

Laporan Pemrograman Web Lanjut

Jobsheet 10 : RESTFUL API

Dosen Pengampu : Dimas Wahyu Wibowo, ST.,MT.



Disusun Oleh:

Queenadhynar Azarine Dwipa A.

2341760109

SIB 2B

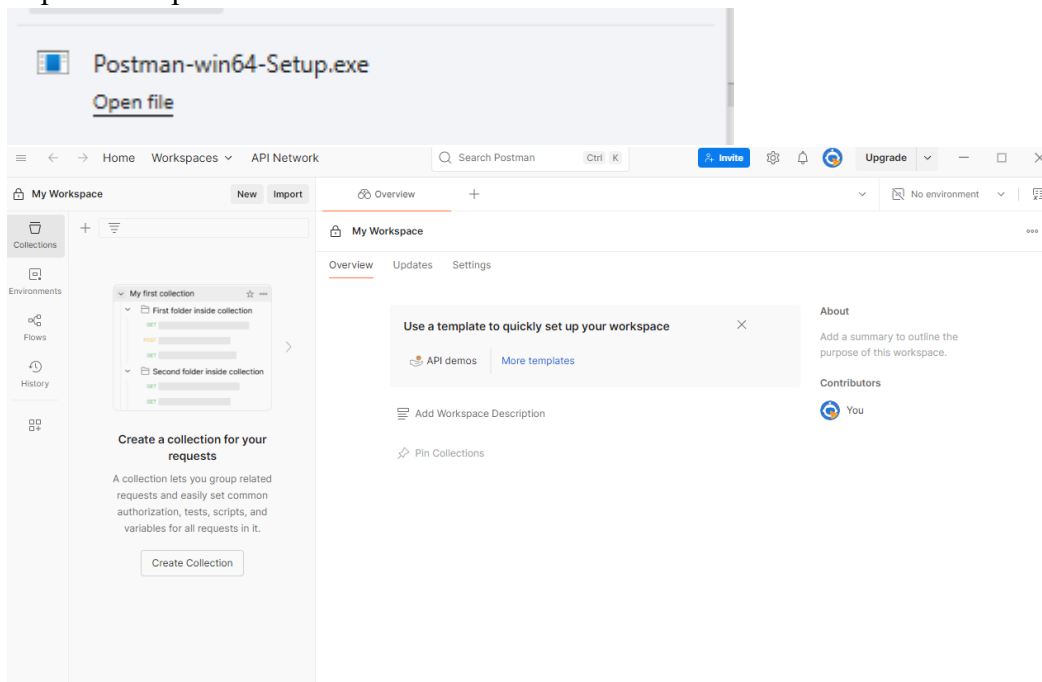
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023/2024

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.

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\pwl_pos> 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 (2.3.0)
- Locking lcobucci/jwt (4.0.4)
- Locking stella-maris/clock (0.1.7)
- Locking tymon/jwt-auth (2.1.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
```

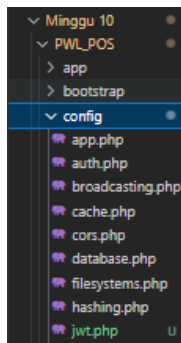
3. Setelah berhasil menginstall JWT, lanjutkan dengan publish konfigurasi file dengan perintah berikut:

`php artisan vendor:publish -`

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

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\pwl_pos> php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"
INFO: Publishing assets.
Copying file [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\pwl_pos\vendor\tymon\jwt-auth\config\config.php] to [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\pwl_pos\config\jwt.php] DONE
```

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.



- 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 D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\pwl_pos> php artisan jwt:secret
jwt-auth secret [3pAXwUtzY6Ux5AL8wW3SHILNV53QR780eXZVTlnpFat99KYiLAZvvHnZpgQva0nv] set successfully.
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\pwl_pos>

'secret' => env('JWT_SECRET'),
```

- 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:

```
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Foundation\Auth\User as Authenticatable; // implementasi

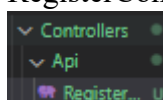
class UserModel extends Authenticatable implements JWTSubject
{
    public function getJWTIdentifier()
    {
        return $this->getKey();
    }

    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.



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

```
RegisterController.php U X
Minggu 10 > PWL_POS > app > Http > Controllers > Api > RegisterController.php > RegisterCo

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 validation 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 }
```

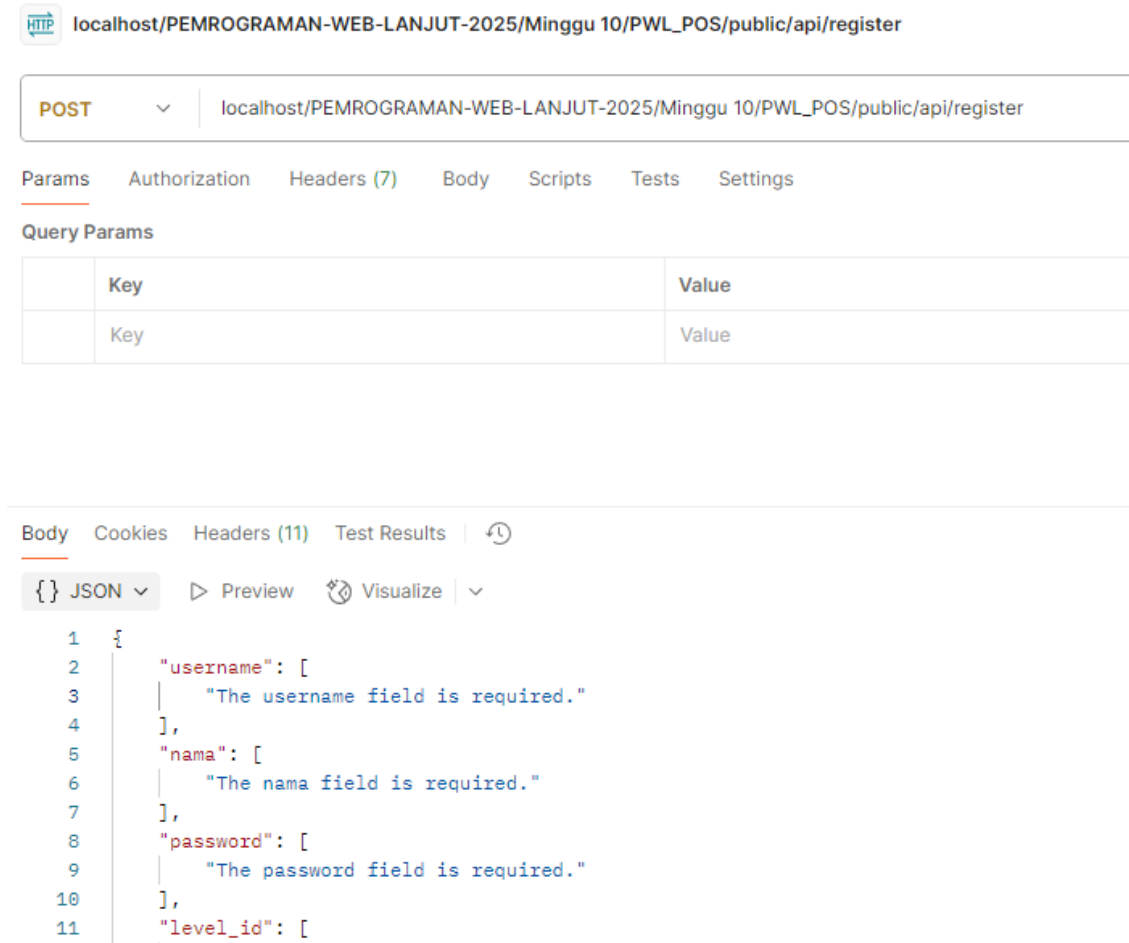
10. Selanjutnya buka routes/api.php, ubah semua kode menjadi seperti berikut.

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

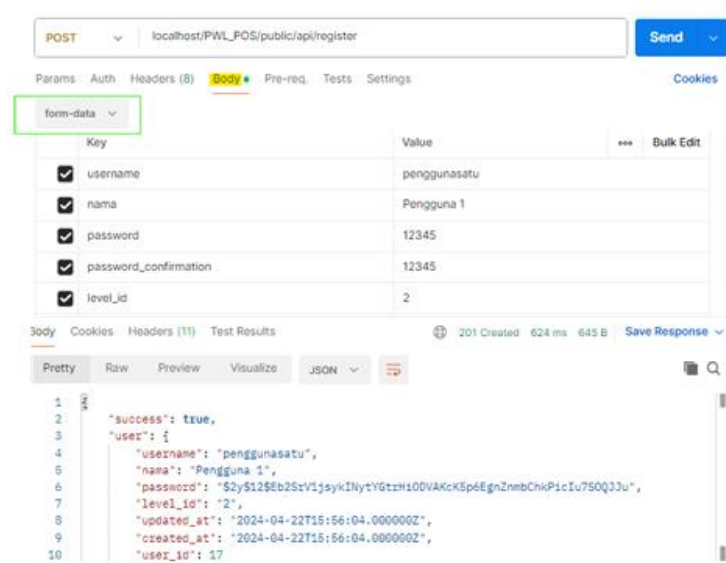
/*
|-----
| API Routes
|-----
|
| Here is where you can register API routes for your application. These
| routes are loaded by the RouteServiceProvider and all of them will
| be assigned to the "api" middleware group. Make something great!
|
*/

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.



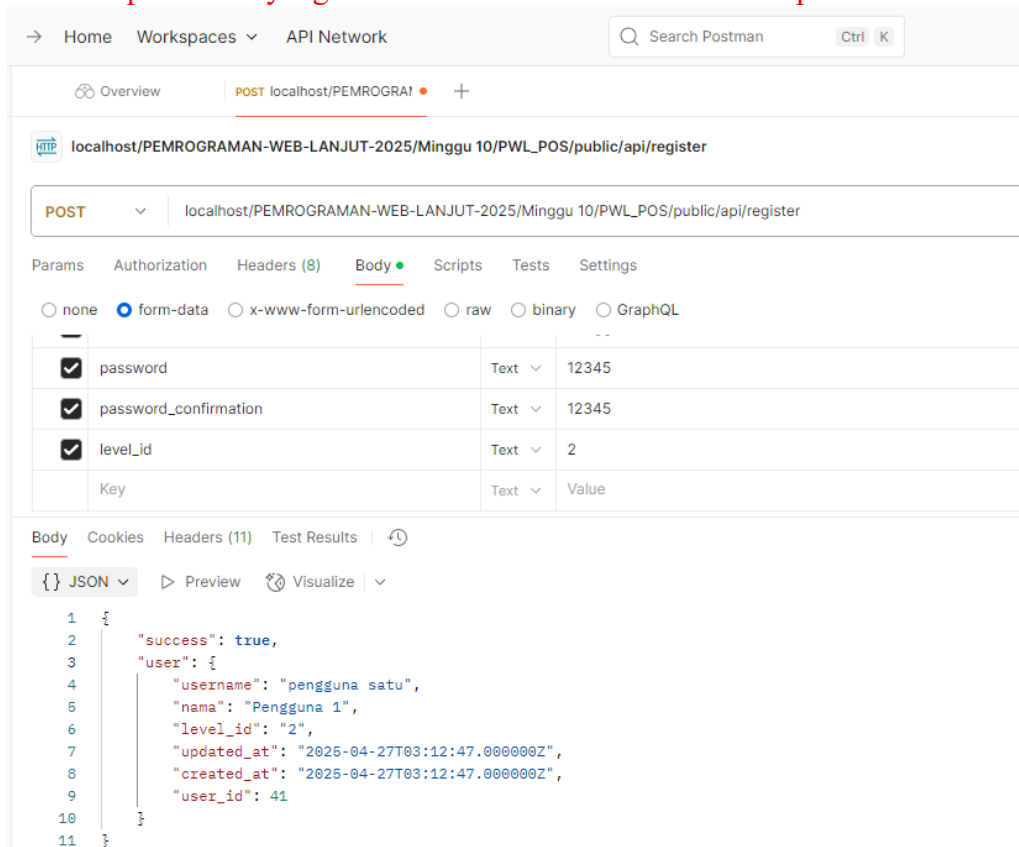
- Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.
12. Sekarang kita coba masukkan data. Klik tab Body dan pilih form-data. Isikan key sesuai dengan kolom data, serta isikan data registrasi menggunakan nilai yang Anda inginkan.



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

di atas.

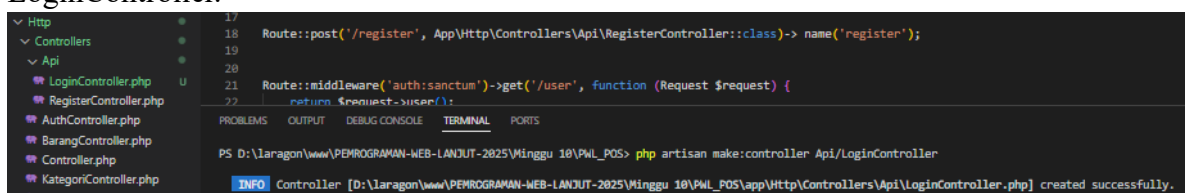
Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.



13. Lakukan commit perubahan file pada Github.

Praktikum 2 - Penerapan Layouting:

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.

```

Minggu 10 > PWL_POS > app > Http > Controllers > Api > LoginController.php > LoginController
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             'nama' => '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 succes
36         return response()->json([
37             'success' => true,
38             'user' => auth()->guard('api')->user(),
39             'token' => $token
40         ], 200);
41     }
42 }

```

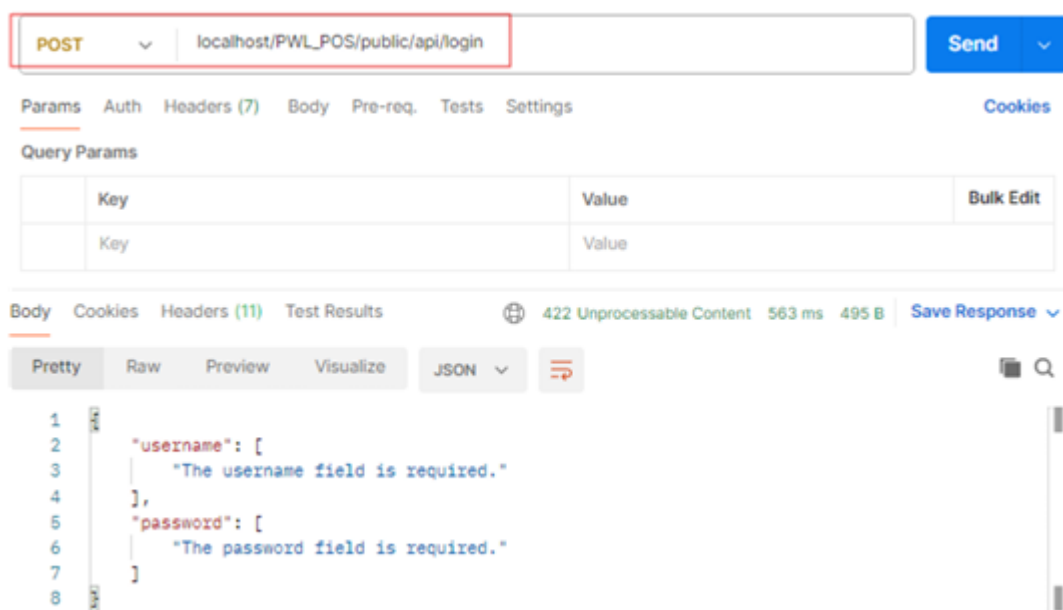
3. Berikutnya tambahkan route baru pada file api.php yaitu /login dan /user.

```

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

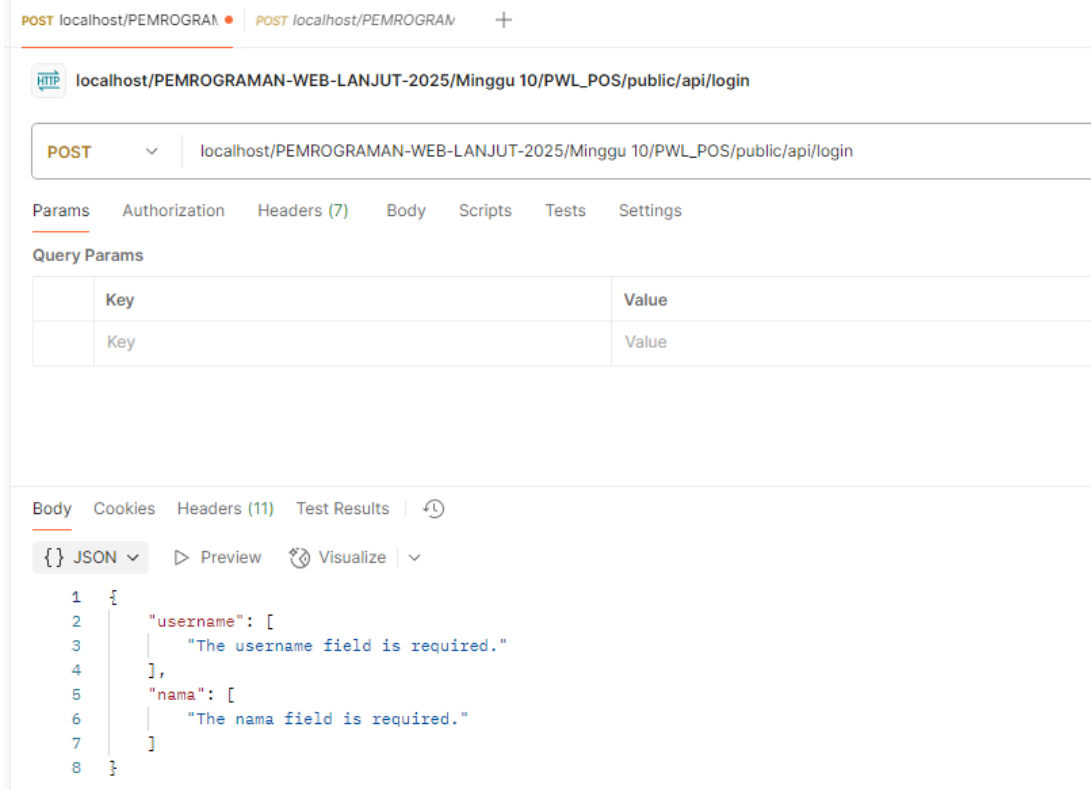
```

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.

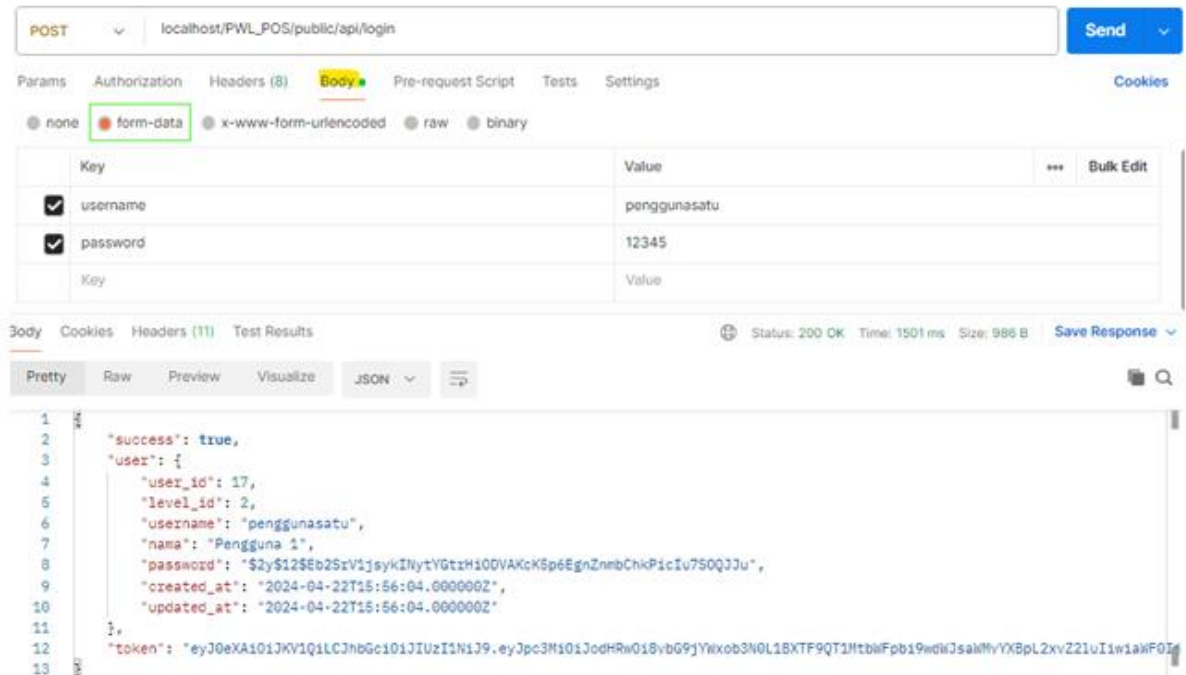


Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



5. 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.



Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/login

POST ▼ | localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/login

Params Authorization Headers (8) **Body** ● Scripts Tests Settings

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

Key	Value
<input checked="" type="checkbox"/> username	Text ▼ pengguna satu
<input checked="" type="checkbox"/> password	Text ▼ 12345
Key	Text ▼ Value

Body Cookies Headers (11) Test Results ↺

{} JSON ▼ ▶ Preview 🔗 Visualize ▼

```

1  {
2      "success": true,
3      "user": {
4          "user_id": 41,
5          "level_id": 2,
6          "username": "pengguna satu",
7          "nama": "Pengguna 1",
8          "user_profile_picture": null,
9          "created_at": "2025-04-27T03:12:47.000000Z",
10         "updated_at": "2025-04-27T03:12:47.000000Z"
11     },

```

Token untuk logout :

"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0L1BFTVJPJ1BTUFOLVdFQl1MQU5KVVQtMjAyNSI9NaW5nZ3U1MjAxMC9QV0xfUE9TL3B1YmxpYy9hcGkvbmG9naW4iLCJpYXQiOjE3NDUzNjExNzIsImV4cCI6MTc0NTc2NDc3MiwibmJmImJ0eXNzQ1NzYxMTcyLCJqdGkiOiJBdGs3NE04M2tDc3JMR3FwIiwic3ViIjoieDEiLCJwcnYiOiIOMWRModGzNGYxYjk4ZjcwZWZhbnJhYWVkZWY0MjM0MTM3MDA2OTBjIn0.8hJgF5v5iJTMPBGfG34dqTI_gQ5Y5HUFBLQ1W-rb99A"

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

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/login

POST localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/login

Params Authorization Headers (8) **Body** Scripts Tests Settings

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

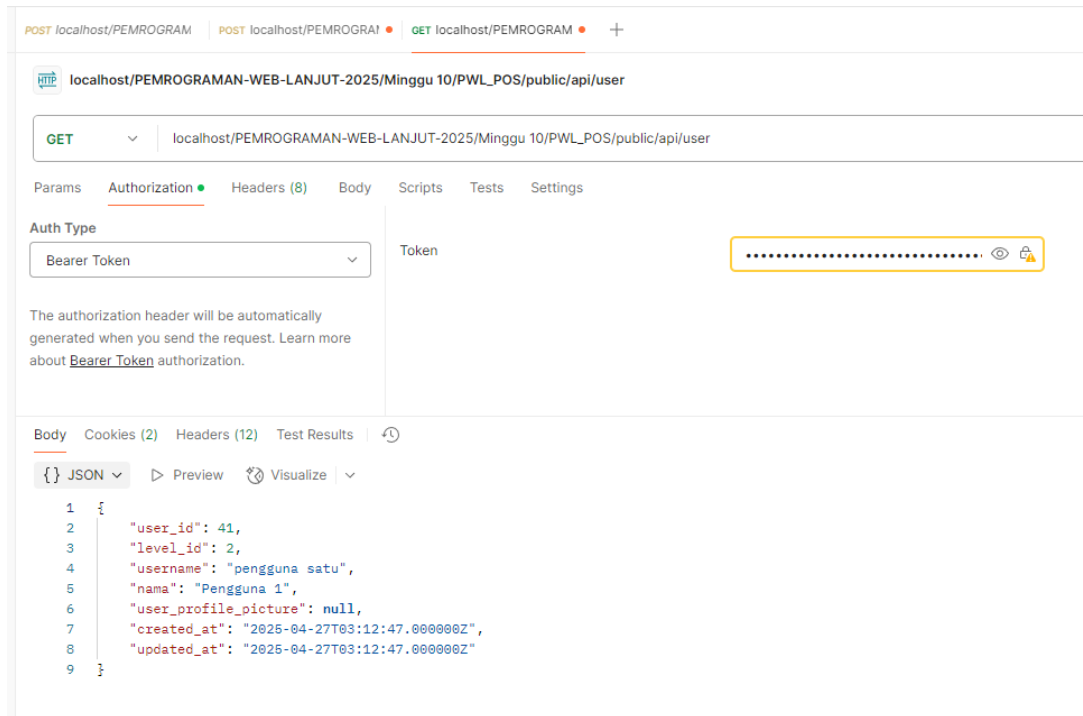
Key	Value
<input checked="" type="checkbox"/> username	Text ▼ penggunasatu
<input checked="" type="checkbox"/> password	Text ▼ 12345
Key	Text ▼ Value

Body Cookies Headers (11) Test Results ↺

{ } JSON ▼ ▶ Preview 📄 Visualize ▼

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

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.**



- Menggunakan method GET dengan url `localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/user`
- Lalu membuka bagian Authorization dan pilih Type Auth Bearer Token yang berfungsi memberi tahu server jika ada user yang sudah login dan mempunyai akses
- Lalu meng-copy token yang muncul pada hasil sebelumnya disaat login, Klik Send lalu akan muncul data user

8. Lakukan commit perubahan file pada Github.

Praktikum 3 - Praktikum 3 – Implementasi jQuery Datatable di AdminLTE :

1. Tambahkan kode berikut pada file .env
`JWT_SHOW_BLACKLIST_EXCEPTION=true`

```
Minggu 10 > PWL_POS > .env
62  JWT_SHOW_BLACKLIST_EXCEPTION=true
```

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

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\PWL_POS> php artisan make:controller Api/LogoutController

INFO Controller [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\PWL_POS\app\Http\Controllers\Api/LogoutController.php] created successfully.
```

3. Buka file tersebut dan ubah kode menjadi seperti berikut.

```

Minggu 10 > PWL_POS > app > Http > Controllers > Api > LogoutController.php > LogoutController
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 }

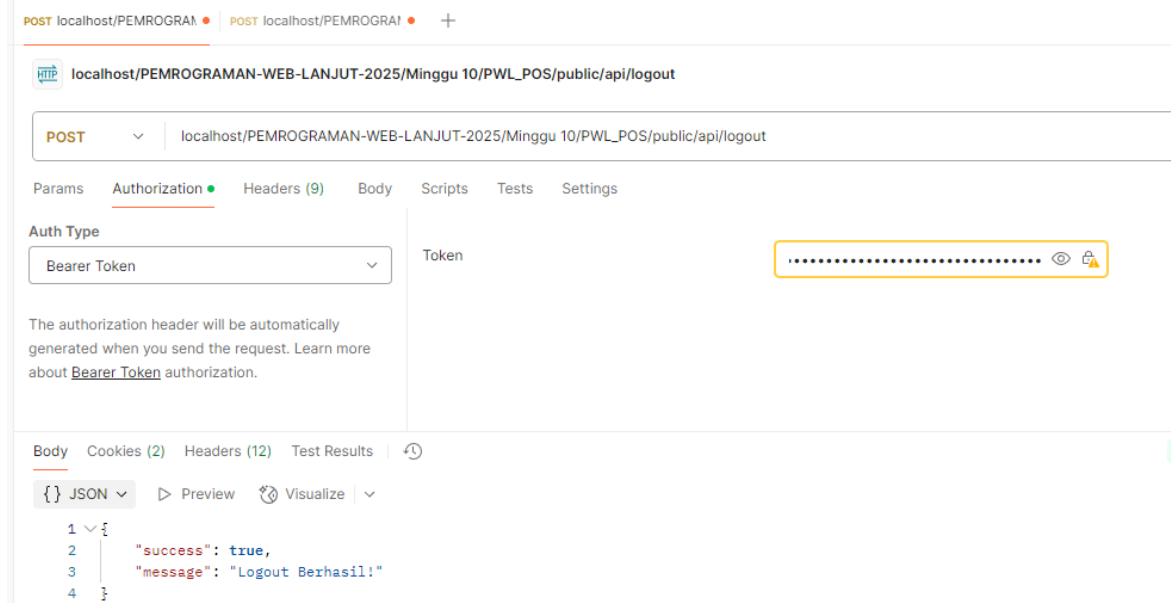
```

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 Filtering Datatables:

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

```
PS D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\PWL_POS> php artisan make:controller Api/LevelController

INFO Controller [D:\laragon\www\PEMROGRAMAN-WEB-LANJUT-2025\Minggu 10\PWL_POS\app\Http\Controllers\Api\LevelController.php] created successfully.
```

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

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > LevelController.php > LevelController > store

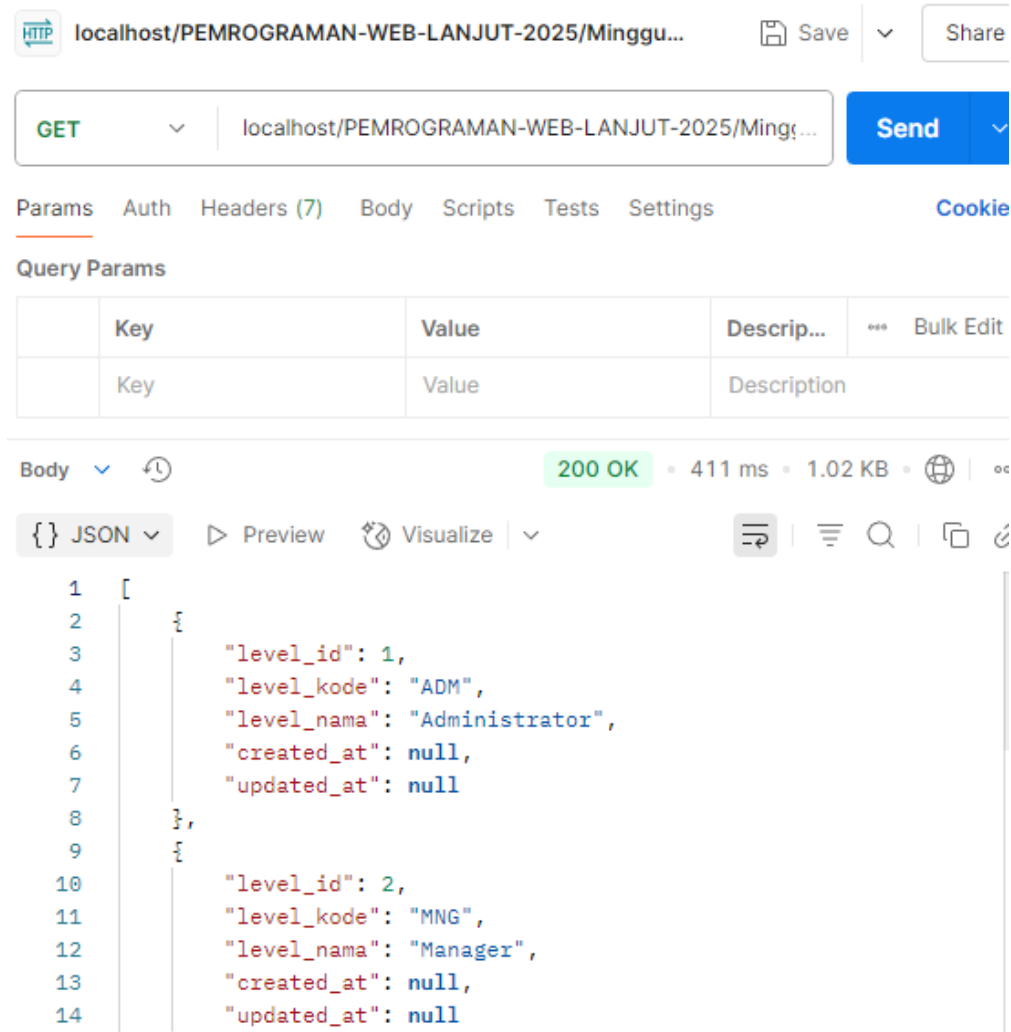
1  <?php
2
3  namespace App\Http\Controllers\Api;
4  use App\Http\Controllers\Controller;
5  use Illuminate\Http\Request;
6  use App\Models\LevelModel;
7
8  class LevelController extends Controller
9  {
10     public function index ()
11     {
12         return LevelModel::all();
13     }
14     public function store(Request $request)
15     {
16         $level = LevelModel::create($request->all());
17         return response()->json($level, 201);
18     }
19
20     public function show(LevelModel $level)
21     {
22         return LevelModel::find($level);
23     }
24
25     public function update(Request $request, LevelModel $level)
26     {
27         $level->update($request->all());
28         return LevelModel::find($level);
29     }
30
31     public function destroy(LevelModel $user)
32     {
33         $user->delete();
34
35         return response()->json([
36             'success' => true,
37             'message' => 'Data terhapus',
38         ]);
39     }
40 }
```

3. Kemudian kita lengkapi routes pada api.php.

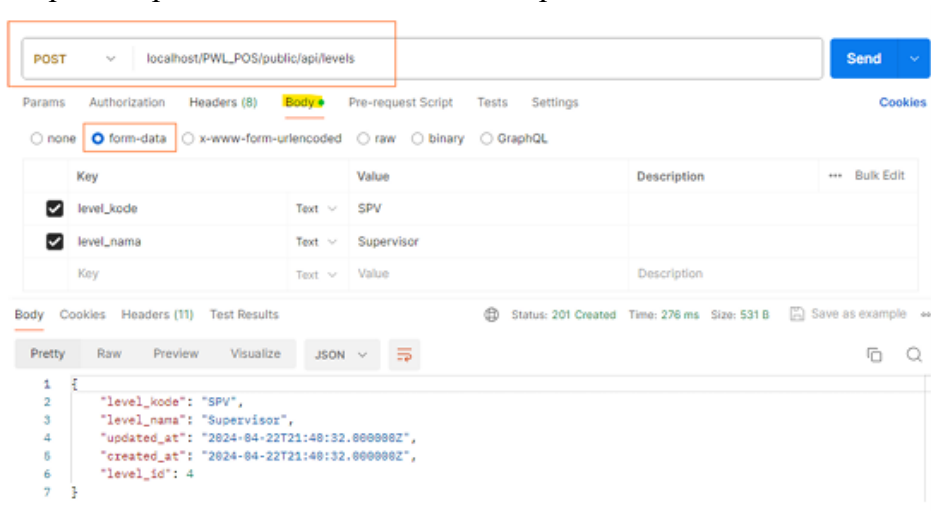
```
Route::get('levels', [LevelController::class, 'index']);
Route::post('levels', [LevelController::class, 'store']);
Route::get('levels/{level}', [LevelController::class, 'show']);
Route::put('levels/{level}', [LevelController::class, 'update']);
Route::delete('levels/{level}', [LevelController::class,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



- Dengan method GET maka akan muncul data semua Level yang ada di database seperti hasil diatas
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 screenshoot hasil percobaan Anda.

POST localhost/PEMROGRAMAN-WEB-LANJUT-2025/Mingt... Send

Params Auth Headers (9) Body Scripts Tests Settings Cookies

form-data

	key	value
<input checked="" type="checkbox"/>	level_kode	Text SPV
<input checked="" type="checkbox"/>	level_nama	Text Supervisor

Body 201 Created • 474 ms • 522 B • [Globe Icon] [More Icon]

{ } JSON Preview Visualize

```
1 {
2   "level_kode": "SPV",
3   "level_nama": "Supervisor",
4   "updated_at": "2025-04-27T16:05:33.000000Z",
5   "created_at": "2025-04-27T16:05:33.000000Z",
6   "level_id": 24
7 }
```

➤ Percobaan diatas untuk menguji apakah POST levels bisa menambahkan data baru ke database. Jika berhasil akan memunculkan hasil seperti diatas

6. Berikutnya lakukan percobaan menampilkan detail data.

Jelaskan dan berikan screenshoot hasil percobaan Anda.

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/1

GET localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/1

Params Authorization Headers (7) Body Scripts Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies (2) Headers (11) Test Results [Refresh Icon]

{ } 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 ]
```

- Untuk menampilkan detail data, cukup dengan method GET lalu menambahkan level_id di URLS setelah levels => *localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/1*
 - Setelah itu klik send maka akan muncul detail data sesuai dengan level_id yang dimasukkan
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.

HTTP PWL_POS (JS10) / localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/1

PUT localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/24?level_kode=SPR

Params Authorization Headers (8) Body Scripts Tests Settings

Query Params

Key	Value
level_kode	SPR
Key	Value

Body Cookies (2) Headers (11) Test Results

{ } JSON Preview Visualize

```

1 [
2   {
3     "level_id": 24,
4     "level_kode": "SPR",
5     "level_nama": "Supervisor",
6     "created_at": "2025-04-27T16:05:33.000000Z",
7     "updated_at": "2025-04-27T16:18:31.000000Z"
8   }
9 ]

```

- Digunakan untuk mengupdate data level tertentu. Request akan memodifikasi nilai nama_level berdasarkan ID yang diberikan.
8. Terakhir lakukan percobaan hapus data. Jelaskan dan berikan screenshot hasil percobaan Anda.

HTTP localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/24

DELETE localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/24

Params Authorization Headers (7) Body Scripts Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies (2) Headers (11) Test Results

{ } JSON Preview Visualize

```

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

```

- Untuk menampilkan hapus data, cukup dengan method DELETE lalu menambahkan level_id di URLS setelah levels => *localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/levels/1*
 - Setelah itu klik send maka akan muncul message bawah data sudah terhapus sesuai dengan level_id yang dimasukkan
9. Lakukan commit perubahan file pada Github.

TUGAS :

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

1. m_user:

The screenshot shows a REST client interface with the following details:

- URL:** localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu...
- Method:** GET
- Status:** 200 OK
- Response Time:** 946 ms
- Body:** JSON

The JSON response is as follows:

```
[
  {
    "user_id": 1,
    "level_id": 1,
    "username": "Admin",
    "nama": "Administrator",
    "user_profile_picture": null,
    "created_at": null,
    "updated_at": "2025-03-10T04:02:22.000000Z"
  },
  {
    "user_id": 2,
    "level_id": 2,
    "username": "manager",
    "nama": "Manager",
    "user_profile_picture": null,
  }
]
```


HTTP ... / localhost/PEMROGRAMAN-WEB-LANJUT-2...

Save

POST localhost/PEMROGRAMAN-WEB-LANJUT-2025/Ming...

Params Auth Headers (8) Body Scripts Tests Settings

form-data

	Key		Value	Descri...
<input checked="" type="checkbox"/>	level_id	Text	2	
<input checked="" type="checkbox"/>	username	Text	liyooo	
<input checked="" type="checkbox"/>	nama	Text	Cahya Aprilliyo	
<input checked="" type="checkbox"/>	password	Text	123456	

Body 201 Created 841 ms 536 B

{ } JSON Preview Visualize

```
1 {
2   "username": "liyooo",
3   "nama": "Cahya Aprilliyo",
4   "level_id": "2",
5   "updated_at": "2025-04-28T14:43:06.000000Z",
6   "created_at": "2025-04-28T14:43:06.000000Z",
7   "user_id": 44
8 }
```

HTTP ... / localhost/PEMROGRAMAN-WEB-LANJUT-2...

GET localhost/PEMROGRAMAN-WEB-LANJUT-2025/Mi

Params Auth Headers (8) Body Scripts Tests Settings

form-data

	Key		Value	Desci
<input checked="" type="checkbox"/>	level_id	Text	2	
<input checked="" type="checkbox"/>	username	Text	liyooo	
<input checked="" type="checkbox"/>	nama	Text	Cahya Aprilliyo	
<input checked="" type="checkbox"/>	password	Text	123456	

Body 200 OK 364 ms

{ } JSON Preview Visualize

```
1 {
2   "user_id": 1,
3   "level_id": 1,
4   "username": "Admin",
5   "nama": "Administrator",
6   "user_profile_picture": null,
7   "created_at": null,
8   "updated_at": "2025-03-10T04:02:22.000000Z"
9 }
```

HTTP ... / localhost/PEMROGRAMAN-WEB-LANJUT-2...

Save

PUT

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Mingt...

Send

Params Auth Headers (8) Body Scripts Tests Settings

form-data

	Key		Value	Descri...	...	Bl
<input checked="" type="checkbox"/>	level_id	Text	2			
<input checked="" type="checkbox"/>	username	Text	liyooo			
<input checked="" type="checkbox"/>	nama	Text	Cahya Aprilliyo			
<input checked="" type="checkbox"/>	password	Text	123456			

Body

200 OK • 342 ms • 547 B •

{ } JSON

Preview

Visualize

⌵ ⌵ 🔍

```
1 {
2   "user_id": 44,
3   "level_id": 2,
4   "username": "liyooo",
5   "nama": "Cahya",
6   "user_profile_picture": null,
7   "created_at": "2025-04-28T14:43:06.000000Z",
8   "updated_at": "2025-04-28T14:46:59.000000Z"
9 }
```

DEL localhost/PEMROGRAM. +

HTTP ... / localhost/PEMROGRAMAN-WEB-LANJUT-2... Save

DELETE localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu...

Params Auth Headers (8) Body Scripts Tests Settings

form-data

	Key	Value	Descri...
<input checked="" type="checkbox"/>	level_id	Text 2	
<input checked="" type="checkbox"/>	username	Text liyooo	
<input checked="" type="checkbox"/>	nama	Text Cahya Aprilliyo	
<input checked="" type="checkbox"/>	password	Text 123456	

Body 200 OK • 352 ms • 413 B

{ } JSON Preview Visualize

```

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

```

2. m_kategori:

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu... Save

GET localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/kategori

Params Auth Headers (6) Body Scripts Tests Settings

Query Params

Key	Value	Descrip...
Key	Value	Description

Body 200 OK • 699 ms • 1.4

{ } JSON Preview Visualize

```

1 [
2   {
3     "kategori_id": 1,
4     "kategori_kode": "SKC",
5     "kategori_nama": "Skincare",
6     "created_at": null,
7     "updated_at": null
8   },
9   {
10    "kategori_id": 2,

```

POST

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Ming...

Params

Auth

Headers (8)

Body

Scripts

Tests

Settings

form-data

	Key		Value	Descrip...
<input checked="" type="checkbox"/>	kategori_kode	Text	ELK	
<input checked="" type="checkbox"/>	kategori_nama	Text	Elektronik	
	Key	Text	Value	Description

Body

201 Created

1.92 s

500

{ } JSON

Preview

Visualize

1

{

2

"kategori_kode": "ELK",

3

"kategori_nama": "Elektronik",

4

"updated_at": "2025-04-28T14:26:50.000000Z",

5

"created_at": "2025-04-28T14:26:50.000000Z",

6

"kategori_id": 32

7

}

GET

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Ming...

Params

Auth

Headers (8)

Body

Scripts

Tests

Settings

form-data

	Key		Value	Descrip...	*
<input checked="" type="checkbox"/>	kategori_kode	Text	ELK		
<input checked="" type="checkbox"/>	kategori_nama	Text	Elektronik		
	Key	Text	Value	Description	

Body

200 OK

333 ms

472

{ } JSON

Preview

Visualize

1

{

2

"kategori_id": 1,

3

"kategori_kode": "SKC",

4

"kategori_nama": "Skincare",

5

"created_at": null,

6

"updated_at": null

7

}

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu...

Save

PUT

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/kategoris/32?kategori_kode=ELT

ParamsAuth

form-data

	Key		Value	Descrip...
<input checked="" type="checkbox"/>	kategori_kode	Text	ELK	
<input checked="" type="checkbox"/>	kategori_nama	Text	Elektronik	
	Key	Text	Value	Descripti

Body

200 OK423 ms

JSON

PreviewVisualize

```
1 {
2   "kategori_id": 32,
3   "kategori_kode": "ELT",
4   "kategori_nama": "Elektronik",
5   "created_at": "2025-04-28T14:26:50.000000Z",
6   "updated_at": "2025-04-28T14:29:10.000000Z"
7 }
```

localhost/PEMROGRAMAN...

NO

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu...

Save

DELETE

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/kategoris/32

ParamsAuthHeaders (8)BodyScriptsTestsSettings

form-data

	Key		Value	Descrip...
<input checked="" type="checkbox"/>	kategori_kode	Text	ELK	
<input checked="" type="checkbox"/>	kategori_nama	Text	Elektronik	
	Key	Text	Value	Description

Body

200 OK343 ms43

JSON

PreviewVisualize

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

3. m_barang :

GET localhost/PEMROGRAM... +

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/barangs

GET localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/barangs

Params Authorization Headers (6) Body Scripts Tests Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (11) Test Results

{ } JSON Preview Visualize

```
1 [
2   {
3     "barang_id": 1,
4     "kategori_id": 1,
5     "barang_kode": "BNG001",
6     "barang_nama": "Toner",
7     "harga_beli": 150000,
8     "harga_jual": 185000,
9     "created_at": null,
10    "updated_at": null
11  },
12  ...
13 ]
```

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu... Save Share

POST localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu... Send

Params Auth Headers (8) Body Scripts Tests Settings Cookies

form-data

<input checked="" type="checkbox"/>	barang_kode	Text	BNG011	
<input checked="" type="checkbox"/>	barang_nama	Text	Body Lotion	
<input checked="" type="checkbox"/>	harga_beli	Text	15000	
<input checked="" type="checkbox"/>	harga_jual	Text	18500	
	Key	Text	Value	Description

Body 201 Created • 1.07 s • 589 B •

{ } JSON Preview Visualize

```
1 {
2   "kategori_id": "1",
3   "barang_kode": "BNG011",
4   "barang_nama": "Body Lotion",
5   "harga_beli": "15000",
6   "harga_jual": "18500",
7   "updated_at": "2025-04-28T14:08:54.000000Z",
8   "created_at": "2025-04-28T14:08:54.000000Z",
9   "barang_id": 19
10 }
```

GET localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu 10/PWL_POS/public/api/barangs/1

Params Auth Headers (8) Body Scripts Tests Settings

form-data

<input checked="" type="checkbox"/>	barang_kode	Text	BNG011	
<input checked="" type="checkbox"/>	barang_nama	Text	Body Lotion	
<input checked="" type="checkbox"/>	harga_beli	Text	15000	
<input checked="" type="checkbox"/>	harga_jual	Text	18500	
	Key	Text	Value	Description

Body 200 OK 362 ms 525

{ } JSON Preview Visualize

```
2 {
3   "barang_id": 1,
4   "kategori_id": 1,
5   "barang_kode": "BNG001",
6   "barang_nama": "Toner",
7   "harga_beli": 150000,
8   "harga_jual": 185000,
9   "created_at": null,
10  "updated_at": null
11 }
```

PUT localhost/PEMROGRAMAN-WEB-LANJUT-2025/Ming...

Params Auth Headers (8) Body Scripts Tests Settings

form-data

<input checked="" type="checkbox"/>	barang_kode	Text	BNG011	
<input checked="" type="checkbox"/>	barang_nama	Text	Body Lotion	
<input checked="" type="checkbox"/>	harga_beli	Text	15000	
<input checked="" type="checkbox"/>	harga_jual	Text	18500	
	Key	Text	Value	Description

Body 200 OK 430 ms 727 B

{ } JSON Preview Visualize

```
12 {
13   "barang_id": 19,
14   "kategori_id": 1,
15   "barang_kode": "BNG011",
16   "barang_nama": "Lotion",
17   "harga_beli": 15000,
18   "harga_jual": 18500,
19   "created_at": "2025-04-28T14:08:54.000000Z",
20   "updated_at": "2025-04-28T14:12:30.000000Z"
21 }
```



localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu...

Save

DELETE

localhost/PEMROGRAMAN-WEB-LANJUT-2025/Minggu...

Params Auth Headers (8) **Body** Scripts Tests Settings

form-data

<input checked="" type="checkbox"/>	barang_kode	Text	BNG011	
<input checked="" type="checkbox"/>	barang_nama	Text	Body Lotion	
<input checked="" type="checkbox"/>	harga_beli	Text	15000	
<input checked="" type="checkbox"/>	harga_jual	Text	18500	
	Key	Text	Value	Description

Body

200 OK • 337 ms • 4

{ } JSON

Preview



Visualize



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