



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

Nama : Dea Putri Nastiti
NIM : 2241720117
Kelas : TI 2H

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 ditabel 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_05_14_110145_add_image_to_m_user_table.php]
created successfully.
```

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

```
database > migrations > 2024_05_14_110145_add_image_to_m_user_table.php > class > down > Closure
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) {
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) {
25             $table->dropColumn('image');
26         });
27     }
28 };
```

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

```
PS C:\laragon\www\PWL_POS> php artisan migrate
INFO Running migrations.
2024_05_14_110145_add_image_to_m_user_table ..... 348ms DONE
```

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

```
app > Models > UserModel.php > UserModel
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 Tymon\JWTAuth\Contracts\JWTSubject;
10 use Illuminate\Database\Eloquent\Casts\Attribute;
11 use Illuminate\Foundation\Auth\User as Authenticatable;
12
```



```
13 class UserModel extends Authenticatable implements JWTSubject
14 {
15     public function getJWTIdentifier()
16     {
17         return $this->getKey();
18     }
19
20     public function getJWTCustomClaims()
21     {
22         return [];
23     }
24
25     use HasFactory;
26
27     protected $table = 'm_user'; //Mendefinisikan nama tabel yang digunakan oleh model ini
28     protected $primaryKey = 'user_id'; //mendefinisikan primary key dari tabel yang digunakan
29     //protected $fillable = ['level_id', 'username', 'nama', 'password'];
30     protected $fillable = ['level_id', 'username', 'nama', 'password', 'image'];
31
32     public function level():BelongsTo{
33         return $this->belongsTo(LevelModel::class, 'level_id', 'level_id');
34     }
35
36     protected function image(): Attribute
37     {
38         return Attribute::make(
39             get: fn ($image) => url('/storage/posts/' . $image),
40         );
41     }
42 }
```

6. Lakukan modifikasi pada Controllers/Api/RegisterController

```
app > Http > Controllers > Api > RegisterController.php > RegisterController > __invoke
1 <?php
2
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\UserModel;
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 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',
21         ]);
22
23         //if validations fails
24         if($validator->fails()){
25             return response()->json($validator->errors(), 422);
26         }
27
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
35         ]);
36     }
37 }
```



```
30
31
32
33
34
35
36
37 //return response JSON user is created
38 if($user){
39     return response()->json([
40         'success' => true,
41         'user' => $user,
42     ], 201);
43 }
44
45 //return JSON process insert failed
46 return response()->json([
47     'succes' => false,
48     ], 409);
49 }
50 }
51
```

7. Anda dapat menambahkan detail untuk spesifikasi image pada validator

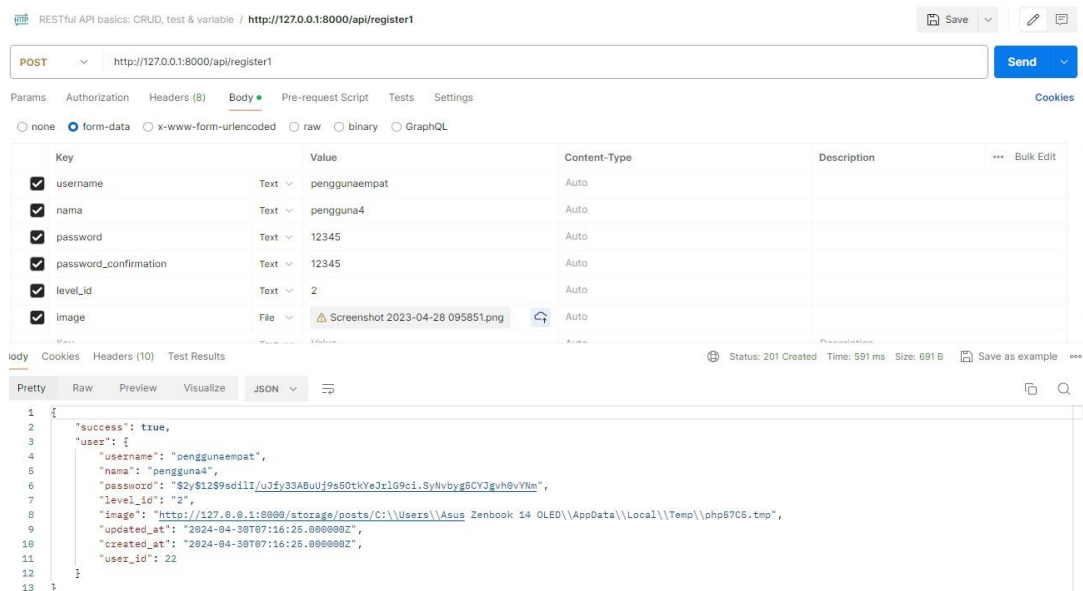
```
$validator = Validator::make($request->all(), [
    'username' => 'required',
    'nama' => 'required',
    'password' => 'required|min:5|confirmed',
    'level_id' => 'required',
    'image' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
]);
```

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

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

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





POST localhost/PWL_POS/public/api/register1 Send

Params Auth Headers (8) Body Scripts Settings Cookies

form-data

Key	Value	Bulk Edit
<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 kisspng-scooter-motorcycle-c...	

Body 201 Created 1045 ms 747 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "user": {
4     "username": "penggunaempat",
5     "nama": "pengguna4",
6     "password": "$2y$12$cGx4jZYPmUY3NbK0WxqjLe5pb0RI6yLU7x1X5xYpDSRCrdTSgrId.",
7     "level_id": "2",
8     "image": "http://localhost/PWL_POS/public/storage/posts/
9       C:\\Users\\user\\AppData\\Local\\Temp\\phpA3AE.tmp",
10    "updated_at": "2024-05-15T05:50:25.000000Z",
11    "created_at": "2024-05-15T05:50:25.000000Z",
12    "user_id": 21
13  }
```

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 ]);
```



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

POST localhost/PWL_POS/public/api/register1

Send

Params Auth Headers (8) Body Scripts Settings Cookies

form-data

Key	Value
username	penggunalima
nama	pengguna5
password	12345
password_confirmation	12345
level_id	2
image	kisspng-scooter-motorcycle-c...

Body

201 Created 808 ms 740 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "success": true,
3   "user": {
4     "username": "penggunalima",
5     "nama": "pengguna5",
6     "password": "$2y$12$.KDCeVa0WBRzF9Kp9N1CM0DH6JuzRAfUTHcY0ae6bRLmgzFN6tZC",
7     "level_id": "2",
8     "image": "http://localhost/PWL_POS/public/storage/posts/8FGqnmhokFMy1fnEgJSe3hy0q1KUJLuTYjQDSGR5.jpg",
9     "updated_at": "2024-05-15T05:57:42.000000Z",
10    "created_at": "2024-05-15T05:57:42.000000Z",
11    "user_id": 23
12  }
13 }
```

Perbedaan dari percobaan sebelumnya adalah pada percobaan 9 tidak menggunakan hash sehingga nama file image yang di upload terlihat jelas. Sedangkan pada percobaan 11 menggunakan hash sehingga nama file disamarkan

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
 - Migration

```
PS C:\laragon\www\PWL_POS> php artisan make:migration add_image_to_m_barang_table
[INFO] Migration [C:\laragon\www\PWL_POS\database\Migrations\2024_05_15_061750_add_image_to_m_barang_table.php] created successfully.
```



```
database > migrations > 2024_05_15_061750_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');
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 };

PS C:\laragon\www\PWL_POS> php artisan migrate

INFO Running migrations.

2024_05_15_061750_add_image_to_m_barang_table ..... 161ms DONE
```

- BarangModel

```
app > Models > BarangModel.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\Database\Eloquent\Relations\HasMany;
9  use Illuminate\Database\Eloquent\Casts\Attribute;
10
11
12 class BarangModel extends Model
13 {
14     protected $table = 'm_barang';
15     protected $primaryKey = 'barang_id';
16
17     protected $fillable = ['kategori_id', 'barang_kode', 'barang_nama', 'harga_beli', 'harga_jual', 'image'];
18
19     protected function image(): Attribute
20     {
21         return Attribute::make(
22             get: fn ($image) => url('/storage/posts/' . $image),
23         );
24     }
25 }
```




- BarangController

```
app > Http > Controllers > Api > BarangController.php > BarangController > update
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use App\Models\BarangModel;
8
9  class BarangController extends Controller
10 {
11     public function index(){
12         return BarangModel::all();
13     }
14
15     public function store(Request $request){
16         $request->validate([
17             'barang_kode' => 'required|string|min:3|unique:m_barang,barang_kode',
18             'barang_nama' => 'required|string|max:100',
19             'harga_beli'  => 'required|integer',
20             'harga_jual'  => 'required|integer',
21             'kategori_id' => 'required|integer',
22             'image'       => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
23         ]);
24
25         $barang = BarangModel::create([
26             'barang_kode' => $request->barang_kode,
27             'barang_nama' => $request->barang_nama,
28             'harga_beli'  => $request->harga_beli,
29             'harga_jual'  => $request->harga_jual,
30             'kategori_id' => $request->kategori_id,
31             'image'       => $request->image->hashName(),
32         ]);
33         return response()->json($barang, 201);
34     }
35
36     public function show(BarangModel $barang){
37         return BarangModel::find($barang);
38     }
39
40     public function update(Request $request, BarangModel $barang){
41         $request->validate([
42             'barang_kode' => 'required|string|min:3|unique:m_barang,barang_kode',
43             'barang_nama' => 'required|string|max:100',
44             'harga_beli'  => 'required|integer',
45             'harga_jual'  => 'required|integer',
46             'kategori_id' => 'required|integer',
47             'image'       => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
48         ]);
49
50         $barang->update([
51             'barang_kode' => $request->barang_kode,
52             'barang_nama' => $request->barang_nama,
53             'harga_beli'  => $request->harga_beli,
54             'harga_jual'  => $request->harga_jual,
55             'kategori_id' => $request->kategori_id,
56             'image'       => $request->image->hashName(),
57         ]);
58         return BarangModel::find($barang->barang_id);
59     }
60
61     public function destroy(BarangModel $barang){
62         $barang->delete();
63
64         return response()->json([
65             'succes' => true,
66             'message' => 'Data terhapus',
67         ]);
68     }
69 }
```

- api.php


```
49 //barang
50 Route::get('barangs', [BarangController::class, 'index']);
51 Route::post('barangs', [BarangController::class, 'store']);
52 Route::get('barangs/{barang}', [BarangController::class, 'show']);
53 Route::put('barangs/{barang}', [BarangController::class, 'update']);
54 Route::delete('barangs/{barang}', [BarangController::class, 'destroy']);
55
```



POST localhost/PWL_POS/public/api/barangs Send

Params Auth Headers (8) Body Scripts Settings Cookies

form-data

<input checked="" type="checkbox"/>	barang_kode	Text	b11
<input checked="" type="checkbox"/>	barang_nama	Text	Pensil
<input checked="" type="checkbox"/>	harga_beli	Text	3000
<input checked="" type="checkbox"/>	harga_jual	Text	4000
<input checked="" type="checkbox"/>	kategori_id	Text	7
<input checked="" type="checkbox"/>	image	File	
	Key	Text	Value

body 201 Created 529 ms 685 B Save as example


Pretty Raw Preview Visualize JSON

```
1 {
2   "barang_kode": "b11",
3   "barang_nama": "Pensil",
4   "harga_beli": "3000",
5   "harga_jual": "4000",
6   "kategori_id": "7",
7   "image": "http://localhost/PWL_POS/public/storage/posts/
8     n2lC3qXZjUAmYyhS5wxUC08of0511285b8y7VFZ3.jpg",
9   "updated_at": "2024-05-15T07:25:02.000000Z",
10  "created_at": "2024-05-15T07:25:02.000000Z",
11  "barang_id": 13
12 }
```

GET localhost/PWL_POS/public/api/barangs/13 Send

Params Auth Headers (8) Body Scripts Settings Cookies

form-data

<input checked="" type="checkbox"/>	barang_kode	Text	b11
<input checked="" type="checkbox"/>	barang_nama	Text	Pensil
<input checked="" type="checkbox"/>	harga_beli	Text	3000
<input checked="" type="checkbox"/>	harga_jual	Text	4000
<input checked="" type="checkbox"/>	kategori_id	Text	7
<input checked="" type="checkbox"/>	image	File	
	Key	Text	Value

body 200 OK 453 ms 883 B Save as example

Pretty Raw Preview Visualize JSON

```
12 {,
13 {
14   "barang_id": 13,
15   "kategori_id": 7,
16   "barang_kode": "b11",
17   "barang_nama": "Pensil",
18   "harga_beli": 3000,
19   "harga_jual": 4000,
20   "created_at": "2024-05-15T07:25:02.000000Z",
21   "updated_at": "2024-05-15T07:25:02.000000Z",
22   "image": "http://localhost/PWL_POS/public/storage/posts/
23     n2lC3qXZjUAmYyhS5wxUC08of0511285b8y7VFZ3.jpg"
24 }
```



2. transaksi

INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\Api\TransaksiController.php] created successfully.

- TransaksiController

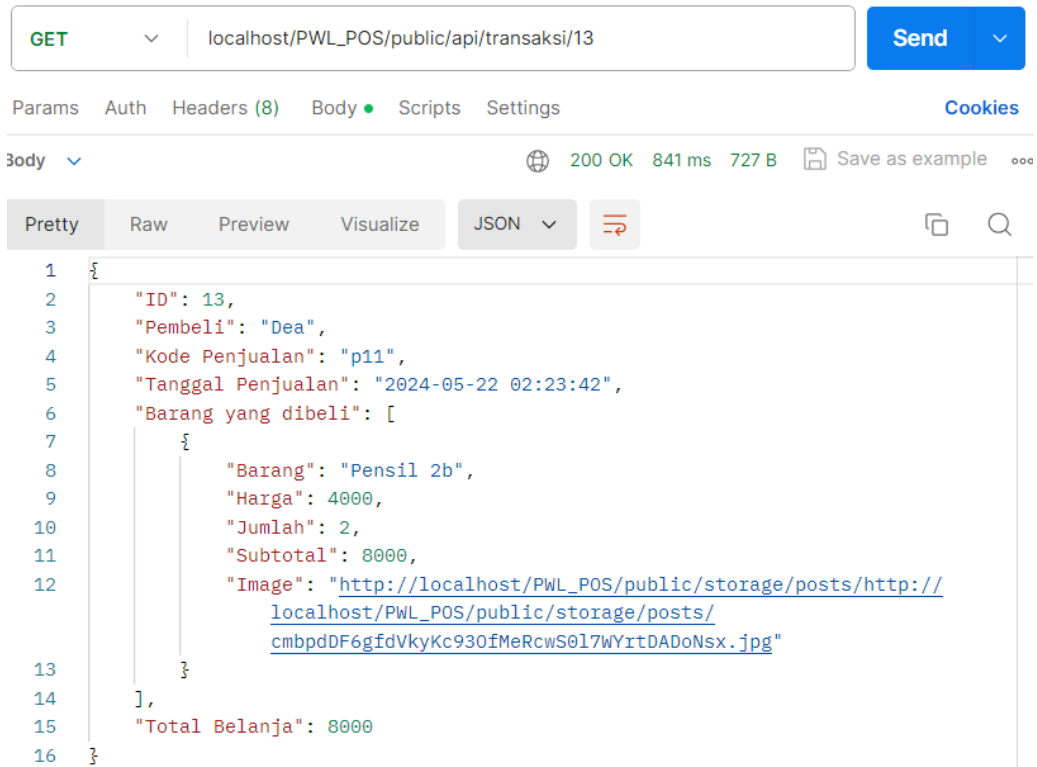
```
app > Http > Controllers > Api > TransaksiController.php > TransaksiController > show
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use App\Models\PenjualanModel;
8
9
10 class TransaksiController extends Controller
11 {
12     public function show($id)
13     {
14         $penjualan = PenjualanModel::with('detail.barang')->findOrFail($id);
15
16         $response = [
17             'ID' => $penjualan->penjualan_id,
18             'Pembeli' => $penjualan->pembeli,
19             'Kode Penjualan' => $penjualan->penjualan_kode,
20             'Tanggal Penjualan' => $penjualan->penjualan_tanggal,
21             'Barang yang dibeli' => [],
22             'Total Belanja' => 0,
23         ];
24
25         foreach ($penjualan->detail as $item) {
26             $barang = $item->barang;
27             $subtotal = $item->harga * $item->jumlah;
28             $response['Barang yang dibeli'][] = [
29                 'Barang' => $barang->barang_nama,
30                 'Harga' => $item->harga,
31                 'Jumlah' => $item->jumlah,
32                 'Subtotal' => $subtotal,
33                 'Image' => url('storage/posts/' . $barang->image),
34             ];
35             $response['Total Belanja'] += $subtotal;
36         }
37
38         return response()->json($response, 200);
39     }
40 }
```

- api.php

```
60 Route::get('transaksi/{id}', [TransaksiController::class, 'show']);
```



- postman



*** Sekian, dan selamat belajar ***