

Tugas Pertemuan 10 Pemrograman Web Lanjut



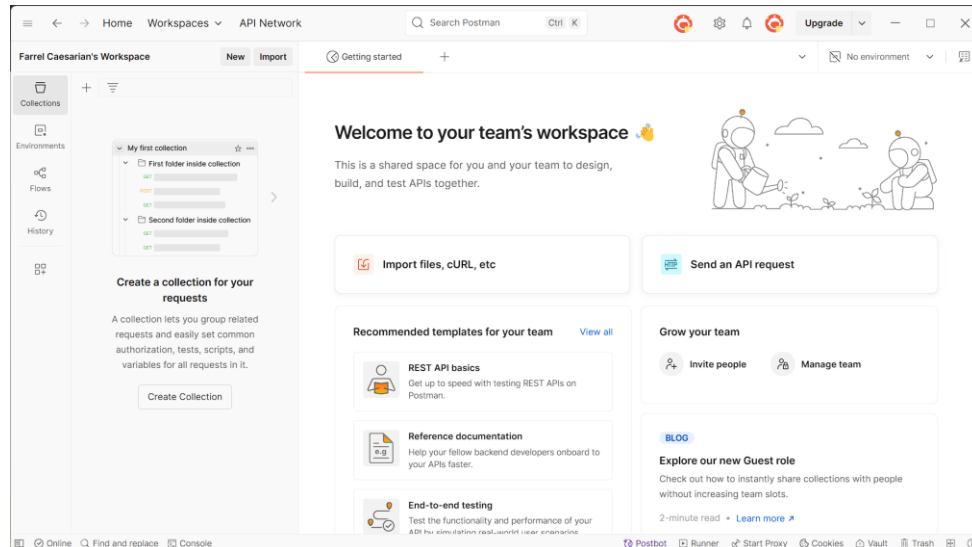
Muhammad Farrel Caesarian
TI-2D/16
2341720053
POLITEKNIK NEGERI MALANG

1. **Praktikum 1 – Membuat RESTful API Register**

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.



Lakukan instalasi JWT dengan mengetikkan perintah berikut:

`composer require tymon/jwt-auth:2.1.1`

Pastikan Anda terkoneksi dengan internet

```
PS C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10> composer require tymon/jwt-auth:2.1.1
./composer.json has been updated
Running composer update tymon/jwt-auth
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
- Locking lcobucci/clock (2.2.0)
- Locking lcobucci/jwt (4.0.4)
- Locking stella-maris/clock (0.1.7)
```

Setelah berhasil menginstall JWT, lanjutkan dengan publish konfigurasi file dengan

perintah berikut:

`php artisan vendor:publish --`

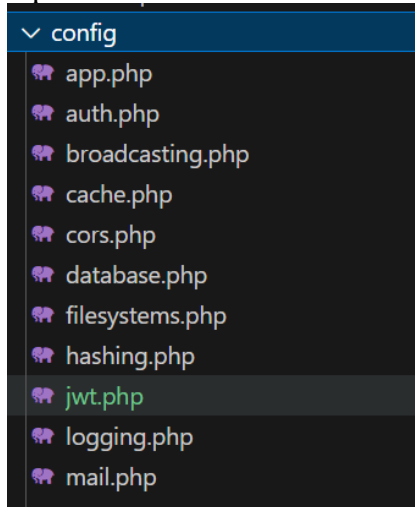
`provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"`

```
PS C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10> php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"

INFO Publishing assets.

Copying file [C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10\vendor\tymon\jwt-auth\config\config.php] to [C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10\config\jwt.php] DONE
```

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.



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

```
PS C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10> php artisan jwt:secret
jwt-auth secret [zpAaB4SojyouIwY2YFUZ35C2bZMtn8721U6cWpQET4gmSvfzLI4rM8MJGOFBTKC6] set successfully.
60
61 JWT_SECRET= zpAaB4SojyouIwY2YFUZ35C2bZMtn8721U6cWpQET4gmSvfzLI4rM8MJGOFBTKC6
62
```

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',
    ],
],
```

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

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

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

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

`php artisan make:controller Api/RegisterController`

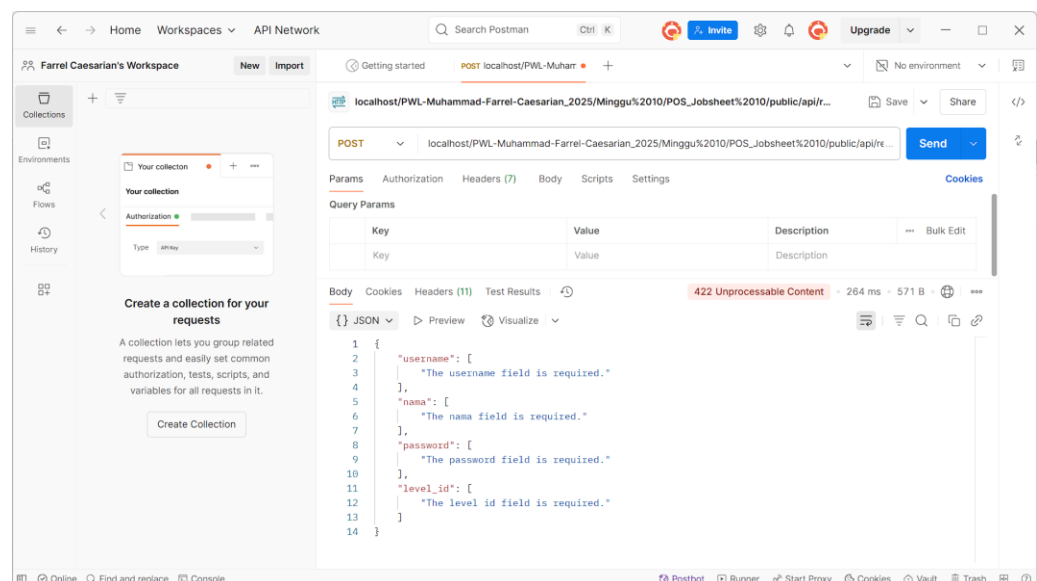
Jika berhasil maka akan ada tambahan controller pada folder Api dengan nama RegisterController

```
app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController >
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\UserModel;
8  use Illuminate\Support\Facades\Validator;
9  class RegisterController extends Controller
10 {
11     public function __invoke(request $request){
12
13         $validator = Validator::make($request->all(),[
14             'username' => 'required|string|max:255|unique:m_user',
15             'nama' => 'required|string|max:255',
16             'password' => 'required|string|min:6|confirmed',
17             'level_id' => 'required'
18         ]);
19
20         if($validator->fails()){
21             return response()->json($validator->errors(), 422);
```

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

```
routes > api.php
1  <?php
2
3  use Illuminate\Http\Request;
4  use Illuminate\Support\Facades\Route;
5  use App\Http\Controllers\Api\RegisterController;
6  /*
7  |-----
8  | API Routes
9  |-----
10 |
11 | Here is where you can register API routes for your application. These
12 | routes are loaded by the RouteServiceProvider and all of them will
13 | be assigned to the "api" middleware group. Make something great!
14 |
15 | */
16
17 Route::post('/register', RegisterController::class)->name('register');
```

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



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.

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

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> username	Text ▾	penggunasatu	
<input checked="" type="checkbox"/> nama	Text ▾	Pengguna 1	
<input checked="" type="checkbox"/> password	Text ▾	123456	
<input checked="" type="checkbox"/> password_confirmation	Text ▾	123456	
<input checked="" type="checkbox"/> level_id	Text ▾	1	
Key	Text ▾	Value	Description

```
1 {
2   "success": "true",
3   "user": {
4     "username": "penggunasatu",
5     "nama": "Pengguna 1",
6     "level_id": "1",
7     "updated_at": "2025-04-23T07:02:10.000000Z",
8     "created_at": "2025-04-23T07:02:10.000000Z",
9     "user_id": 40
10  }
11 }
```

40	1	penggunasatu	Pengguna 1	NULL
----	---	--------------	------------	------

Data terekam di database

Lakukan commit perubahan file pada Github.

Implementasi API Praktikum 1



FarrelCaesarian • 11 minutes ago

2. **Praktikum 2** – Membuat RESTful API Login

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

```
php artisan make:controll
er Api/LoginControllerMuhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10>

INFO Controller [C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10\app\Http
\Controllers\Api>LoginController.php] created successfully.
```

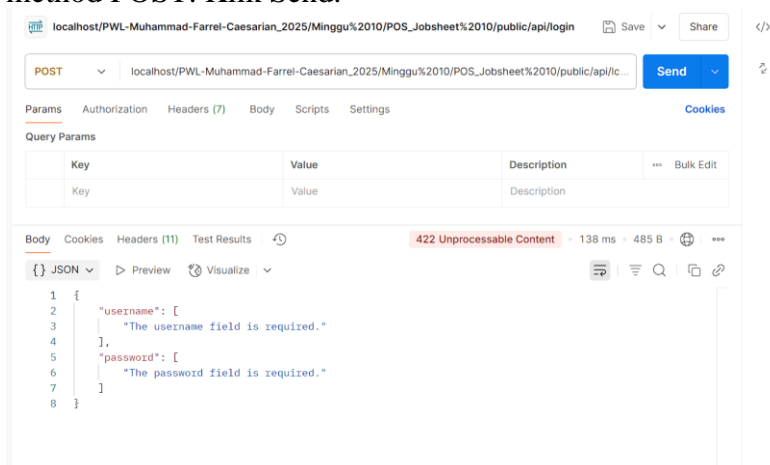
Buka file tersebut, dan ubah kode menjadi seperti berikut.

```
app > Http > Controllers > Api > LoginController.php > PHP Intelephense > LoginController > _in
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         $validator = Validator::make($request->all(),[
13             'username' => 'required|string|max:255',
14             'password' => 'required|string|min:6'
15         ]);
16
17         if($validator->fails()){
18             return response()->json($validator->errors(), 422);
19         }
20
21         $credentials = $request->only('username', 'password');
```

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

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

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.



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.

```
use Tymon\JWTAuth\Contracts\JWTSubject;

class UserModel extends Authenticatable implements JWTSubject
```

Tambahkan kode ini pada UserModel

localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/login Save Share

POST localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/c... Send

Params Authorization Headers (8) Body Scripts Settings Cookies

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

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> username	Text penggunasatu			
<input checked="" type="checkbox"/> password	Text 123456			
Key	Text Value	Description		

Body Cookies Headers (11) Test Results 200 OK • 662 ms • 997 B

{ } JSON Preview Visualize

```
1 {
2   "success": "true",
3   "user": {
4     "user_id": 40,
5     "level_id": 1,
6     "username": "penggunasatu",
7     "nama": "Pengguna 1",
8     "profile_picture": null,
9     "created_at": "2025-04-23T07:02:10.000000Z",
10    "updated_at": "2025-04-23T07:02:10.000000Z"
11  },
12  "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
```

Data Salah

localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/login Save Share </>

POST localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/c... Send

Params Authorization Headers (8) Body Scripts Settings Cookies

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

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> username	Text penggunasatu			
<input checked="" type="checkbox"/> password	Text 1234567			
Key	Text Value	Description		

Body Cookies Headers (11) Test Results 401 Unauthorized • 439 ms • 446 B

{ } JSON Preview Visualize

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

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

localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/user

Save

Share

</>

GET

localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/u...

Send

>

ParamsAuthorizationHeaders (7)BodyScriptsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

BodyCookies (2)Headers (10)Test Results

200 OK • 270 ms • 8.23 KB •

HTMLPreviewVisualize

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Login Pengguna</title>
8   <link rel="stylesheet"
9     href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700&
    display=fallback">
10  <link rel="stylesheet"
11    href="http://localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/
    adminlte/plugins/fontawesome-free/css/all.min.css">
12  <link rel="stylesheet"
13    href="http://localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/
    adminlte/plugins/icheck-bootstrap/icheck-bootstrap.min.css">
```

Lakukan commit perubahan file pada Github

Implementasi API Login Praktik...

FarrelCaesarian • just now

↑

3. **Praktikum 3 – Membuat RESTful API Logout**
Tambahkan kode berikut pada file .env
`JWT_SHOW_BLACKLIST_EXCEPTION=true`
`JWT_SECRET=zpAaB4SojyouIwY2YFUZ35C2bZM`
`JWT_SHOW_BLACKLIST_EXCEPTION=true`

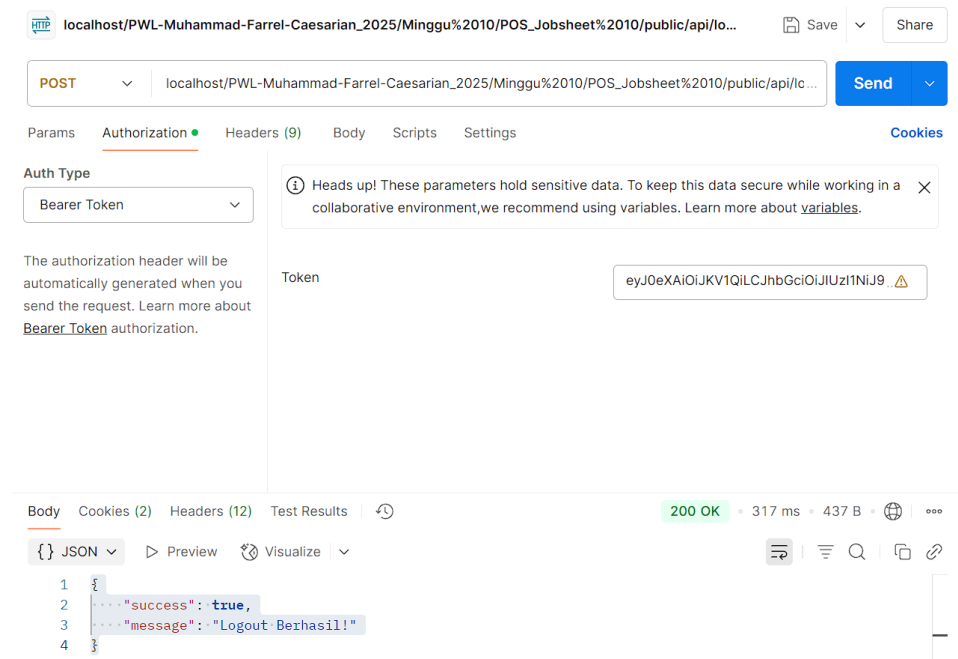
Buat Controller baru dengan nama LogoutController.
`php artisan make:controller Api/LogoutController`

```
PS C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10> php artisan make:controll
er Api/LogoutController

INFO Controller [C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10\app\Http
\Controllers\Api\LogoutController.php] created successfully.
```


Lalu kita tambahkan routes pada api.php
`Route::post('/logout', LogoutController::class)->name('logout');`

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.



Lakukan commit perubahan file pada Github

Implementasi Logout API Prakti...



FarrelCaesarian • just now



4. **Praktikum 4 – Implementasi CRUD dalam RESTful API**
Pertama, buat controller untuk mengolah API pada data level.

[php artisan make:controller Api/LevelController](#)

```
PS C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10> php artisan make:controller Api/LevelController

INFO Controller [C:\laragon\www\PWL-Muhammad-Farrel-Caesarian_2025\Minggu 10\POS_Jobsheet 10\app\Http\Controllers\Api\LevelController.php] created successfully.
```

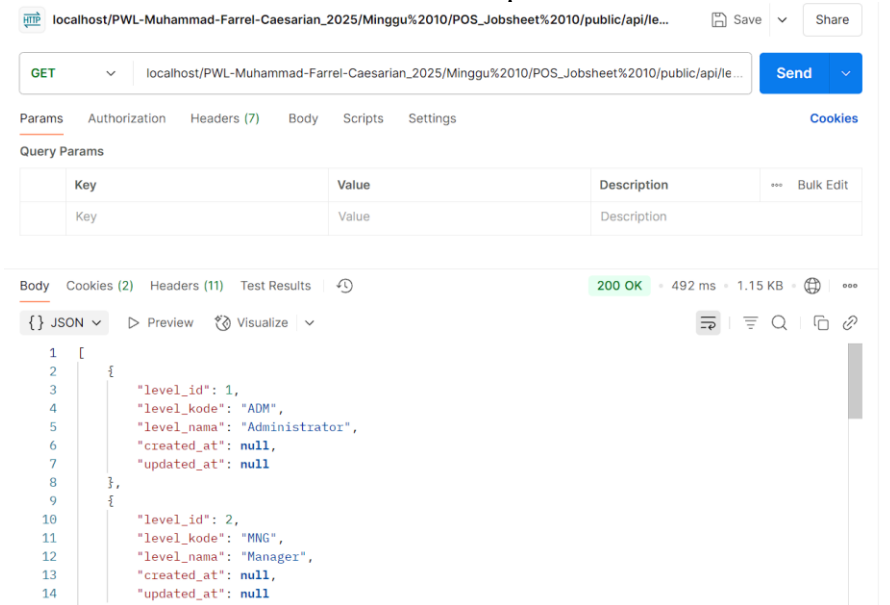
Setelah berhasil, buka file tersebut dan tuliskan kode seperti berikut yang berisi fungsi CRUDnya

```
app > Http > Controllers > Api > LevelController.php > PHP Intelephense > LevelController
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\LevelModel;
8
9 class LevelController extends Controller
10 {
11     public function index(){
12         return LevelModel::all();
13     }
14
15     public function store(request $request){
16         $level = LevelModel::create($request->all());
17         return response()->json($level,201);
18     }
19
20     public function show(LevelModel $level)
```

Kemudian kita lengkapi routes pada api.php

```
//level
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']);
```

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 screenshoot hasil percobaan Anda.



localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/le...

GET localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/le... Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

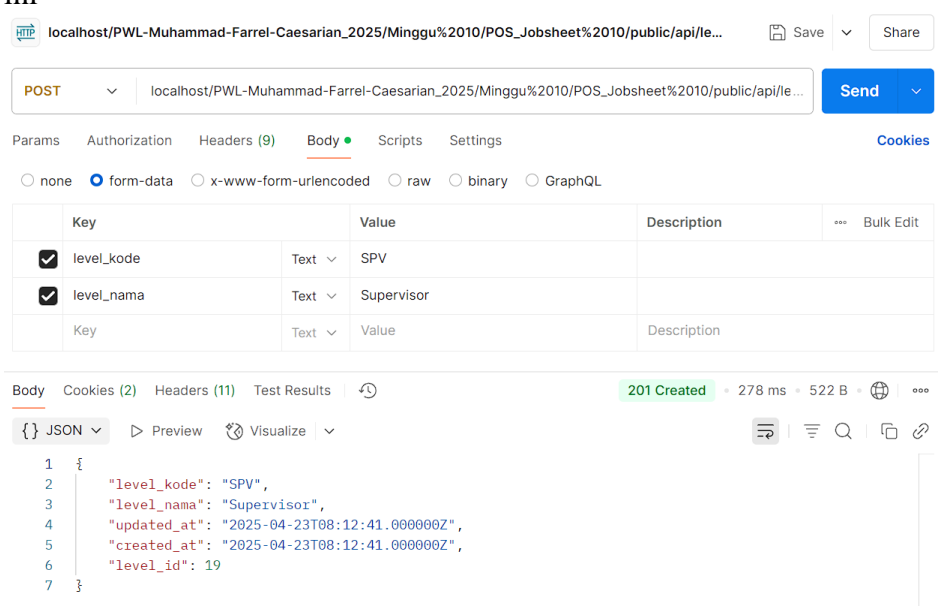
Body Cookies (2) Headers (11) Test Results 200 OK • 492 ms • 1.15 KB

JSON Preview Visualize

```
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 ]
```

Menampilkan daftar level pada system pos

Kemudian, lakukan percobaan penambahan data dengan URL : localhost/PWL_POSmain/public/api/levels dan method POST seperti di bawah ini



localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/le... Save Share

POST localhost/PWL-Muhammad-Farrel-Caesarian_2025/Minggu%2010/POS_Jobsheet%2010/public/api/le... 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	

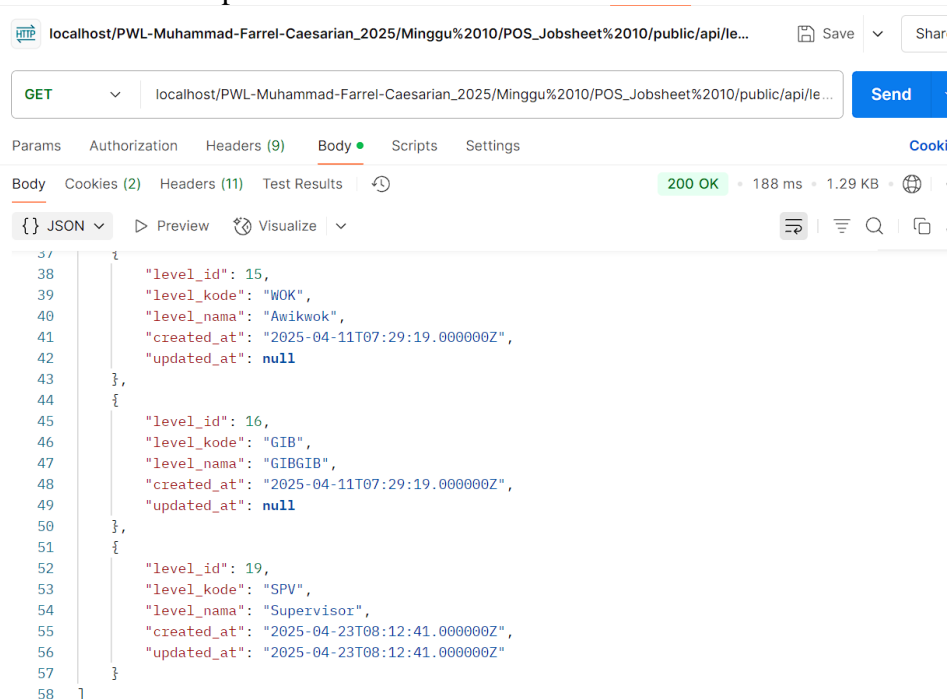
Body Cookies (2) Headers (11) Test Results 201 Created • 278 ms • 522 B

JSON Preview Visualize

```
1 {
2   "level_kode": "SPV",
3   "level_nama": "Supervisor",
4   "updated_at": "2025-04-23T08:12:41.000000Z",
5   "created_at": "2025-04-23T08:12:41.000000Z",
6   "level_id": 19
7 }
```

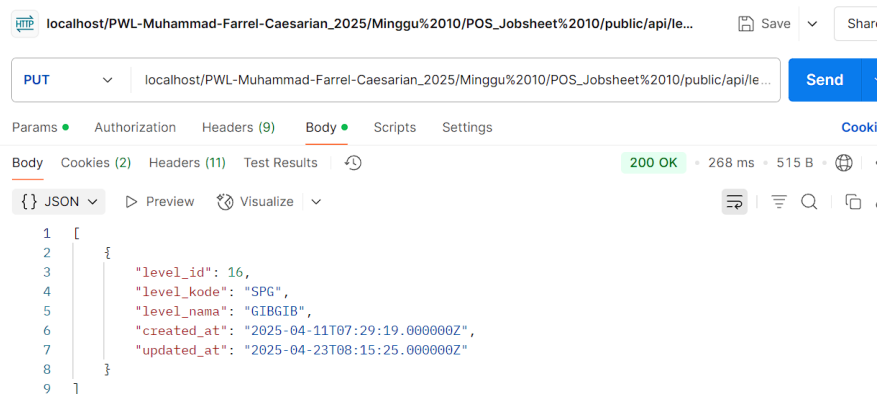
Kita dapat membuat level baru melalui API

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



Menggunakan method GET, dan detail data dari level akan ditampilkan.

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

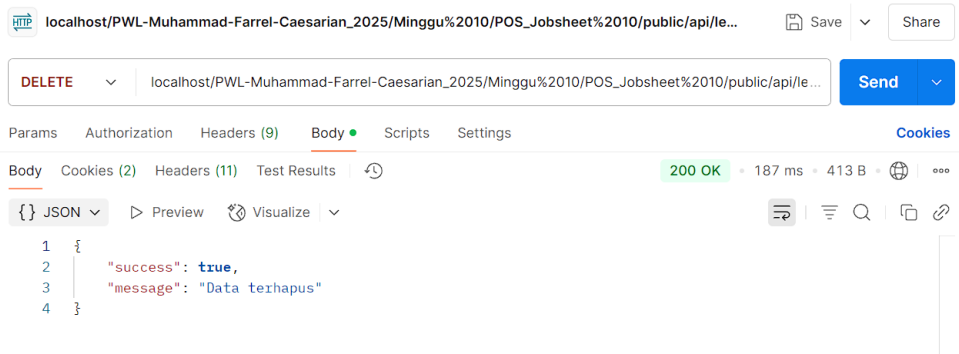


Terakhir lakukan percobaan hapus data. Jelaskan dan berikan screenshot hasil percobaan Anda

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

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

Kode destroy yang benar



Saya menghapus level_id 16 (SPG)

level_id	level_kode	level_nama	created_at	updated_at
1	ADM	Administrator	NULL	NULL
2	MNG	Manager	NULL	NULL
3	STF	Staff	NULL	2025-03-30 09:18:58
6	CUS	Pelanggan	2025-03-06 13:35:44	NULL
14	KSR	Kasir	2025-03-30 09:18:45	2025-03-30 09:18:45
15	WOK	Awikwok	2025-04-11 07:29:19	NULL
19	SPV	Supervisor	2025-04-23 08:12:41	2025-04-23 08:12:41

Data terhapus dari database.

Commit GitHub

Implementasi CRUD API Praktikum 4

 FarrelCaesarian • just now

5. 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/{level}', [UserController::class, 'show']);
Route::put('users/{level}', [UserController::class, 'update']);
Route::delete('users/{level}', [UserController::class, 'destroy']);

app > Http > Controllers > Api > UserController.php > PHP Intelephense > UserController
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\UserModel;
8
9  class UserController extends Controller
10 {
11     public function index(){
12         return UserModel::all();
13     }
14
15     public function store(request $request){
16         $user = UserModel::create($request->all());
17         return response()->json($user,201);
18     }
19 }
```

6.	
----	--