



Mata Kuliah : Pemrograman Web Lanjut (PWL)
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis
Semester : 4 (empat) / 6 (enam)
Pertemuan ke- : 10 (tujuh)

Fallujah Ramadi Camshah – 09

SIB 2B

JOBSHEET 10

RESTFUL API

Sebelumnya kita sudah membahas mengenai *authentication*, *authorization*, dan *middleware* pada Laravel. Dimana kita telah membuat fungsi login, register, logout, serta pemilihan role dan penerapan session pada halaman web. Pada pertemuan kali ini, kita akan mempelajari penerapan RESTFUL API 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. RESTFUL API

Representational State Transfer (REST) adalah gaya arsitektur perangkat lunak yang mendefinisikan seperangkat prinsip untuk merancang jaringan aplikasi terdistribusi. RESTful API adalah aplikasi pemrograman antarmuka yang mengikuti prinsip-prinsip REST untuk mentransfer data antara klien dan server.

RESTful API adalah salah satu arsitektur dalam API (*Application Program Interface*) yang menggunakan request HTTP untuk mengakses data. Data diakses dengan menggunakan HTTP method GET, PUT, POST dan DELETE yang merujuk pada operasi pembacaan, pembaruan, pembuatan dan penghapusan pada resource. Selain HTTP method, dalam RESTful atau REST digunakan juga HTTP response untuk mendefinisikan respon data yang dikembalikan. Format respon yang umum digunakan berupa JSON (Javascript Object Notation).



B. JSON Web Token (JWT)

JWT adalah singkatan dari JSON Web Token. Ini adalah standar terbuka (RFC 7519) yang mendefinisikan format token yang kompak dan mandiri untuk mentransfer klaim antara dua pihak. JWT sering digunakan dalam otentikasi dan pertukaran informasi yang aman di lingkungan yang tidak terpercaya, seperti internet.

JWT terdiri dari tiga bagian yang dipisahkan oleh titik ("."): header, payload, dan signature. Setiap bagian ini terdiri dari data JSON yang dienkripsi menggunakan algoritma tertentu dan kemudian disatukan untuk membentuk token yang lengkap. Header berisi jenis token dan tipe algoritma yang digunakan untuk enkripsi. Payload berisi klaim atau informasi yang ingin disampaikan. Signature digunakan untuk memverifikasi bahwa token belum berubah dan datanya berasal dari sumber yang dipercayai.

JWT sering digunakan dalam sistem otentikasi dan otorisasi modern, seperti aplikasi web dan layanan web API, karena fleksibilitasnya dalam menyampaikan informasi terenkripsi secara ringkas.

Kita dapat menggunakan JWT untuk:

- Authentication

Ketika pengguna melakukan authentication dan mendapatkan token, maka setiap permintaan berikutnya akan menyertakan token tersebut, dan memungkinkan pengguna untuk mengakses route, service, dan resources yang diizinkan.

- Pertukaran informasi

JSON Web Token adalah cara yang baik untuk mengirimkan informasi antar pihak dengan aman. Dengan token yang sudah ditandatangani dengan algoritma RSA, maka kita bisa tahu siapa yang melakukan request tersebut.

Berikut adalah cara kerja JWT :

JWT (JSON Web Token) adalah cara untuk mentransfer informasi antara dua pihak secara aman sebagai objek JSON. Ini terdiri dari tiga bagian: header, payload, dan signature. Setelah pengguna berhasil autentikasi, server menghasilkan token JWT yang disematkan dalam permintaan HTTP. Server kemudian memvalidasi token untuk memberikan akses ke sumber daya yang diminta. Ini memberikan autentikasi yang aman dan stateless tanpa memerlukan penyimpanan status sesi di server.



Praktikum 1 – Membuat RESTful API Register

1. Sebelum memulai membuat REST API, terlebih dahulu download aplikasi Postman di <https://www.postman.com/downloads>.

Aplikasi ini akan digunakan untuk mengerjakan semua tahap praktikum pada Jobsheet ini.

2. Lakukan instalasi JWT dengan mengetikkan perintah berikut:

```
composer require tymon/jwt-auth:2.1.1
```

Pastikan Anda terkoneksi dengan internet.

3. Setelah berhasil menginstall JWT, lanjutkan dengan publish konfigurasi file dengan perintah berikut:

```
php artisan vendor:publish --  
provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"
```

4. Jika perintah di atas berhasil, maka kita akan mendapatkan 1 file baru yaitu config/jwt.php. Pada file ini dapat dilakukan konfigurasi jika memang diperlukan.

5. Setelah itu jalankan perintah berikut untuk membuat secret key JWT.

```
php artisan jwt:secret
```

Jika berhasil, maka pada file .env akan ditambahkan sebuah baris berisi nilai key JWT_SECRET.

6. Selanjutnya lakukan konfigurasi guard API. Buka config/auth.php. Ubah bagian 'guards' menjadi seperti berikut.

```
'guards' => [  
    'web' => [  
        'driver' => 'session',  
        'provider' => 'users',  
    ],  
    'api' => [  
        'driver' => 'jwt',  
        'provider' => 'users',  
    ],  
],
```

7. Kita akan menambahkan kode di model UserModel, ubah kode seperti berikut:



```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Tymon\JWTAuth\Contracts\JWTSubject;
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';
}
```

8. Berikutnya kita akan membuat controller untuk register dengan menjalankan perintah berikut.

```
php artisan make:controller Api/RegisterController
```

Jika berhasil maka akan ada tambahan controller pada folder Api dengan nama RegisterController.

9. Buka file tersebut, dan ubah kode menjadi seperti berikut.

```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Models\UserModel;
6  use App\Http\Controllers\Controller;
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 ]);
21
22 //if validations fails
23 if($validator->fails()){
24     return response()->json($validator->errors(), 422);
25 }
26
27 //create user
28 $user = UserModel::create([
29     'username' => $request->username,
30     'nama' => $request->nama,
31     'password' => bcrypt($request->password),
32     'level_id' => $request->level_id,
33 ]);
34
35 //return response JSON user is created
36 if($user){
37     return response()->json([
38         'success' => true,
39         'user' => $user,
40     ], 201);
41 }
42
43 //return JSON process insert failed
44 return response()->json([
45     'success' => false,
46 ], 409);
47 }
48 }
```

10. Selanjutnya buka routes/api.php, ubah semua kode menjadi seperti berikut.



```
<?php

use App\Http\Controllers\Api\RegisterController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/

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

11. Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman.

Buka aplikasi Postman, isi URL localhost/PWL_POS/public/api/register serta method POST. Klik Send.

POST localhost/PWL_POS/public/api/register Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body 422 Unprocessable Content 272 ms 534 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "username": [
3      "The username field is required."
4    ],
5    "nama": [
6      "The nama field is required."
7    ],
8    "password": [
9      "The password field is required."
10   ]
11 }
```

Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.



Getting started | POST http://127.0.0.1:8000/a | No environment

http://127.0.0.1:8000/api/register

POST | http://127.0.0.1:8000/api/register | Send

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

Query Params

Key	Value	Description
Key	Value	Description

Body | Cookies | Headers (9) | Test Results | 422 Unprocessable Content | 262 ms | 470 B

JSON | Preview | Visualize

```
1 {
2   "username": [
3     "The username field is required."
4   ],
5   "nama": [
6     "The nama field is required."
7   ],
8   "password": [
9     "The password field is required."
10  ],
11  "level_id": [
12    "The level id field is required."
13  ]
14 }
```

12. Sekarang kita coba masukkan data. Klik tab Body dan pilih form-data. Isikan key sesuai dengan kolom data, serta isikan data registrasi menggunakan nilai yang Anda inginkan.



POST localhost/PWL_POS/public/api/register Send

Params Auth Headers (8) **Body** Pre-req. Tests Settings Cookies

form-data

Key	Value	...	Bulk Edit
<input checked="" type="checkbox"/> username	penggunasatu		
<input checked="" type="checkbox"/> nama	Pengguna 1		
<input checked="" type="checkbox"/> password	12345		
<input checked="" type="checkbox"/> password_confirmation	12345		
<input checked="" type="checkbox"/> level_id	2		

Body Cookies Headers (11) Test Results 201 Created 624 ms 645 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {"success": true,  
2    "user": {  
3      "username": "penggunasatu",  
4      "nama": "Pengguna 1",  
5      "password": "$2y$12$Eb2SrV1jsykINytYGtrHi0DVAKcK5p6EgnZnmbChkPicIu7S0QJJU",  
6      "level_id": "2",  
7      "updated_at": "2024-04-22T15:56:04.000000Z",  
8      "created_at": "2024-04-22T15:56:04.000000Z",  
9      "user_id": 17  
10 }
```

Setelah klik tombol Send, jika berhasil maka akan keluar pesan sukses seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.

Getting started POST http://127.0.0.1:8000/a. + No environment

http://127.0.0.1:8000/api/register?username=user1&nama=Fallujah&password=12345&password_confirmation... Save

POST http://127.0.0.1:8000/api/register?username=user1&nama=Fallujah&password=12345&password_confir Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> username	user1			
<input checked="" type="checkbox"/> nama	Fallujah			
<input checked="" type="checkbox"/> password	12345			
<input checked="" type="checkbox"/> password_confirmation	12345			
<input checked="" type="checkbox"/> level_id	2			

Body Cookies Headers (9) Test Results 201 Created 366 ms 451 B

{ } JSON Preview Visualize

```
1  {  
2    "success": true,  
3    "user": {  
4      "username": "user1",  
5      "nama": "Fallujah",  
6      "level_id": "2",  
7      "updated_at": "2025-04-22T04:07:50.000000Z",  
8      "created_at": "2025-04-22T04:07:50.000000Z",  
9      "user_id": 19  
10 }  
11 }
```

13. Lakukan commit perubahan file pada Github.



Praktikum 2 – Membuat RESTful API Login

1. Kita buat file controller dengan nama LoginController.

`php artisan make:controller Api/LoginController`

Jika berhasil maka akan ada tambahan controller pada folder Api dengan nama LoginController.

2. Buka file tersebut, dan ubah kode menjadi seperti berikut.

```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Validator;
8
```



```
9  class LoginController extends Controller
10 {
11     public function __invoke(Request $request)
12     {
13         //set validation
14         $validator = Validator::make($request->all(), [
15             'username' => 'required',
16             'password' => 'required'
17         ]);
18
19         //if validation fails
20         if ($validator->fails()) {
21             return response()->json($validator->errors(), 422);
22         }
23
24         //get credentials from request
25         $credentials = $request->only('username', 'password');
26
27         //if auth failed
28         if (!$token = auth()->guard('api')->attempt($credentials)) {
29             return response()->json([
30                 'success' => false,
31                 'message' => 'Username atau Password Anda salah'
32             ], 401);
33         }
34
35         //if auth success
36         return response()->json([
37             'success' => true,
38             'user' => auth()->guard('api')->user(),
39             'token' => $token
40         ], 200);
41     }
42 }
```

3. Berikutnya tambahkan route baru pada file api.php yaitu /login dan /user.

```
use App\Http\Controllers\Api\LoginController;

Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
```

4. Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman. Buka aplikasi Postman, isi URL localhost/PWL_POS/public/api/login serta method POST. Klik Send.



POST localhost/PWL_POS/public/api/login Send

Params Auth Headers (7) Body Pre-req. Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
Key	Value	

Body Cookies Headers (11) Test Results 422 Unprocessable Content 563 ms 495 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "username": [
3     "The username field is required."
4   ],
5   "password": [
6     "The password field is required."
7   ]
8 }
```

Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.

Getting started POST http://127.0.0.1:8000/a. + No environment

http://127.0.0.1:8000/api/login Save

POST http://127.0.0.1:8000/api/login Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (9) Test Results 422 Unprocessable Content 29 ms 384 B

{ } JSON Preview Visualize

```
1 {
2   "username": [
3     "The username field is required."
4   ],
5   "password": [
6     "The password field is required."
7   ]
8 }
```

- Selanjutnya, isikan username dan password sesuai dengan data user yang ada pada database. Klik tab Body dan pilih form-data. Isikan key sesuai dengan kolom data, serta isikan data user. Klik tombol Send, jika berhasil maka akan keluar tampilan seperti berikut. Copy nilai token yang diperoleh pada saat login karena akan diperlukan pada saat logout.



POST localhost/PWL_POS/public/api/login Send

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

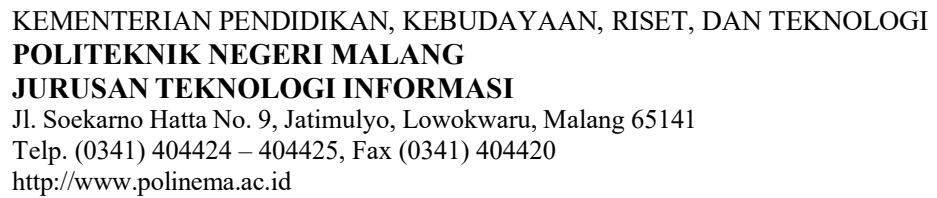
none **form-data** x-www-form-urlencoded raw binary

Key	Value	...	Bulk Edit
<input checked="" type="checkbox"/> username	penggunasatu		
<input checked="" type="checkbox"/> password	12345		
Key	Value		

Body Cookies Headers (11) Test Results Status: 200 OK Time: 1501 ms Size: 986 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "success": true,
3    "user": {
4      "user_id": 17,
5      "level_id": 2,
6      "username": "penggunasatu",
7      "nama": "Pengguna 1",
8      "password": "$2y$12$Eb2SrV1jsykINytYGtzHi0DVAKcK6p6EgnZnmbChkPicIu7S0Q3Ju",
9      "created_at": "2024-04-22T15:56:04.000000Z",
10     "updated_at": "2024-04-22T15:56:04.000000Z"
11   },
12   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ3c3MiOiJodHRwOi8vbG9jYXRob3N0L1BXTF9QT1MtbnwFpbj9wdWJsaWMyYXBpL2xvZ2luIiwiaWF0Ij0i"
13 }
```



GETTING STARTED

Getting started | POST http://127.0.0.1:8000/a... + No environment < >

http://127.0.0.1:8000/api/login?username=user1&password=12345 Save </>

POST http://127.0.0.1:8000/api/login?username=user1&password=12345 Send </>

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

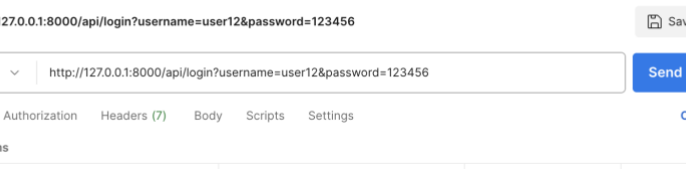
Key	Value	Description	Bulk Edit
username	user1		
password	12345		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK • 368 ms • 791 B • < >

{ JSON Preview Visualize </>

```

1 {
2   "success": true,
3   "user": {
4     "user_id": 19,
5     "level_id": 2,
6     "foto_profil": null,
7     "username": "user1",
8     "nama": "Fallujah",
9     "created_at": "2025-04-22T04:07:50.000000Z",
10    "updated_at": "2025-04-22T04:07:50.000000Z"
11  },
12  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vMTI3LjAuMC4xOjgwMDAvYXByb2xvZ2luIiwiaWF0IjoxNzQzMDM3LjCjEHAioJe3NDUzMOEwMzcsIm5iZiI6MTcONTISNzQzNywiYWianRpdIjoishSHJQNkRwQmJlMwXJOXJXaSInInN1YiI6IjE5ESIiwicHJ2IjoindFkZjg4MzRmRWMI5OGY3MGVmYTYYWFlZWGVmNDIzNDEzNzAwNmJkwYyJ9.HhkNBZWwhvwK6LvFDVylufJf1mInsZs-Tgc9YFWYwJc"
13 }
  
```



The screenshot shows the Burp Suite interface. At the top, the 'Getting started' tab is active. Below it, the 'POST' method is selected for the request to `http://127.0.0.1:8000/a`. The request body is a JSON object with 'username' and 'password' fields. The response is a 401 Unauthorized status. The 'Body' tab is selected, showing the JSON response.

Getting started POST `http://127.0.0.1:8000/a`

`http://127.0.0.1:8000/api/login?username=user12&password=123456`

POST `http://127.0.0.1:8000/api/login?username=user12&password=123456`

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

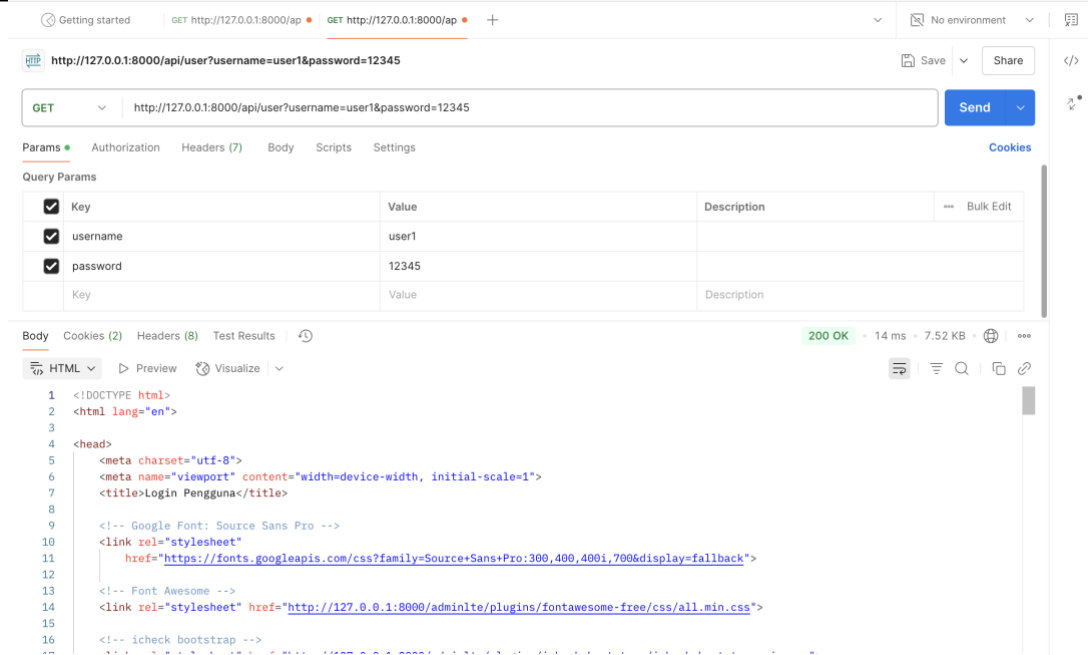
Key	Value	Description	Bulk Edit
username	user12		
password	123456		
Key	Value	Description	

Body Cookies Headers (9) Test Results 401 Unauthorized 34 ms 343 B

{ } JSON Preview Visualize

```
1 {
2   "success": false,
3   "message": "Username atau Password anda salah"
4 }
```

Hal. 13 / 29



Data user yang login tidak muncul, malah mengambil file index view login

8. Lakukan commit perubahan file pada Github.

Praktikum 3 – Membuat RESTful API Logout

1. Tambahkan kode berikut pada file `.env`
`JWT_SHOW_BLACKLIST_EXCEPTION=true`
2. Buat Controller baru dengan nama `LogoutController`.
`php artisan make:controller Api/LogoutController`
3. Buka file tersebut dan ubah kode menjadi seperti berikut.



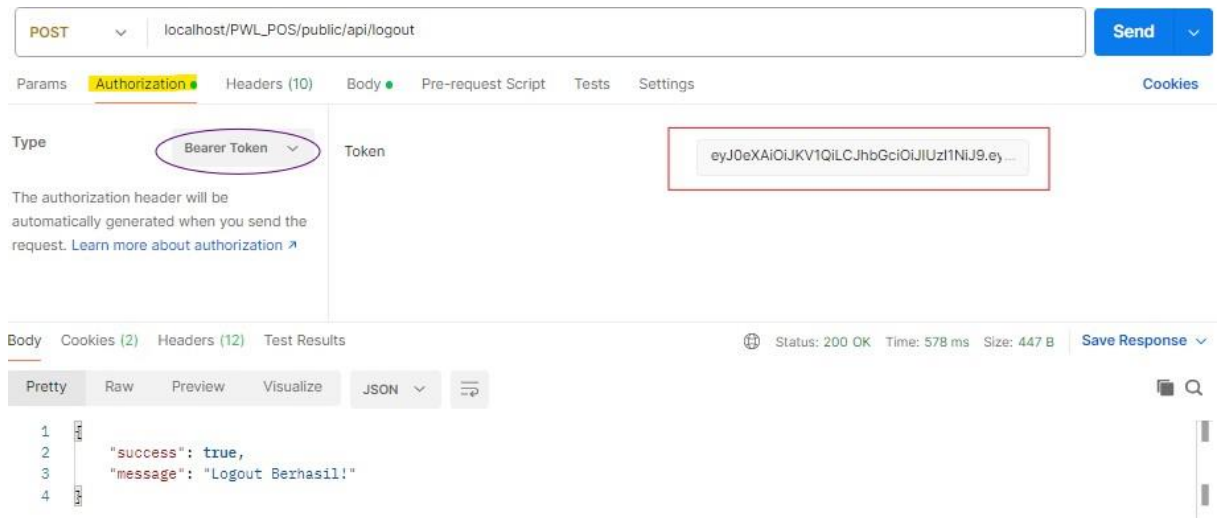
```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4  use Illuminate\Http\Request;
5  use App\Http\Controllers\Controller;
6  use Tymon\JWTAuth\Facades\JWTAuth;
7  use Tymon\JWTAuth\Exceptions\JWTException;
8  use Tymon\JWTAuth\Exceptions\TokenExpiredException;
9  use Tymon\JWTAuth\Exceptions\TokenInvalidException;
10
11 class LogoutController extends Controller
12 {
13     public function __invoke(Request $request)
14     {
15         //remove token
16         $removeToken = JWTAuth::invalidate(JWTAuth::getToken());
17
18         if($removeToken) {
19             //return response JSON
20             return response()->json([
21                 'success' => true,
22                 'message' => 'Logout Berhasil!',
23             ]);
24         }
25     }
26 }
```



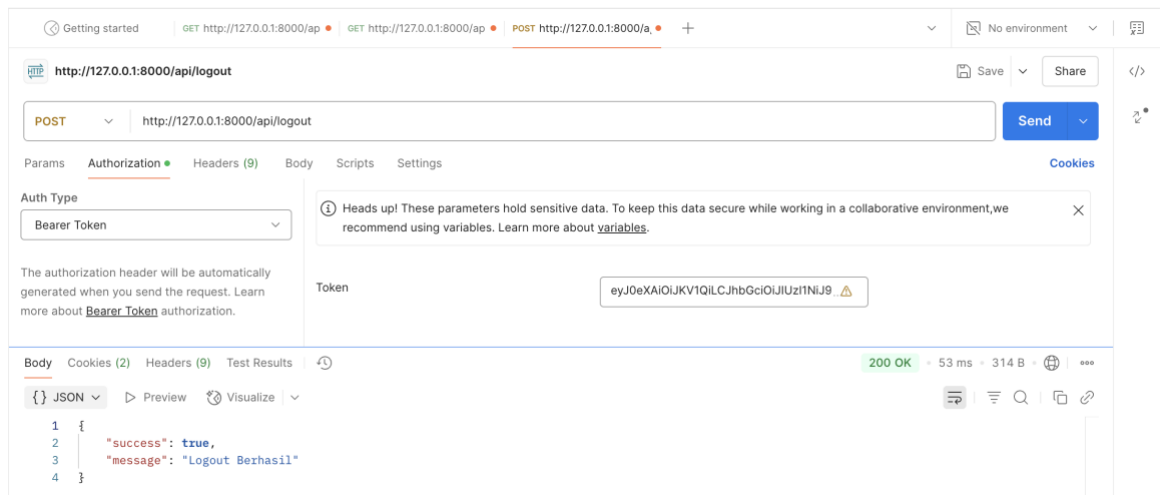
4. Lalu kita tambahkan routes pada api.php

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

5. Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman. Buka aplikasi Postman, isi URL localhost/PWL_POS/public/api/logout serta method POST.
6. Isi token pada tab Authorization, pilih Type yaitu Bearer Token. Isikan token yang didapat saat login. Jika sudah klik Send.



Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



7. Lakukan commit perubahan file pada Github.

Praktikum 4 – Implementasi CRUD dalam RESTful API

Pada praktikum ini kita akan menggunakan tabel m_level untuk dimodifikasi menggunakan RESTful API.

1. Pertama, buat controller untuk mengolah API pada data level.

```
php artisan make:controller Api/LevelController
```




2. Setelah berhasil, buka file tersebut dan tuliskan kode seperti berikut yang berisi fungsi CRUDnya.

```
namespace App\Http\Controllers\Api;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\LevelModel;

class LevelController extends Controller
{
    public function index()
    {
        return LevelModel::all();
    }
}
```



```
public function store(Request $request)
{
    $level = LevelModel::create($request->all());
    return response()->json($level, 201);
}

public function show(LevelModel $level)
{
    return LevelModel::find($level);
}

public function update(Request $request, LevelModel $level)
{
    $level->update($request->all());
    return LevelModel::find($level);
}

public function destroy(LevelModel $user)
{
    $user->delete();

    return response()->json([
        'success' => true,
        'message' => 'Data terhapus',
    ]);
}
```

3. Kemudian kita lengkapi routes pada api.php.

```
use App\Http\Controllers\Api\LevelController;

Route::get('levels', [LevelController::class, 'index']);
Route::post('levels', [LevelController::class, 'store']);
Route::get('levels/{level}', [LevelController::class, 'show']);
Route::put('levels/{level}', [LevelController::class, 'update']);
Route::delete('levels/{level}', [LevelController::class, 'destroy']);
```

4. Jika sudah. Lakukan uji coba API mulai dari fungsi untuk menampilkan data. Gunakan URL: localhost/PWL_POS-main/public/api/levels dan method GET. **Jelaskan dan berikan screenshot hasil percobaan Anda.**



Index

The screenshot shows a REST client interface with a GET request to `http://127.0.0.1:8000/api/levels`. The response is a 200 OK status with a response time of 279 ms and a body size of 971 B. The response body is a JSON array of level data.

Key	Value	Description
Key	Value	Description

```
1 [
2   {
3     "level_id": 1,
4     "level_kode": "ADM",
5     "level_nama": "Administrator",
6     "created_at": null,
7     "updated_at": null
8   },
9   {
10    "level_id": 2,
11    "level_kode": "MNG",
12    "level_nama": "Manajer",
13    "created_at": null,
14    "updated_at": null
15  },
16  {
17    "level_id": 3,
18    "level_kode": "STF",
19    "level_nama": "Staff/Kasir",
20    "created_at": null,
21    "updated_at": null
22  },
23  {
24    "level_id": 5,
25    "level_kode": "PIC"
```

5. Kemudian, lakukan percobaan penambahan data dengan URL : `localhost/PWL_POS-main/public/api/levels` dan method POST seperti di bawah ini.



POST localhost/PWL_POS/public/api/levels Send

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

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

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> level_kode	Text SPV			
<input checked="" type="checkbox"/> level_nama	Text Supervisor			
Key	Text Value	Description		

Body Cookies Headers (11) Test Results Status: 201 Created Time: 276 ms Size: 531 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "level_kode": "SPV",
3   "level_nama": "Supervisor",
4   "updated_at": "2024-04-22T21:40:32.000000Z",
5   "created_at": "2024-04-22T21:40:32.000000Z",
6   "level_id": 4
7 }
```

Jelaskan dan berikan screenshoot hasil percobaan Anda.

Getting started GET http://127.0.0.1:8000/ap GET http://127.0.0.1:8000/a POST http://127.0.0.1:8000/a POST http://127.0.0.1:8000/a No environment

http://127.0.0.1:8000/api/levels Save

POST http://127.0.0.1:8000/api/levels Send

Params Authorization Headers (9) Body Scripts Settings Cookies

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

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> level_kode	Text SPV			
<input checked="" type="checkbox"/> level_nama	Text Supervisor			
Key	Text Value	Description		

Body Cookies (2) Headers (9) Test Results 500 Internal Server Error 1.18 s 1.13 MB

HTML Preview Visualize

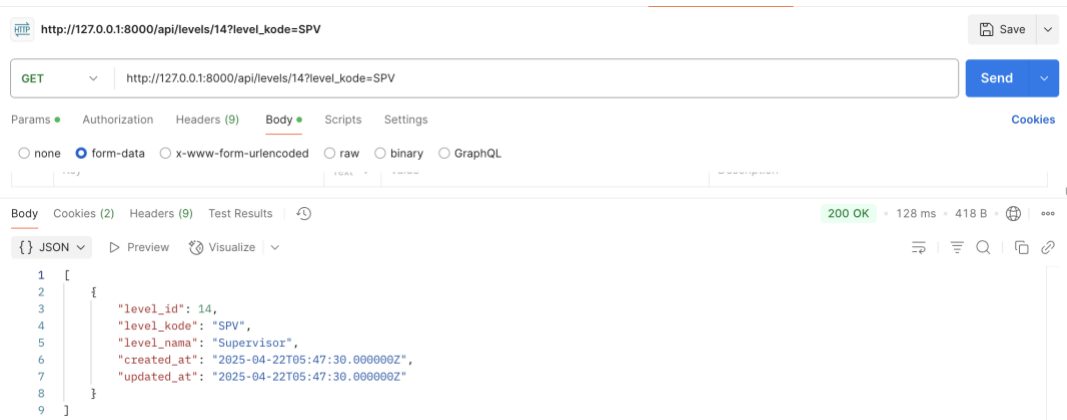
```
1 <!DOCTYPE html>
2 <html lang="en" class="auto">
3 <!--
4 Illuminate\Database\UniqueConstraintViolationException: SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry 'SPV' for ke
5
6 #0 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Connection.php(783): Illuminate\Da
7 #1 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/MySqlConnection.php(34): Illuminat
8 #2 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Query/Processors/MySqlProcessor.ph
9 #3 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Query/Builder.php(3549): Illuminat
10 #4 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Eloquent/Builder.php(1982): Illumi
11 #5 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Eloquent/Model.php(1334): Illumina
12 #6 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Eloquent/Model.php(1299): Illumina
13 #7 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Eloquent/Model.php(1138): Illumina
14 #8 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Eloquent/Builder.php(1025): Illumi
15 #9 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Support/helpers.php(320): Illuminate\Databa
16 #10 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Database/Eloquent/Builder.php(1024): tap(C
17 #11 /Applications/XAMPP/xamppfiles/htdocs/PWL_2025/week10/PWL_POS/vendor/laravel/framework/src/Illuminate/Support/Testing/AssertableCollection.php(23): Ttl
```

Ketika di send itu terjadi error, namun data nya berhasil masuk ke database

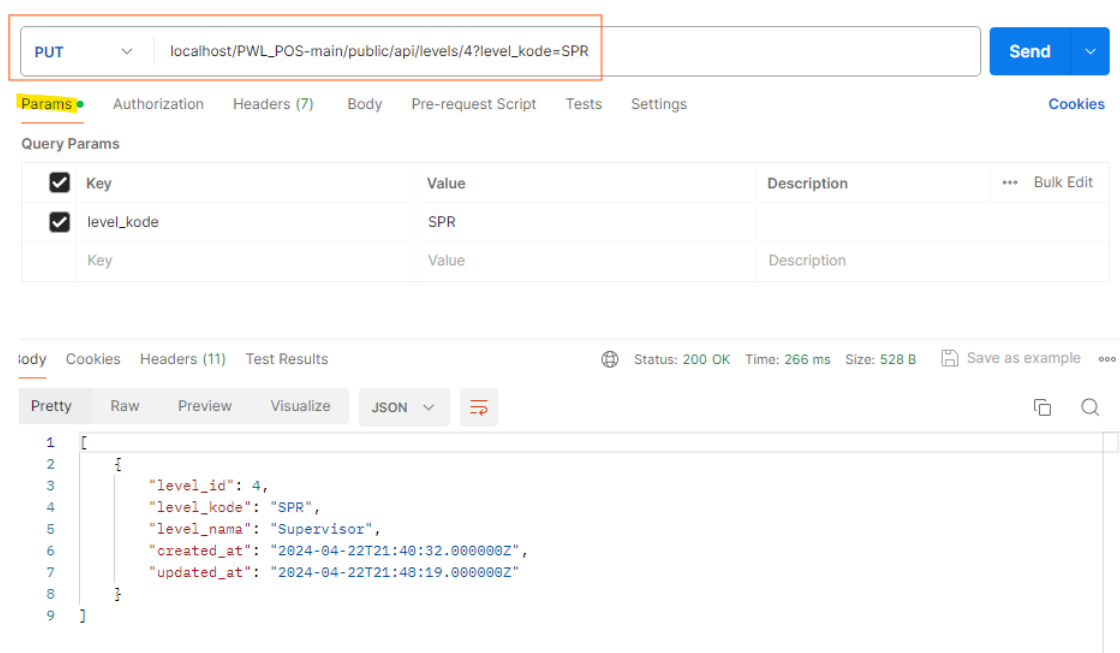
		level_id	level_kode	level_nama	created_at	updated_at
<input type="checkbox"/>	Edit Copy Delete	1	ADM	Administrator	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	2	MNG	Manajer	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	3	STF	Staff/Kasir	NULL	NULL
<input type="checkbox"/>	Edit Copy Delete	5	CUS	Pelanggan	2025-03-03 12:40:50	2025-03-24 15:00:18
<input type="checkbox"/>	Edit Copy Delete	11	TES	Testing	2025-03-26 19:48:24	2025-03-26 19:48:49
<input type="checkbox"/>	Edit Copy Delete	12	WRH	Warehouse	2025-04-09 08:07:20	NULL
<input type="checkbox"/>	Edit Copy Delete	14	SPV	Supervisor	2025-04-22 05:47:30	2025-04-22 05:47:30



6. Berikutnya lakukan percobaan menampilkan detail data. **Jelaskan dan berikan screenshot hasil percobaan Anda.**



7. Jika sudah, kita coba untuk melakukan edit data menggunakan localhost/PWL_POS-main/public/api/levels/{id} dan method PUT. Isikan data yang ingin diubah pada tab Param.



Jelaskan dan berikan screenshot hasil percobaan Anda.



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>

http://127.0.0.1:8000/api/levels/14?level_kode=SPR

PUT http://127.0.0.1:8000/api/levels/14?level_kode=SPR Send

Params Authorization Headers (9) Body Scripts Settings Cookies

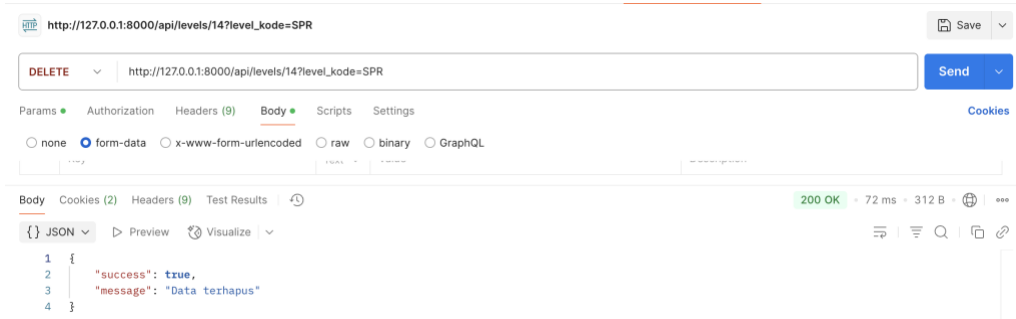
☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Body Cookies (2) Headers (9) Test Results 200 OK 135 ms 418 B

```
{
  "level_id": 14,
  "level_kode": "SPR",
  "level_nama": "Supervisor",
  "created_at": "2025-04-22T05:47:38.000000Z",
  "updated_at": "2025-04-22T06:41:38.000000Z"
}
```



8. Terakhir lakukan percobaan hapus data. **Jelaskan dan berikan screenshoot hasil percobaan Anda.**



9. Lakukan commit perubahan file pada Github.

TUGAS

Implementasikan CRUD API pada tabel lainnya yaitu tabel m_user, m_kategori, dan m_barang



m_user index

http://127.0.0.1:8000/api/users

GET http://127.0.0.1:8000/api/users

Params Authorization Headers (9) Body Scripts Settings

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

Key	Value	Description
level_id	5	
username	testUser	
nama	userTest	
password	123456	

Body Cookies Headers (9) Test Results

200 OK · 73 ms · 2.51 KB

```
{
  "updated_at": null,
  "user_id": 20,
  "level_id": 5,
  "foto_profil": null,
  "username": "testUser",
  "nama": "userTest",
  "created_at": "2025-04-23T07:23:54.000000Z",
  "updated_at": "2025-04-23T07:23:54.000000Z"
}
```

store

http://127.0.0.1:8000/api/users

POST http://127.0.0.1:8000/api/users

Params Authorization Headers (9) Body Scripts Settings

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

Key	Value	Description
level_id	5	
username	testUser	
nama	userTest	
password	123456	

Body Cookies Headers (9) Test Results

201 Created · 329 ms · 430 B

```
{
  "level_id": "5",
  "username": "testUser",
  "nama": "userTest",
  "updated_at": "2025-04-23T07:23:54.000000Z",
  "created_at": "2025-04-23T07:23:54.000000Z",
  "user_id": 20
}
```

Show

GET http://127.0.0.1:8000/ap GET http://127.0.0.1:8000/ap DEL http://127.0.0.1:8000/ap

http://127.0.0.1:8000/api/users/20?level_id=5

GET http://127.0.0.1:8000/api/users/20?level_id=5

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value	Description
level_id	5	

Body Cookies Headers (9) Test Results

200 OK · 55 ms · 442 B

```
{
  "user_id": 20,
  "level_id": 5,
  "foto_profil": null,
  "username": "testUser",
  "nama": "userTest",
  "created_at": "2025-04-23T07:23:54.000000Z",
  "updated_at": "2025-04-23T07:23:54.000000Z"
}
```




Update

PUT <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[http://127.0.0.1:8000/api/users/20?nama=UserEdit](#) Save Share

PUT <http://127.0.0.1:8000/api/users/20?nama=UserEdit> Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> nama	UserEdit		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK 115 ms 442 B

JSON Preview Visualize

```
1 {
2   "user_id": 20,
3   "level_id": 5,
4   "foto_profil": null,
5   "username": "testUser",
6   "nama": "UserEdit",
7   "created_at": "2025-04-23T07:23:54.090909Z",
8   "updated_at": "2025-04-23T07:28:46.090909Z"
9 }
```

Delete

PUT <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[http://127.0.0.1:8000/api/users/20?nama=UserEdit](#) Save

DELETE <http://127.0.0.1:8000/api/users/20?nama=UserEdit> Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

<input type="checkbox"/> Key	Value	Description	Bulk Edit
<input type="checkbox"/> level_nama	Supervisor		
<input checked="" type="checkbox"/> nama	UserEdit		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK 104 ms 317 B

JSON Preview Visualize

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



m_kategori index

PUT <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[http://127.0.0.1:8000/api/kategoris](#) Save Share

GET <http://127.0.0.1:8000/api/kategoris> Send

Params Authorization Headers (9) Body Scripts Settings Cookies

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

Key	Value	Description
<input checked="" type="checkbox"/> kategori_id	Text 12	
<input checked="" type="checkbox"/> kategori_kode	Text TEST	
<input checked="" type="checkbox"/> kategori_nama	Text kategoriTest	
Key	Value	Description

Body Cookies Headers (9) Test Results 200 OK - 109 ms - 1.82 KB

JSON Preview Visualize

```
72 {
73   "kategori_id": 11,
74   "kategori_kode": "FRG",
75   "kategori_nama": "Parfum",
76   "created_at": "2025-04-09T08:46:21.000000Z",
77   "updated_at": null
78 },
79 {
80   "kategori_id": 12,
81   "kategori_kode": "TEST",
82   "kategori_nama": "kategoriTest",
83   "created_at": "2025-04-23T07:43:45.000000Z",
84   "updated_at": "2025-04-23T07:43:45.000000Z"
85 }
86 }
```

store

PUT <http://127.0.0.1:8000/ap> POST <http://127.0.0.1:8000/a> DEL <http://127.0.0.1:8000/ap> +

[http://127.0.0.1:8000/api/kategoris](#) Save Share

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

Params Authorization Headers (9) Body Scripts Settings Cookies

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

Key	Value	Description
<input checked="" type="checkbox"/> kategori_id	Text 12	
<input checked="" type="checkbox"/> kategori_kode	Text TEST	
<input checked="" type="checkbox"/> kategori_nama	Text kategoriTest	
Key	Value	Description

Body Cookies Headers (9) Test Results 201 Created - 113 ms - 433 B

JSON Preview Visualize

```
1 {
2   "kategori_id": 12,
3   "kategori_kode": "TEST",
4   "kategori_nama": "kategoriTest",
5   "created_at": "2025-04-23T07:43:45.000000Z",
6   "updated_at": "2025-04-23T07:43:45.000000Z"
7 }
```

Show

GET <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[http://127.0.0.1:8000/api/kategoris/12?kategori_kode=TEST](#) Save Share

GET http://127.0.0.1:8000/api/kategoris/12?kategori_kode=TEST Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> kategori_kode	Text TEST	
Key	Value	Description

Body Cookies Headers (9) Test Results 200 OK - 90 ms - 428 B

JSON Preview Visualize

```
1 {
2   "kategori_id": 12,
3   "kategori_kode": "TEST",
4   "kategori_nama": "kategoriTest",
5   "created_at": "2025-04-23T07:43:45.000000Z",
6   "updated_at": "2025-04-23T07:43:45.000000Z"
7 }
```



Update

PUT http://127.0.0.1:8000/ap GET http://127.0.0.1:8000/ap DEL http://127.0.0.1:8000/ap + No environment

http://127.0.0.1:8000/api/kategoris/12?kategori_kode=TST Save Share

PUT http://127.0.0.1:8000/api/kategoris/12?kategori_kode=TST Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> kategori_kode	TST		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK 60 ms 427 B Visualize

```
1 {
2   "kategori_id": 12,
3   "kategori_kode": "TST",
4   "kategori_nama": "kategoriTest",
5   "created_at": "2025-04-23T07:43:45.000000Z",
6   "updated_at": "2025-04-23T07:47:49.000000Z"
7 }
```

Delete

PUT http://127.0.0.1:8000/ap GET http://127.0.0.1:8000/ap DEL http://127.0.0.1:8000/ap + No environment

http://127.0.0.1:8000/api/kategoris/12?kategori_kode=TST Save

DELETE http://127.0.0.1:8000/api/kategoris/12?kategori_kode=TST Send

Params Authorization Headers (8) Body Scripts Settings Cookies

Query Params

<input type="checkbox"/> Key	Value	Description	Bulk Edit
<input type="checkbox"/> level_nama	Supervisor		
<input checked="" type="checkbox"/> kategori_kode	TST		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK 38 ms 321 B Visualize

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



m_barang index

PUT <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[Save](#) [Share](#)

[http://127.0.0.1:8000/api/barangs](#)

GET [Send](#)

Params Authorization Headers (9) Body Scripts Settings Cookies

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

Key	Value	Description
barang_kode	TEST01	
barang_nama	TestBarang	
harga_beli	1000	
harga_jual	5000	

Body Cookies Headers (9) Test Results 200 OK 56 ms 3.3 KB

[JSON](#) [Preview](#) [Visualize](#)

```
168 {
169   "harga_jual": 4300,
170   "created_at": "2025-04-07T14:45:15.000000Z",
171   "updated_at": null
172 },
173 {
174   "barang_id": 19,
175   "kategori_id": 5,
176   "barang_kode": "TEST01",
177   "barang_nama": "TestBarang",
178   "harga_beli": 1000,
179   "harga_jual": 5000,
180   "created_at": "2025-04-23T07:55:22.000000Z",
181   "updated_at": "2025-04-23T07:55:22.000000Z"
182 }
```

store

PUT <http://127.0.0.1:8000/ap> POST <http://127.0.0.1:8000/a> DEL <http://127.0.0.1:8000/ap> +

[Save](#) [Share](#)

[http://127.0.0.1:8000/api/barangs](#)

POST [Send](#)

Params Authorization Headers (9) Body Scripts Settings Cookies

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

Key	Value	Description
barang_kode	TEST01	
barang_nama	TestBarang	
harga_beli	1000	
harga_jual	5000	

Body Cookies Headers (9) Test Results 201 Created 58 ms 485 B

[JSON](#) [Preview](#) [Visualize](#)

```
1 {
2   "barang_id": 19,
3   "kategori_id": "5",
4   "barang_kode": "TEST01",
5   "barang_nama": "TestBarang",
6   "harga_beli": "1000",
7   "harga_jual": "5000",
8   "updated_at": "2025-04-23T07:55:22.000000Z",
9   "created_at": "2025-04-23T07:55:22.000000Z"
10 }
```

Show

GET <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[Save](#) [Share](#)

[http://127.0.0.1:8000/api/barangs/19?barang_kode=TEST01](#)

GET [Send](#)

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description
barang_kode	TEST01	

Body Cookies Headers (9) Test Results 200 OK 58 ms 474 B

[JSON](#) [Preview](#) [Visualize](#)

```
1 {
2   "barang_id": 19,
3   "kategori_id": 5,
4   "barang_kode": "TEST01",
5   "barang_nama": "TestBarang",
6   "harga_beli": 1000,
7   "harga_jual": 5000,
8   "created_at": "2025-04-23T07:55:22.000000Z",
9   "updated_at": "2025-04-23T07:55:22.000000Z"
10 }
```



Update

PUT <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[Save](#) [Share](#)

PUT http://127.0.0.1:8000/api/barangs/19?barang_kode=BRG01 [Send](#)

Params • Authorization Headers (7) Body Scripts Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> barang_kode	BRG01		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK • 100 ms • 473 B

JSON Preview Visualize

```
1 {
2   "barang_id": 19,
3   "kategori_id": 5,
4   "barang_kode": "BRG01",
5   "barang_nama": "TestBarang",
6   "harga_beli": 1000,
7   "harga_jual": 5000,
8   "created_at": "2025-04-23T07:55:22.000000Z",
9   "updated_at": "2025-04-23T07:58:07.000000Z"
10 }
```

Delete

PUT <http://127.0.0.1:8000/ap> GET <http://127.0.0.1:8000/ap> DEL <http://127.0.0.1:8000/ap> +

[Save](#)

DELETE http://127.0.0.1:8000/api/barangs/19?barang_kode=BRG01 [Send](#)

Params • Authorization Headers (8) Body Scripts Settings Cookies

Query Params

<input type="checkbox"/> Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> barang_kode	BRG01		
<input type="checkbox"/> level_nama	Supervisor		
Key	Value	Description	

Body Cookies Headers (9) Test Results 200 OK • 70 ms • 319 B

JSON Preview Visualize

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

*** Sekian, dan selamat belajar ***