



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

**KHUZAIMA FILLA JANUARTHA**

**SIB2A / 17**

## **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',  
    ],  
],
```



```
'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';
}
```

```
public function getJWTIdentifier()
{
    return $this->getKey();
}

public function getJWTCustomClaims()
{
    return [];
}
```

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 }
```





```
<?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'
        ]);

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

        //create user
        $user = UserModel::create([
```





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

//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);
}
```

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



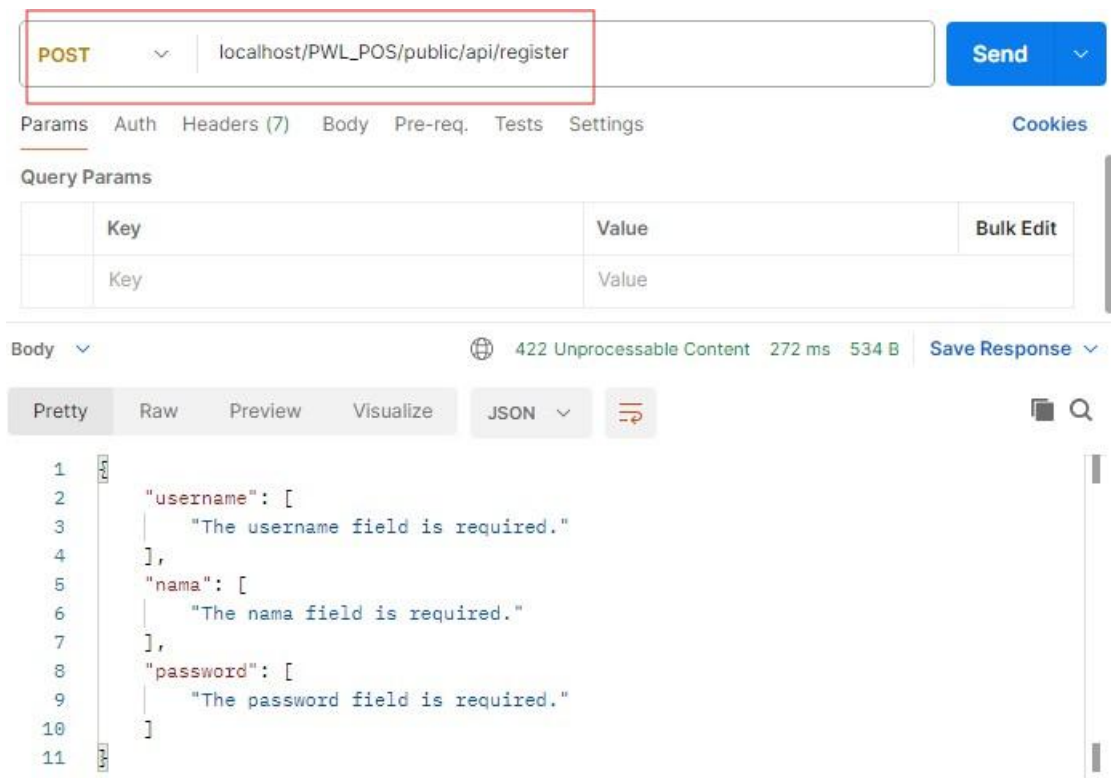
```
<?php

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.



Jika berhasil akan muncul error validasi seperti gambar di atas.

**Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.**



```
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		

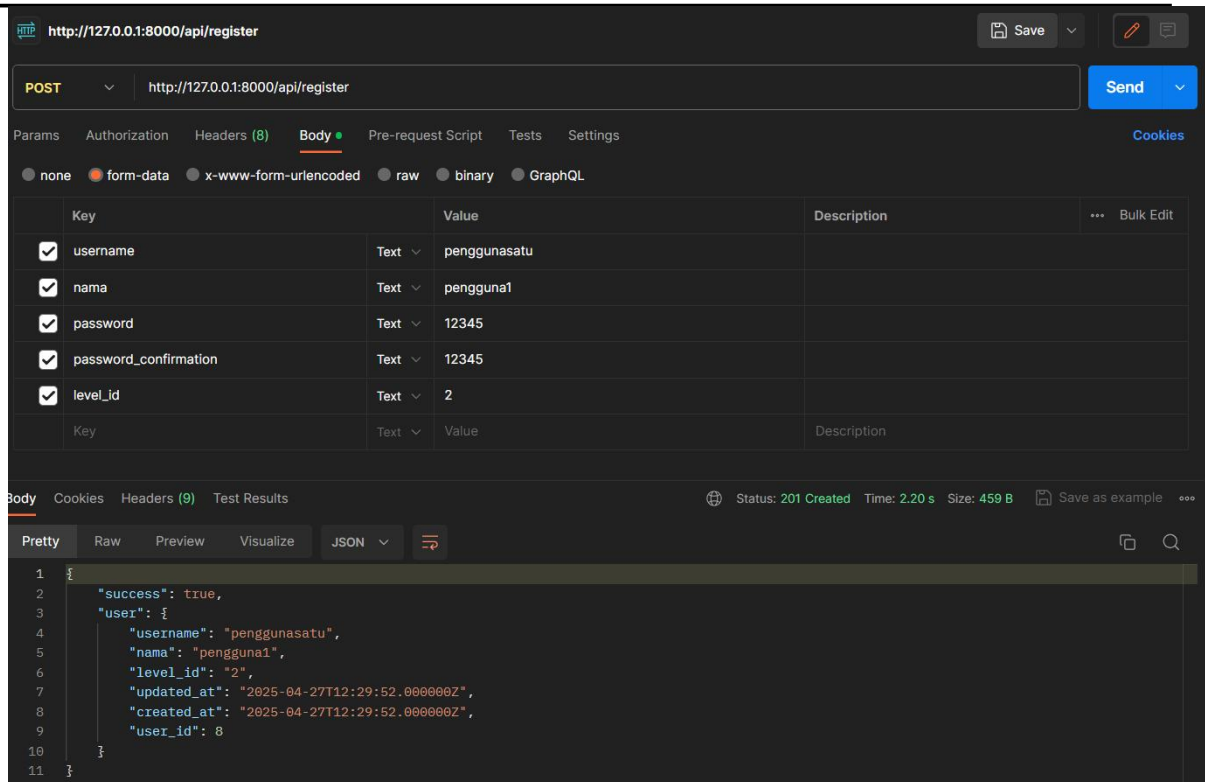
Body Cookies Headers (11) Test Results 201 Created 624 ms 645 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "success": true,
3    "user": {
4      "username": "penggunasatu",
5      "nama": "Pengguna 1",
6      "password": "$2y$12$Eb2SrV1jsyKINytYGtrHi0DVAKcK5p6EgnZnmbChkPicIu7S0QJJJu",
7      "level_id": "2",
8      "updated_at": "2024-04-22T15:56:04.000000Z",
9      "created_at": "2024-04-22T15:56:04.000000Z",
10     "user_id": 17
11   }
12 }
```

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.



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 }
```

```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class LoginController extends Controller
{
    public function __invoke(Request $request)
    {
        //set validation
        $validator = Validator::make($request->all(), [
            'username' => 'required',
            'password' => 'required'
        ]);
    }
}
```





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

//get credentials from request
$credentials = $request->only('username', 'password');

//if auth failed
if (!$token = auth()->guard('api')->attempt($credentials)) {
    return response()->json([
        'success' => false,
        'message' => 'Username atau Password Anda salah'
    ], 401);
}

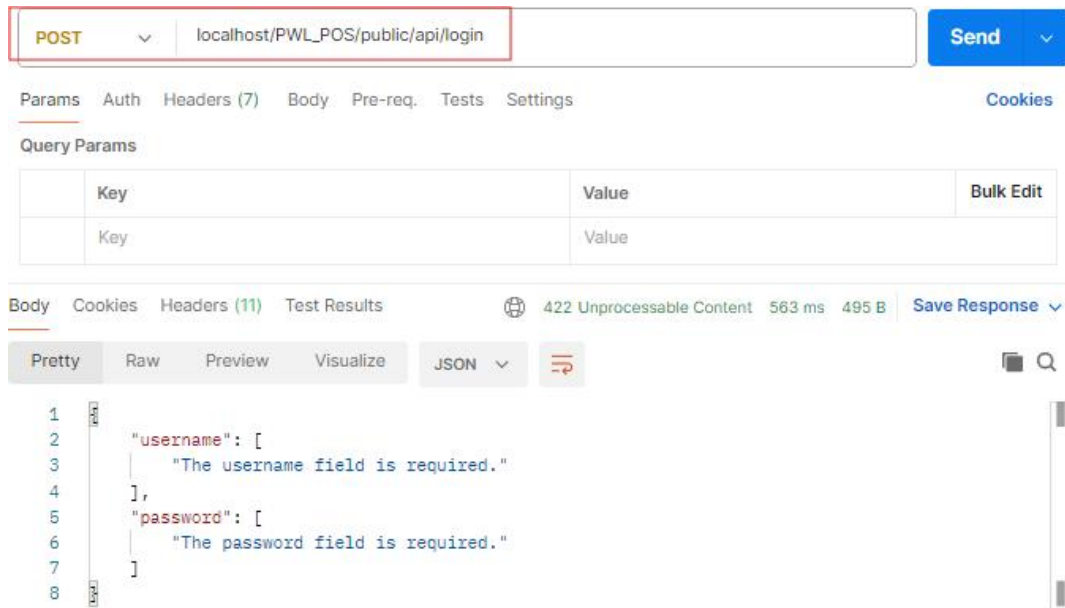
//if auth success
return response()->json([
    'success' => true,
    'user' => auth()->guard('api')->user(),
    'token' => $token
], 200);
}
```

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

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

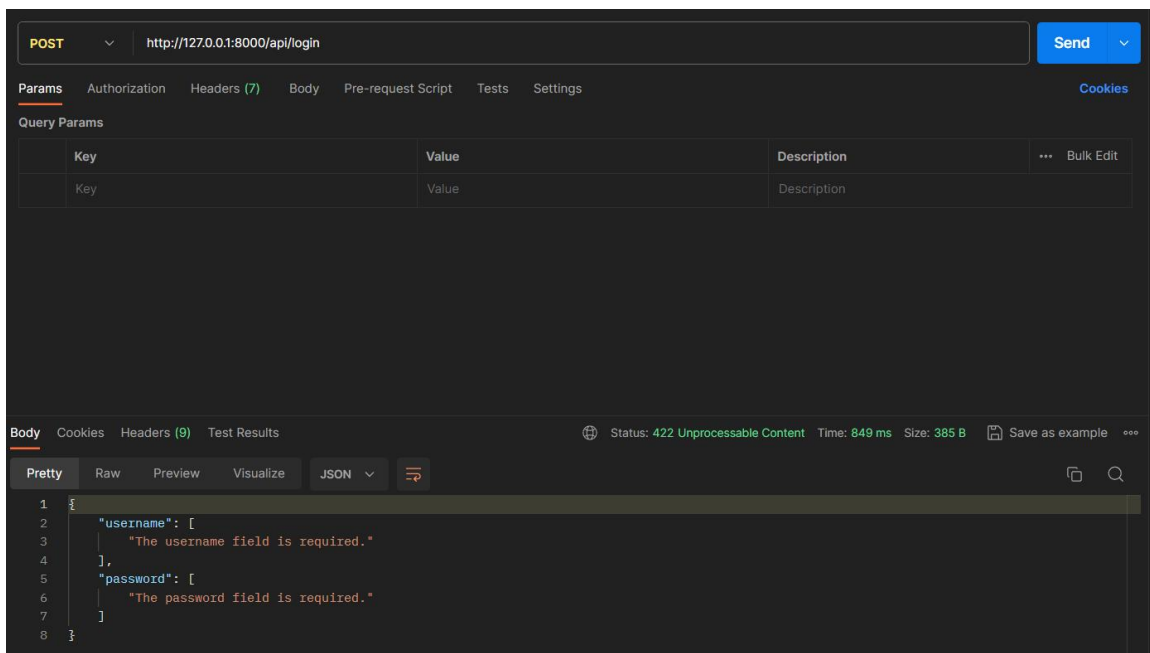
Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
```

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.



Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.



- 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  {"success": true,
2
3  "user": {
4    "user_id": 17,
5    "level_id": 2,
6    "username": "penggunasatu",
7    "nama": "Pengguna 1",
8    "password": "$2y$12$Eb2SxV1jsykINytYGtzH10DVAKcK6p6EgnZnmbChkPicIu7S0QJJU",
9    "created_at": "2024-04-22T15:56:04.000000Z",
10   "updated_at": "2024-04-22T15:56:04.000000Z"
11  },
12  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0L1BXTF9QT1MtbnVwFpb19wdWJsakMvYXBlL2xvZ2luIiwiaWF0Ij0i"
13 }
```



Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.

POST <http://127.0.0.1:8000/api/login> 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
username	penggunasatu	
password	12345	

Body Cookies Headers (9) Test Results Status: 200 OK Time: 2.02 s Size: 800 B Save as example

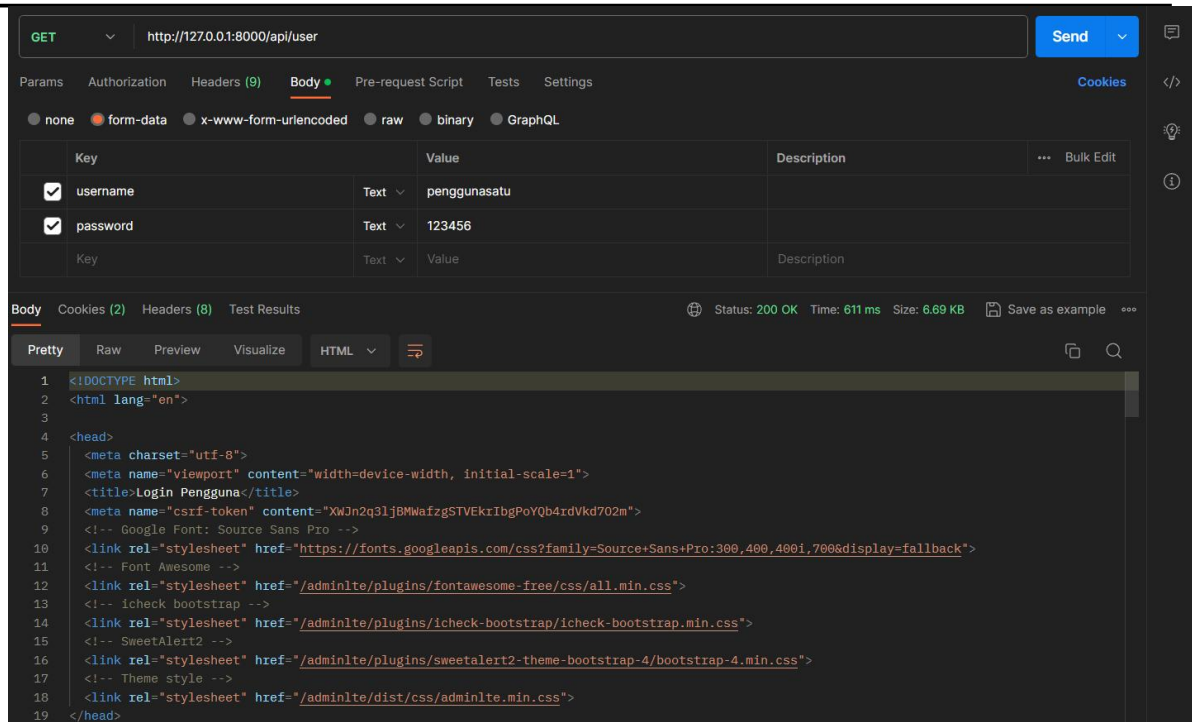
```
1 {
2   "success": true,
3   "user": {
4     "user_id": 8,
5     "level_id": 2,
6     "username": "penggunasatu",
7     "nama": "pengguna1",
8     "created_at": "2025-04-27T12:29:52.000000Z",
9     "updated_at": "2025-04-27T12:29:52.000000Z",
10    "photo_profile": null
11  },
12  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vMTI3LjAuMC4xOjgwMDAvYXBpL2xvZ2luIiwiaWF0IjoxNzU4NjIzLCJleHAiOjE3NDU3NjIyMjMsIm5iZiI6MTc0NTc1ODYyMywianRpIjoic"

```

6. Lakukan percobaan yang untuk data yang salah dan berikan screenshoot hasil percobaan Anda.

```
"success": false,
"message": "Username atau Password Anda salah"
```

7. Coba kembali melakukan login dengan data yang benar. Sekarang mari kita coba menampilkan data user yang sedang login menggunakan URL `localhost/PWL_POS/public/api/user` dan method GET. **Jelaskan hasil dari percobaan tersebut.**



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 }
```



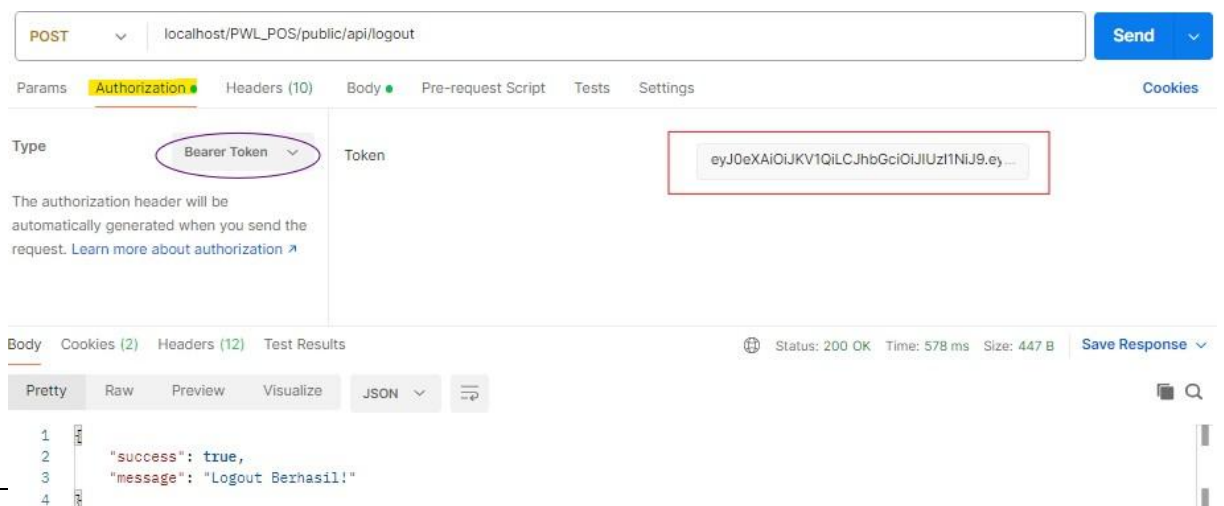


```
o > Http > Controllers > Api > LogoutController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use Illuminate\Http\Request;
6  use App\Http\Controllers\Controller;
7  use Tymon\JWTAuth\Facades\JWTAuth;
8  use Tymon\JWTAuth\Exceptions\JWTException;
9  use Tymon\JWTAuth\Exceptions\TokenExpiredException;
10 use Tymon\JWTAuth\Exceptions\TokenInvalidException;
11
12 class LogoutController extends Controller
13 {
14     public function __invoke(Request $request)
15     {
16         //remove token
17         $removeToken = JWTAuth::invalidate(JWTAuth::getToken());
18
19         if($removeToken) {
20             //return response JSON
21             return response()->json([
22                 'success' => true,
23                 'message' => 'Logout Berhasil!',
24             ]);
25         }
26     }
27 }
28
```

4. Lalu kita tambahkan routes pada api.php

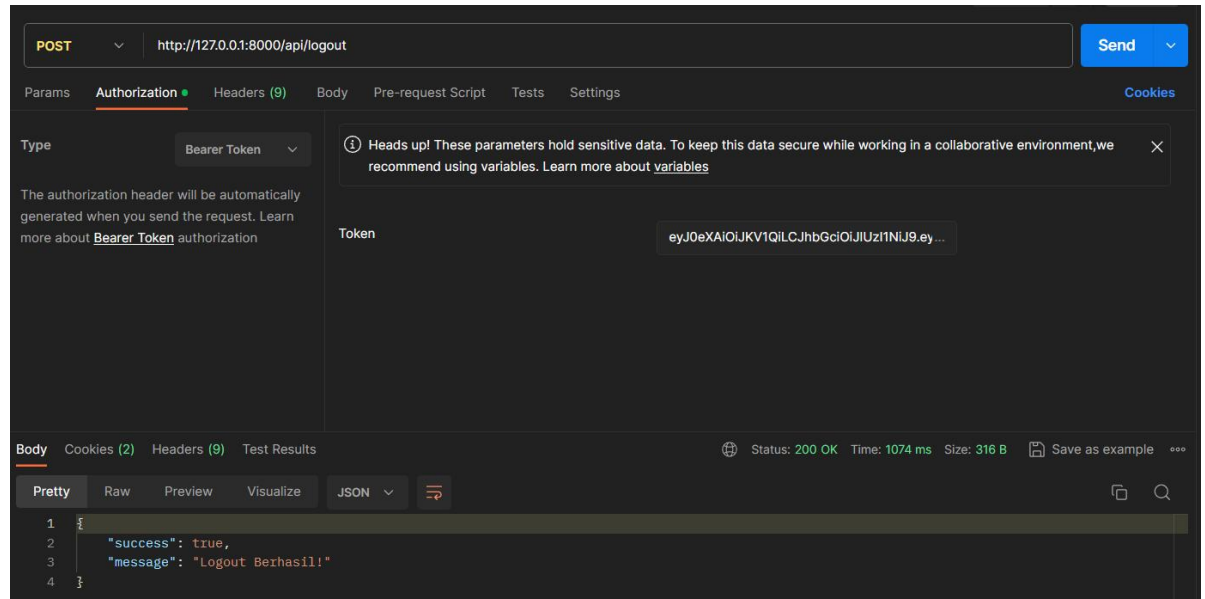
```
Route::post('/logout', App\Http\Controllers\Api\LogoutController::class)->name('logout');
```

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',
    ]);
}
```

```
<?php

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](http://localhost/PWL_POS-main/public/api/levels) dan method GET. **Jelaskan dan berikan screenshot hasil percobaan Anda.**

```
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']);
```





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/levels> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

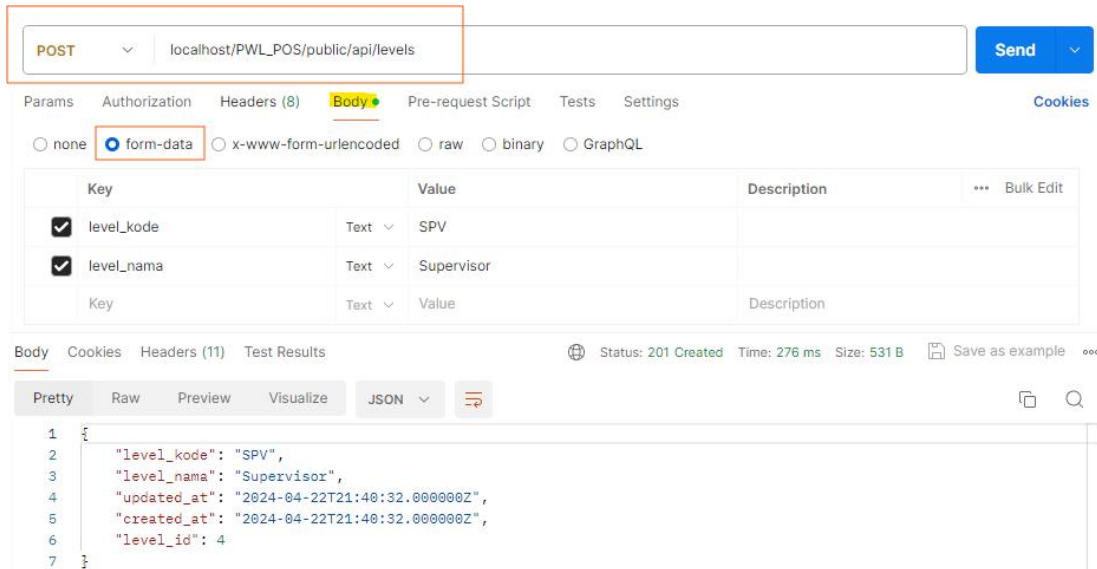
Key	Value	Description
Key	Value	Description

Body Cookies (2) Headers (9) Test Results Status: 200 OK Time: 750 ms Size: 950 B Save as example

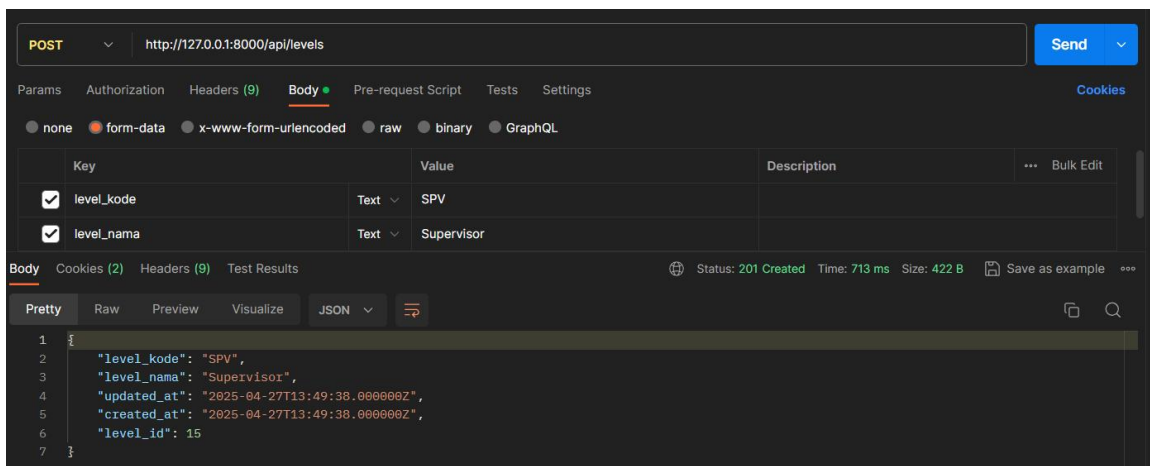
Pretty Raw Preview Visualize JSON

```
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": "Manager",
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 ]
```

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



Jelaskan dan berikan screenshot hasil percobaan Anda.



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





GET <http://127.0.0.1:8000/api/levels/2> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies (2) Headers (9) Test Results Status: 200 OK Time: 706 ms Size: 365 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "level_id": 2,
4     "level_kode": "MMG",
5     "level_nama": "Manager",
6     "created_at": null,
7     "updated_at": null
8   }
9 ]
```

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.

PUT [localhost/PWL\\_POS-main/public/api/levels/4?level\\_kode=SPR](localhost/PWL_POS-main/public/api/levels/4?level_kode=SPR) Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> level_kode	SPR			
Key	Value	Description		

Body Cookies Headers (11) Test Results Status: 200 OK Time: 266 ms Size: 528 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "level_id": 4,
4     "level_kode": "SPR",
5     "level_nama": "Supervisor",
6     "created_at": "2024-04-22T21:40:32.000000Z",
7     "updated_at": "2024-04-22T21:48:19.000000Z"
8   }
9 ]
```

Jelaskan dan berikan screenshoot 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>

PUT

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

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	level_kode	STFF			
	Key	Value	Description		

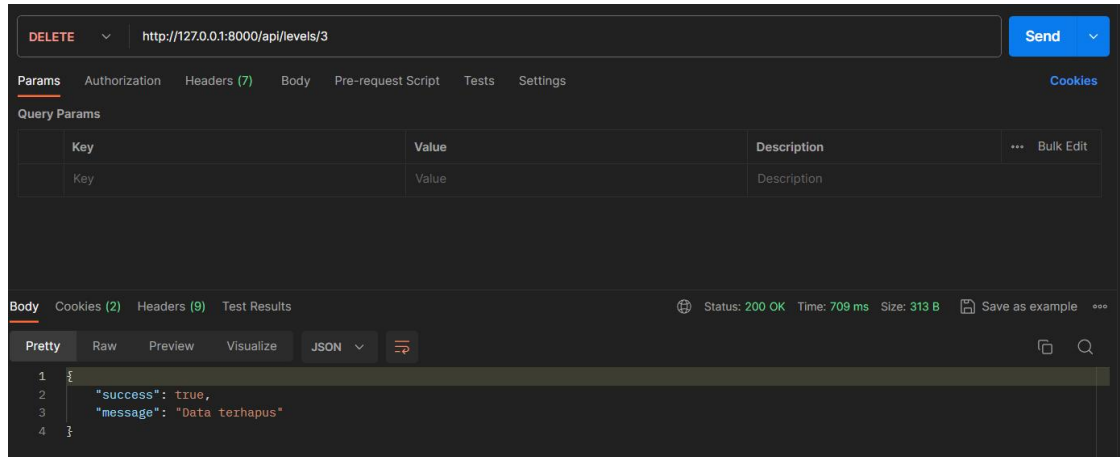
Body Cookies (2) Headers (9) Test Results Status: 200 OK Time: 533 ms Size: 396 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "level_id": 3,
4     "level_kode": "STFF",
5     "level_nama": "Staff/Kasir",
6     "created_at": null,
7     "updated_at": "2025-04-27T13:55:52.000000Z"
8   }
9 ]
```



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`

```
//user
Route::get('users', [UserController::class, 'index']);
Route::post('users', [UserController::class, 'store']);
Route::get('users/{user}', [UserController::class, 'show']);
Route::put('users/{user}', [UserController::class, 'update']);
Route::delete('users/{user}', [UserController::class, 'destroy']);

//kategori
Route::get('kategoris', [KategoriController::class, 'index']);
Route::post('kategoris', [KategoriController::class, 'store']);
Route::get('kategoris/{kategori}', [KategoriController::class, 'show']);
Route::put('kategoris/{kategori}', [KategoriController::class, 'update']);
Route::delete('kategoris/{kategori}', [KategoriController::class, 'destroy']);

//barang
Route::get('barangs', [BarangController::class, 'index']);
Route::post('barangs', [BarangController::class, 'store']);
Route::get('barangs/{barang}', [BarangController::class, 'show']);
Route::put('barangs/{barang}', [BarangController::class, 'update']);
Route::delete('barangs/{barang}', [BarangController::class, 'destroy']);
```



```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\UserModel;
use Illuminate\Http\Request;

class UserController extends Controller
{
    public function index(){
        return UserModel::with('level')->get();
    }

    public function store(Request $request){
        $user = UserModel::create($request->all());
        return response()->json($user->load('level'), 201);
    }

    public function show(UserModel $user){
        return UserModel::find($user);
    }

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

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

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



```
<?php

namespace App\Http\Controllers\Api;

use App\Http\Controllers\Controller;
use App\Models\KategoriModel;
use Illuminate\Http\Request;

class KategoriController extends Controller
{
    public function index(){
        return KategoriModel::all();
    }

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

    public function show(KategoriModel $kategori){
        return KategoriModel::find($kategori);
    }

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

    public function destroy(KategoriModel $kategori){
        $kategori->delete();

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





```
> Http > Controllers > Api > BarangController.php > ...
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
9  class BarangController extends Controller
10 {
11     public function index(){
12         return BarangModel::with('kategori')->get();
13     }
14
15     public function store(Request $request){
16         $barang = BarangModel::create($request->all());
17         return response()->json($barang->load('kategori'), 201);
18     }
19
20     public function show(BarangModel $barang){
21         return $barang->load('kategori');
22     }
23
24     public function update(Request $request, BarangModel $barang){
25         $barang->update($request->all());
26         return response()->json($barang->load('kategori'), 200);
27     }
28     public function destroy(BarangModel $barang){
29         $barang->delete();
30
31         return response()->json([
32             'success' => true,
33             'message' => 'Data terhapus',
34         ]);
35     }
36 }
37
```





## POSTMAN

### User

HTTP USER / SHOW Save Send

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

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 38.37 s Size: 4.64 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "user_id": 1,
4     "level_id": 1,
5     "username": "admin",
6     "nama": "Administrator",
7     "created_at": null,
8     "updated_at": "2025-04-08T14:26:46.000000Z",
9     "photo_profile": "profiles/profile_1_1744122406.png",
10    "level": {
11      "level_id": 1,
12      "level_kode": "ADM",
13      "level_nama": "Administrator",
14      "created_at": null,
15      "updated_at": null
16    }
17  },
18  {
19    "user_id": 2,
20    "level_id": 2,
21    "username": "manager",
22    "nama": "Manager",
23    "created_at": null,
```

GET http://127.0.0.1:8000/api/users/1 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 733 ms Size: 455 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "user_id": 1,
4     "level_id": 1,
5     "username": "admin",
6     "nama": "Administrator",
7     "created_at": null,
8     "updated_at": "2025-04-08T14:26:46.000000Z",
9     "photo_profile": "profiles/profile_1_1744122406.png"
10  }
11 ]
```



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** <http://127.0.0.1:8000/api/users> **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_id	Text	2		
<input checked="" type="checkbox"/>	username	Text	userviaapi		
<input checked="" type="checkbox"/>	nama	Text	userviaapi		
<input checked="" type="checkbox"/>	password	Text	12345		

Body Cookies Headers (9) Test Results Status: 201 Created Time: 1368 ms Size: 536 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "level_id": "2",
3   "username": "userviaapi",
4   "nama": "userviaapi",
5   "updated_at": "2025-04-28T00:13:02.000000Z",
6   "created_at": "2025-04-28T00:13:02.000000Z",
7   "user_id": 29,
8   "level": {
9     "level_id": 2,
10    "level_kode": "MNG",
11    "level_nama": "Manager",
12    "created_at": null,
13    "updated_at": null
14  }
15 }
```

**PUT** [http://127.0.0.1:8000/api/users/29?level\\_id=2&username=ubahuserapi&nama=ubahuserapi&password=ubahuserapi](http://127.0.0.1:8000/api/users/29?level_id=2&username=ubahuserapi&nama=ubahuserapi&password=ubahuserapi) **Send**

Params Authorization Headers (7) **Body** Pre-request Script Tests Settings Cookies

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

Body Cookies Headers (9) Test Results Status: 200 OK Time: 683 ms Size: 600 B Save as example

Pretty Raw Preview Visualize JSON

```
2 {
3   "user_id": 2,
4   "level_id": 2,
5   "username": "manager",
6   "nama": "Manager",
7   "created_at": null,
8   "updated_at": "2025-04-08T14:48:27.000000Z",
9   "photo_profile": null
10 },
11 {
12   "user_id": 29,
13   "level_id": 2,
14   "username": "ubahuserapi",
15   "nama": "ubahuserapi",
16   "created_at": "2025-04-28T00:13:02.000000Z",
17   "updated_at": "2025-04-28T00:18:32.000000Z",
18   "photo_profile": null
19 }
20 ]
```



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>

DELETE  <http://127.0.0.1:8000/api/users/29>

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

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

## Kategori

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

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"/>	kategori_kode	Text <input type="button" value="Send"/> KTGAPI			
<input checked="" type="checkbox"/>	kategori_nama	Text <input type="button" value="Send"/> Kategori API			

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "kategori_kode": "KTGAPI",
3   "kategori_nama": "Kategori API",
4   "updated_at": "2025-04-29T03:08:29.000000Z",
5   "created_at": "2025-04-29T03:08:29.000000Z",
6   "kategori_id": 26
7 }
```

KATEGORI / UPDATE

PUT  [http://127.0.0.1:8000/api/kategoris/26?kategori\\_kode=KTGAPI2&kategori\\_nama=Kategori Api Kedua](http://127.0.0.1:8000/api/kategoris/26?kategori_kode=KTGAPI2&kategori_nama=Kategori Api Kedua)

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	kategori_kode	KTGAPI2			
<input checked="" type="checkbox"/>	kategori_nama	Kategori Api Kedua			

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "kategori_id": 26,
4     "kategori_kode": "KTGAPI2",
5     "kategori_nama": "Kategori Api Kedua",
6     "created_at": "2025-04-29T03:08:29.000000Z",
7     "updated_at": "2025-04-29T03:11:16.000000Z"
8   }
9 ]
```



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 KATEGORI / DELETE

DELETE http://127.0.0.1:8000/api/kategoris/26

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 1228 ms Size: 313 B Save as example

Pretty Raw Preview Visualize JSON

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

HTTP KATEGORI / SHOW

GET http://127.0.0.1:8000/api/kategoris

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 509 ms Size: 1.46 KB Save as example

Pretty Raw Preview Visualize JSON

```
39 {
40   "kategori_kode": "CML",
41   "kategori_nama": "Cemilan",
42   "created_at": "2025-03-16T12:38:31.000000Z",
43   "updated_at": null
44 },
45 {
46   "kategori_id": 8,
47   "kategori_kode": "MNR",
48   "kategori_nama": "Minuman Ringan",
49   "created_at": "2025-03-16T12:37:03.000000Z",
50   "updated_at": null
51 },
52 {
53   "kategori_id": 22,
54   "kategori_kode": "GNC",
55   "kategori_nama": "Gantungan Kunci",
56   "created_at": "2025-04-05T05:21:17.000000Z",
57   "updated_at": null
58 },
59 {
60   "kategori_id": 23,
61   "kategori_kode": "GNB",
62   "kategori_nama": "Gantungan Budi",
63   "created_at": "2025-04-05T05:21:17.000000Z",
64   "updated_at": null
65 }
```

Barang



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>

BARANG / SHOW

Save

Send

GET

http://127.0.0.1:8000/api/barangs

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Send and Download

Query Params

Key	Value	Description
Key	Value	Description

Body

Cookies

Headers (9)

Test Results

Status: 200 OK

Time: 720 ms

Size: 3.19 KB

Save as example

Pretty

Raw

Preview

Visualize

JSON

Send

```
1 [
2   {
3     "barang_id": 1,
4     "kategori_id": 1,
5     "barang_kode": "BRG001",
6     "barang_nama": "TV",
7     "harga_beli": 3000000,
8     "harga_jual": 3500000,
9     "created_at": null,
10    "updated_at": null,
11    "kategori": {
12      "kategori_id": 1,
13      "kategori_kode": "KTG001",
14      "kategori_nama": "Elektronik",
15      "created_at": null,
16      "updated_at": null
17    }
18  },
19  ]
```



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 BARANG / CREATE Save Edit

POST http://127.0.0.1:8000/api/barangs 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"/>	kategori_id	Text 1			
<input checked="" type="checkbox"/>	barang_kode	Text TSA			
<input checked="" type="checkbox"/>	barang_nama	Text Test API			
<input checked="" type="checkbox"/>	harga_beli	Text 10000			
<input checked="" type="checkbox"/>	harga_jual	Text 12000			
	Key	Text Value	Description		

Body Cookies Headers (9) Test Results Status: 201 Created Time: 797 ms Size: 602 B Save as example ...

Pretty Raw Preview Visualize JSON ...

```
1 {
2   "kategori_id": "1",
3   "barang_kode": "TSA",
4   "barang_nama": "Test API",
5   "harga_beli": "10000",
6   "harga_jual": "12000",
7   "updated_at": "2025-04-29T03:16:52.000000Z",
8   "created_at": "2025-04-29T03:16:52.000000Z",
9   "barang_id": 20,
10  "kategori": {
11    "kategori_id": 1,
12    "kategori_kode": "KTG001",
13    "kategori_nama": "Elektronik",
14    "created_at": null
```





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 BARANG / UPDATE

PUT [http://127.0.0.1:8000/api/barangs/20?kategori\\_id=1&barang\\_kode=TSA2&barang\\_nama=Test API 2&harga\\_beli=12000&harga\\_jual=12500](http://127.0.0.1:8000/api/barangs/20?kategori_id=1&barang_kode=TSA2&barang_nama=Test API 2&harga_beli=12000&harga_jual=12500) Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	kategori_id	1			
<input checked="" type="checkbox"/>	barang_kode	TSA2			
<input checked="" type="checkbox"/>	barang_nama	Test API 2			
<input checked="" type="checkbox"/>	harga_beli	12000			
<input checked="" type="checkbox"/>	harga_jual	12500			
	Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 7.22 s Size: 600 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "barang_id": 20,
3   "kategori_id": "1",
4   "barang_kode": "TSA2",
5   "barang_nama": "Test API 2",
6   "harga_beli": "12000",
7   "harga_jual": "12500",
8   "created_at": "2025-04-29T03:16:52.000000Z",
9   "updated_at": "2025-04-29T03:19:16.000000Z",
10  "kategori": {
11    "kategori_id": 1,
12    "kategori_kode": "KTG001",
13    "kategori_nama": "Elektronik",
14    "created_at": null
15  }
```

HTTP BARANG / DELETE

DELETE <http://127.0.0.1:8000/api/barangs/20> Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
--	-----	-------	-------------	-----	-----------

Body Cookies Headers (9) Test Results Status: 200 OK Time: 723 ms Size: 313 B Save as example

Pretty Raw Preview Visualize JSON

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

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