-2147483648 -213041.png

Body Cookies Headers (11) Test Results Visualize

201 Created 6.96 s 706 B Save Response

```
{
  "success": true,
  "user": {
    "username": "penggunaempat",
    "nama": "pengguna4",
    "level_id": "2",
    "image": "http://localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/storage/posts/C:\\Users\\Acer\\AppData\\Local\\Temp\\phpDF98.tmp",
    "updated_at": "2025-04-30T13:44:31.000000Z",
    "created_at": "2025-04-30T13:44:31.000000Z",
    "user_id": 36
  }
}
```

Activate Windows  
Go to Settings to activate Windows

Data user tersimpan di tabel m\_user berupa path tanpa di hash

36	2 NULL	penggunaempat pengguna4	\$2y\$12\$Khavd2Joh/sw9D0SZFJU.kKX.LJzhAbsPPf0eUeAZ...	2025-04-30 20:44:31	2025-04-30 20:44:31	C:\\Users\\Acer\\AppData\\Local\\Temp\\phpDF98.tmp
----	--------	-------------------------	--	---------------------	---------------------	--

## 10. Pada Controllers/Api/RegisterController bagian create user ganti dengan

```
28 // Create user
29 $user = UserModel::create([
30     'username' => $request->username,
31     'nama' => $request->nama,
32     'password' => bcrypt($request->password),
33     'level_id' => $request->level_id,
34     'image' => $request->image->hashName(),
35 ]);
36
```

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

PWL-POS Register api / register Save Share

POST localhost/Semester%204/PWL25/Minggu11/PWL\_POS/public/api/register1? Send

Params Authorization Headers (8) Body Scripts Settings Cookies

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

<input checked="" type="checkbox"/> username	Text	staff03
<input checked="" type="checkbox"/> nama	Text	STF_Kevin
<input checked="" type="checkbox"/> password	Text	12345
<input checked="" type="checkbox"/> password_confirmation	Text	12345
<input checked="" type="checkbox"/> level_id	Text	3
<input checked="" type="checkbox"/> image	File	IMG_2391.PNG

Body Cookies Headers (11) Test Results Visualize

201 Created 2.59 s 694 B Save Response

```
{
  "success": true,
  "user": {
    "username": "staff03",
    "nama": "STF_Kevin",
    "level_id": "3",
    "image": "http://localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/storage/posts/wQfY8lNpA3ot25Q0qeE5hrocoi5PMe3NN05aHxg.jpg",
    "updated_at": "2025-04-30T14:03:29.000000Z",
    "created_at": "2025-04-30T14:03:29.000000Z",
    "user_id": 37
  }
}
```

Activate Windows  
Go to Settings to activate Windows

37	3 NULL	staff03	STF_Kevin	\$2y\$12\$1G939wrtzAaUdeYCaKkeEONZXSJm8aV4Vz1Hz...	2025-04-30 21:00:29	2025-04-30 21:00:29	wQfY8lNpA3ot25Q0qeE5hrocoi5PMe3NN05aHxg.jpg
----	--------	---------	-----------	--	---------------------	---------------------	---

### Jawab:

Field image user akan disimpan menggunakan hash. `hashName()` adalah method untuk menghasilkan nama file yang unik dengan cara menghash nama file asli. Hash digunakan untuk alasan keamanan dan mencegah konflik apabila ada user yang



mengupload image dengan nama yang sama.

## TUGAS

Implementasikan API untuk upload file/gambar pada tabel lainnya yaitu m\_barang dan gunakan pada transaksi. Uji coba dengan method GET untuk memanggil data yang sudah diinputkan.

### a. Tabel m\_barang

- Buat migrasi untuk menambahkan column image

```
PS F:\laragon\www\Semester 4\PWL25\Minggu11\PWL_POS> php artisan make:migration add_image_to_m_barang_table
[INFO] Migration [F:\laragon\www\Semester 4\PWL25\Minggu11\PWL_POS\database\migrations\2025_04_30_215626_add_image_to_m_barang_table.php] created successfully.
```

- Tambahkan function/kode program pada file migration yang telah dibuat

```
Minggu11 > PWL_POS > database > migrations > 2025_04_30_215626_add_image_to_m_barang_table.php > ...
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::table('m_barang', function (Blueprint $table) {
15             $table->string('image')->nullable();
16         });
17     }
18
19     /**
20      * Reverse the migrations.
21      */
22     public function down(): void
23     {
24         Schema::table('m_barang', function (Blueprint $table) {
25             $table->dropColumn('image');
26         });
27     }
28 }
```

- Jalankan migrasi

```
PS F:\laragon\www\Semester 4\PWL25\Minggu11\PWL_POS> php artisan migrate
[INFO] Running migrations.
2025_04_30_215626_add_image_to_m_barang_table ..... 951ms DONE
```

	barang_id	kategori_id	barang_kode	barang_nama	harga_beli	harga_jual	created_at	updated_at	image
<input type="checkbox"/>	1	1	ELT-001	Samsung Smart TV 32 inch	3000000	3500000	NULL	2025-04-17 10:43:05	NULL
<input type="checkbox"/>	2	1	ELT-002	Air Conditioner Sharp	4000000	4200000	NULL	2025-03-20 15:46:57	NULL

- Modifikasi pada file BarangModel

```
Minggu11 > PWL_POS > app > Models > BarangModel.php > PHP > BarangModel > image()
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\Database\Eloquent\Concerns\HasAttributes;
9 use Tymon\JWTAuth\Contracts\JWTSubject;
10
11 /**
12  * @property string $table = "m_barang";
13  */
14 class BarangModel extends Model
15 {
16     use HasFactory;
17
18     /**
19      * The table associated with the model.
20      */
21     protected $table = "m_barang";
22
23     /**
24      * The primary key for the model.
25      */
26     protected $primaryKey = "barang_id";
27
28     /**
29      * The attributes that are mass assignable.
30      */
31     protected $fillable = [
32         'kategori_id',
33         'barang_kode',
34         'barang_nama',
35         'harga_beli',
36         'harga_jual',
37         'image'
38     ];
39
40     /**
41      * The attributes that should be hidden for serialization.
42      */
43     protected $hidden = [
44         'password',
45         'remember_token',
46     ];
47
48     /**
49      * The attributes that should be cast to native types.
50      */
51     protected $casts = [
52         'password' => 'string',
53     ];
54
55     /**
56      * Get the category that belongs to the barang.
57      */
58     public function kategori(): BelongsTo
59     {
60         return $this->belongsTo(KategoriModel::class, 'kategori_id', 'kategori_id');
61     }
62
63     /**
64      * Get the image attribute.
65      */
66     public function image(): Attribute
67     {
68         return Attribute::make(
69             get: fn ($image) => url(path: './storage/barang/' . $image),
70         );
71     }
72 }
```



- Modifikasi function pada file Api/BarangController

```
1 <?php
2
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\BarangModel;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Validator;
9
10 class BarangController extends Controller
11 {
12     2 references | 0 overrides
13     public function index(): JsonResponse|mixed
14     {
15         $barang = BarangModel::all();
16         return response()->json([
17             'success' => true,
18             'data' => $barang,
19         ]);
20     }
21
22     2 references | 0 overrides
23     public function store(Request $request): JsonResponse|mixed
24     {
25         $validator = Validator::make($request->all(), [
26             'kategori_id' => 'required|exists:m_kategori,kategori_id',
27             'barang_kode' => 'required|unique:m_barang,barang_kode',
28             'barang_nama' => 'required',
29             'harga_beli' => 'required|numeric',
30             'harga_jual' => 'required|numeric',
31             'image' => 'nullable|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
32         ]);
33         if ($validator->fails()) {
34             return response()->json([
35                 'success' => false,
36                 'message' => $validator->errors->first(),
37             ], 400);
38         }
39
40         $barang = new BarangModel();
41         $barang->kategori_id = $request->kategori_id;
42         $barang->barang_kode = $request->barang_kode;
43         $barang->barang_nama = $request->barang_nama;
44         $barang->harga_beli = $request->harga_beli;
45         $barang->harga_jual = $request->harga_jual;
46         $barang->image = $request->image;
47         $barang->save();
48
49         return response()->json([
50             'success' => true,
51             'data' => $barang,
52         ], 201);
53     }
54 }
```

\*Selengkapnya : [PWL25/Minggu11/PWL\\_POS/app/Http/Controllers/Api/BarangController.php](https://github.com/vitasaraswati/PWL25/blob/main/PWL_POS/app/Http/Controllers/Api/BarangController.php)  
at [a266073cba8990ebf4936f6411a965d3eec83379](https://github.com/vitasaraswati/PWL25) · vitasaraswati/PWL25 · GitHub

- Test pada aplikasi Postman
  - Create data barang dengan method POST

POST localhost/Semester%204/PWL25/Minggu11/PWL\_POS/public/api/barangs

Params Authorization Headers (8) Body Scripts Settings

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

Key	Value	Description
barang_nama	Gulaku Gula Pasir 1kg	
harga_beli	16500	
harga_jual	20000	
image	gulaku_premium_sugar_1kg.png	
barang_kode	SBK-001	
kategori_id	14	

Body Cookies Headers (11) Test Results

JSON

```
1 {
2   "success": true,
3   "data": {
4     "kategori_id": "14",
5     "barang_kode": "SBK-001",
6     "barang_nama": "Gulaku Gula Pasir 1kg",
7     "harga_beli": "16500",
8     "harga_jual": "20000",
9     "image": "http://localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/storage/barang/01Ak2S2GfAgIngeRLEK35QottuJvbE11cqhVv1jb.png",
10    "updated_at": "2025-04-30T15:43:20.000000Z",
11    "created_at": "2025-04-30T15:43:20.000000Z",
12    "barang_id": 39
13  }
14 }
```

201 Created · 1.68 s · 765 B · Save Response

Activate Windows  
Go to Settings to activate Windows.

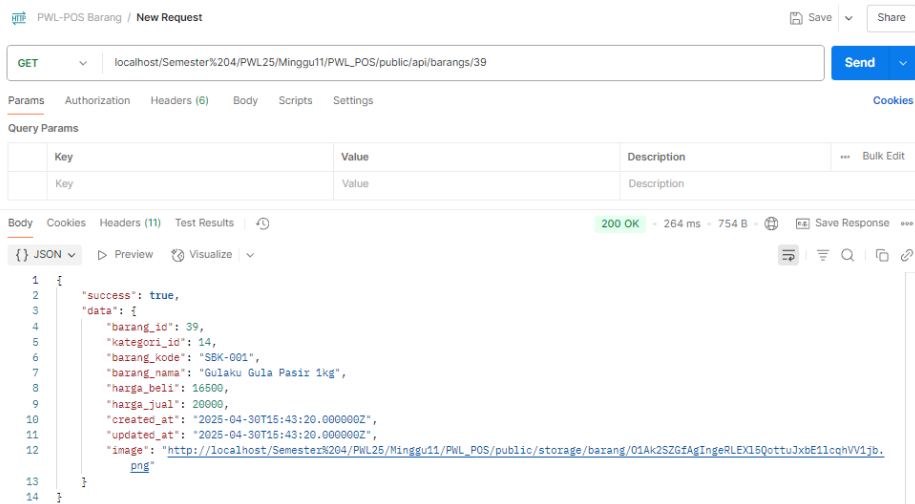
Data berhasil tersimpan

<input type="checkbox"/>				39	14 SBK-001	Gulaku Gula Pasir 1kg	16500	20000	2025-04-30 22:43:20	2025-04-30 22:43:20	01Ak2S2GfAgIngeRLEK35QottuJvbE11cqhVv1jb.png
--------------------------	--	--	--	----	------------	-----------------------	-------	-------	---------------------	---------------------	--



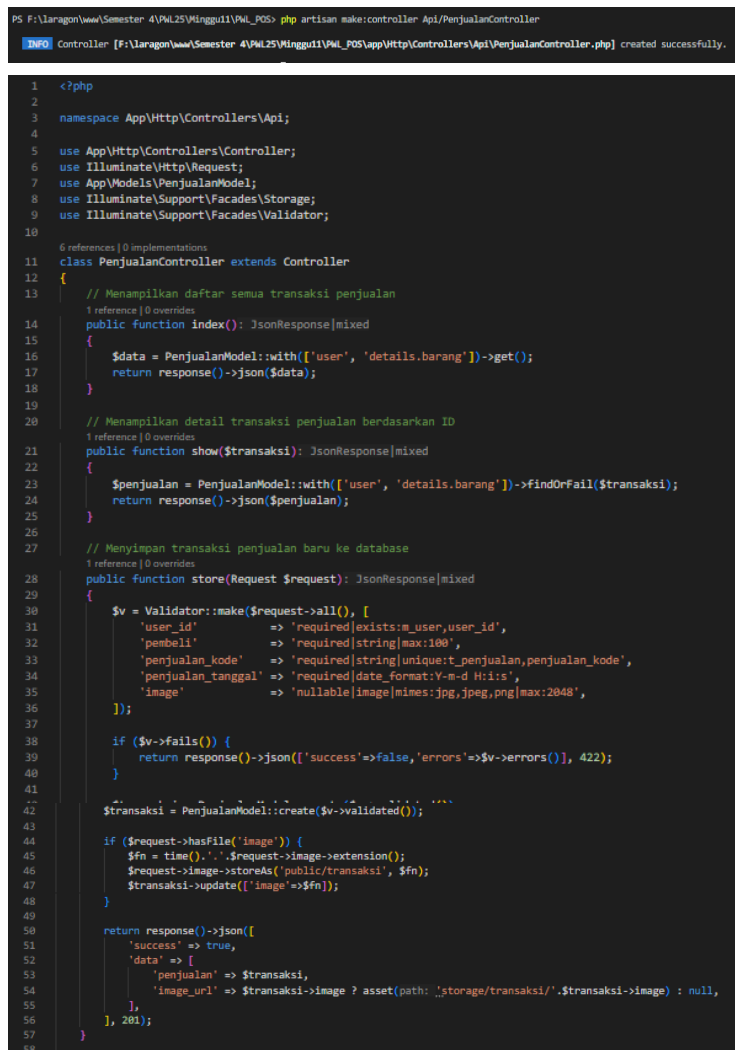


- Read/menampilkan data barang dengan method GET



## b. Tabel t\_penjualan

- Buat controller Api/PenjualanController





*\*Selengkapnya:*

[PWL25/Minggu11/PWL\\_POS/app/Http/Controllers/Api/PenjualanController.php at 2cda27738398fbc3ead2b1435b2d868798479519 · vitaraswati/PWL25 · GitHub](#)

- Buat migrasi untuk menambahkan column image

```
PS F:\Laragon\www\Semester 4\PWL25\Minggu11\PWL_POS> php artisan make:migration add_image_to_t_penjualan_table
[INFO] Migration [F:\Laragon\www\Semester 4\PWL25\Minggu11\PWL_POS\database\migrations\2025_04_30_232804_add_image_to_t_penjualan_table.php] created successfully.
```

- Tambahkan function/kode program pada file migration yang telah dibuat

```
Minggu11 > PWL_POS > database > migrations > 2025_04_30_232804_add_image_to_t_penjualan_table.php > ...

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12      public function up(): void
13      {
14          Schema::table('t_penjualan', function (Blueprint $table): void {
15              $table->string('image')->nullable()->after('penjualan_tanggal');
16          });
17      }
18
19      /**
20       * Reverse the migrations.
21       */
22      public function down(): void
23      {
24          Schema::table('t_penjualan', function (Blueprint $table): void {
25              $table->dropColumn('image');
26          });
27      }
28  };

```

- Jalankan migrasi

```
PS F:\Laragon\www\Semester 4\PWL25\Minggu11\PWL_POS> php artisan migrate
[INFO] Running migrations.
2025_04_30_232804_add_image_to_t_penjualan_table ..... 419ms DONE

Showing rows 0 - 21 (22 total, Query took 0.0015 seconds.)
SELECT * FROM `t_penjualan`
Profiling [ Edit inline ] [ Edit ] [ Explain SQL ] [ Create PHP code ] [ Refresh ]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

+---+
| penjualan_id | user_id | pembeli | penjualan_kode | penjualan_tanggal | image | created_at | updated_at |
|---|---|---|---|---|---|---|---|
| 1 | 29 | Sofyan W | PNJ-001 | 2025-04-07 09:23:00 | NULL | NULL | 2025-04-17 00:23:32
```

- Modifikasi pada file PenjualanModel

```
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\Database\Eloquent\Relations\HasMany;
9  use App\Models\UserModel;
10 use App\Models\DetailPenjualanModel;
11 use Illuminate\Database\Eloquent\casts\Attribute;
12 use Tymon\JWTAuth\Contracts\JWTSubject;
13 use Illuminate\Foundation\Auth\User as Authenticatable; //implementasi class Authenticatable
14
15 class PenjualanModel extends Model
16 {
17     0 references | 0 implementations
18     0 references | 0 overrides
19     public function getJWTIdentifier(): mixed{
20         return $this->getKey();
21     }
22     use HasFactory;
23     0 references
24     protected $table = 't_penjualan';
25     0 references
26     protected $primaryKey = 'penjualan_id';
27     0 references
28     protected $fillable = ['user_id', 'pembeli', 'penjualan_kode', 'penjualan_tanggal', 'image'];
29

```



```
49 }  
50  
51 0 references | 0 overrides  
52 public function image(): Attribute  
53 {  
54     return Attribute::make(  
55         get: fn ($image): string|UrlGenerator => url(path: './storage/transaksi/' . $image),  
56     );  
57 }
```

- Menambahkan route pada file api.php

```
10 use App\Http\Controllers\Api\PenjualanController;  
  
64 //Route API Transaksi Penjualan  
65 Route::get('/penjualans', [PenjualanController::class, 'index']);  
66 Route::post('/penjualans', [PenjualanController::class, 'store']);  
67 Route::get('/penjualans/{penjualan}', [PenjualanController::class, 'show']);  
68 Route::put('/penjualans/{penjualan}', [PenjualanController::class, 'update']);  
69 Route::delete('/penjualans/{penjualan}', [PenjualanController::class, 'destroy']);
```

- Test pada aplikasi Postman
  - Create data transaksi dengan method POST

The screenshot shows a Postman interface for a POST request to `localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/api/penjualans`. The request body is in JSON format with the following data:

Key	Value	Description
user_id	37	
pembeli	Nabilla	
penjualan_kode	PNJ-023	
penjualan_tanggal	2025-05-01 00:00:00	
image	nota_trn_01_05_2025.png	

The response is a 201 Created status with a JSON body:

```
{  
  "success": true,  
  "data": {  
    "penjualan": {  
      "user_id": "37",  
      "pembeli": "Nabilla",  
      "penjualan_kode": "PNJ-023",  
      "penjualan_tanggal": "2025-05-01 00:00:00",  
      "image": "http://localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/storage/transaksi/1746832672.png",  
      "updated_at": "2025-04-30T17:04:34.000000Z",  
      "created_at": "2025-04-30T17:04:32.000000Z",  
      "penjualan_id": 67  
    }  
  }  
}
```

- Read/menampilkan data transaksi dengan method GET

The screenshot shows a Postman interface for a GET request to `localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/api/penjualans?penjualan_kode=67&pembeli=Nabilla`. The response is a 200 OK status with a JSON body:

```
{  
  "penjualan_id": 67,  
  "user_id": 37,  
  "pembeli": "Nabilla",  
  "penjualan_kode": "PNJ-023",  
  "penjualan_tanggal": "2025-05-01 00:00:00",  
  "image": "http://localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/storage/transaksi/1746832672.png",  
  "created_at": "2025-04-30T17:04:32.000000Z",  
  "updated_at": "2025-04-30T17:04:34.000000Z",  
  "user": {  
    "user_id": 37,  
    "level_id": 3,  
    "profile_photo": null,  
    "username": "staff03",  
    "nama": "STF_Kevin",  
    "created_at": "2025-04-30T14:03:29.000000Z",  
    "updated_at": "2025-04-30T14:03:29.000000Z",  
    "image": "http://localhost/Semester%204/PWL25/Minggu11/PWL_POS/public/storage/posts/w0Fy81npAJoTo250d0eE6nxo6p1SPMe3NND5aHkg.jpg"  
  }  
}
```