



Nama : Muhammad Bagus Indrawan

Kelas : TI-2H

NIM : 2241720217

Mata Kuliah : Pemrograman Web Lanjut (PWL)

Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis

Semester : 4 (empat) / 6 (enam)

Pertemuan ke- : 10 (tujuh)

## JOBSHEET 10

### RESTFUL API

#### 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"
```

```
PS C:\laragon\www\PWL_POS> php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"  
INFO Publishing assets.  
Copying file [C:\laragon\www\PWL_POS\vendor\tymon\jwt-auth\config\config.php] to [C:\laragon\www\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.



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.

```
PS C:\laragon\www\PWL_POS> php artisan jwt:secret  
jwt-auth secret [QJGeheQqTKy7smbZ8804akLEY1M1fAPeMSvXtBakI72lWR60eLE3SooBFX0DDgND] set successfully.
```

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

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

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



```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use Tymon\JWTAuth\Contracts\JWTSubject;
use Illuminate\Foundation\Auth\User as Authenticatable;

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

    public function getJWTCustomClaims(){
        return [];
    }

    protected $table = 'm_user';
    protected $primaryKey = 'user_id';
}
```

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

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

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

```
PS C:\laragon\www\PWL_POS> php artisan make:controller Api/RegisterController

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

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



```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Models\UserModel;
6  use App\Http\Controllers\Controller;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Validator;
9
10 class RegisterController extends Controller
11 {
12     public function __invoke(Request $request)
13     {
```



```
14 //set validation
15 $validator = Validator::make($request->all(), [
16     'username' => 'required',
17     'nama' => 'required',
18     'password' => 'required|min:5|confirmed',
19     'level_id' => 'required'
20 ]);
21
22 //if validations fails
23 if($validator->fails()){
24     return response()->json($validator->errors(), 422);
25 }
26
27 //create user
28 $user = UserModel::create([
29     'username' => $request->username,
30     'nama' => $request->nama,
31     'password' => bcrypt($request->password),
32     'level_id' => $request->level_id,
33 ]);
34
35 //return response JSON user is created
36 if($user){
37     return response()->json([
38         'success' => true,
39         'user' => $user,
40     ], 201);
41 }
42
43 //return JSON process insert failed
44 return response()->json([
45     'success' => false,
46 ], 409);
47 }
48 }
```

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



```
<?php

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

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

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

11. Jika sudah, kita akan melakukan uji coba REST API melalui aplikasi Postman. Buka aplikasi Postman, isi URL localhost/PWL\_POS/public/api/register serta method POST. Klik Send.

POST localhost/PWL\_POS/public/api/register Send

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

Query Params

Key	Value	Bulk Edit
Key	Value	

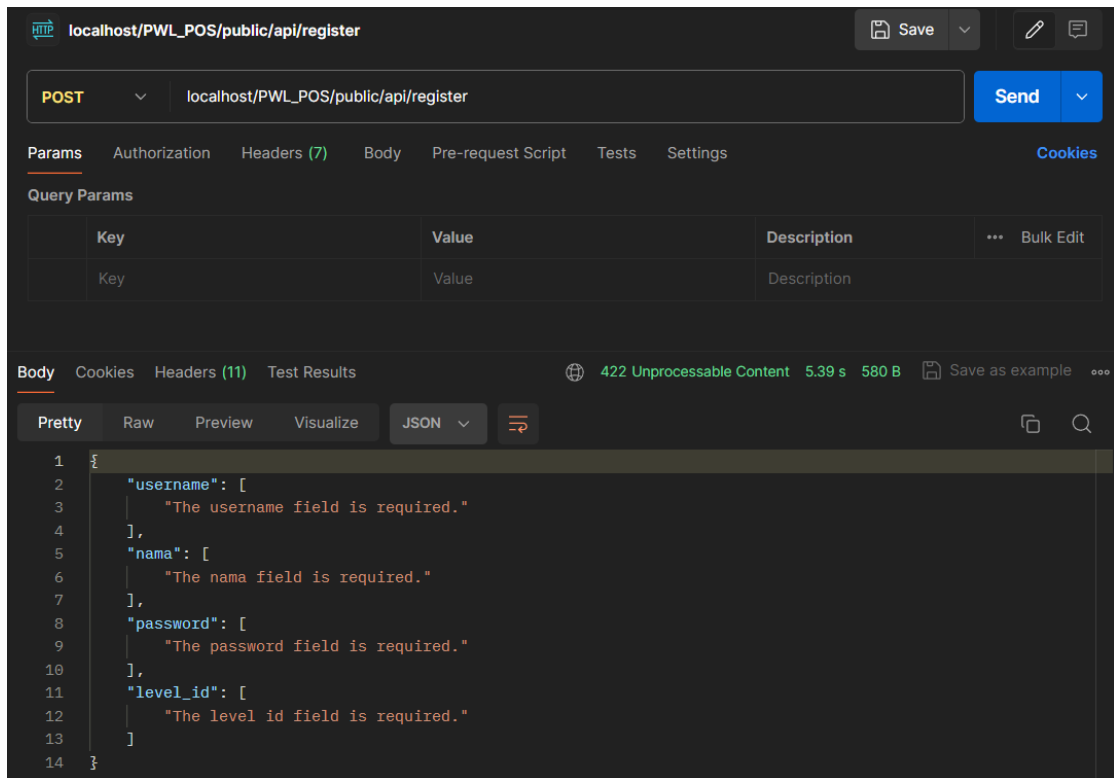
Body 422 Unprocessable Content 272 ms 534 B Save Response

Pretty Raw Preview Visualize JSON

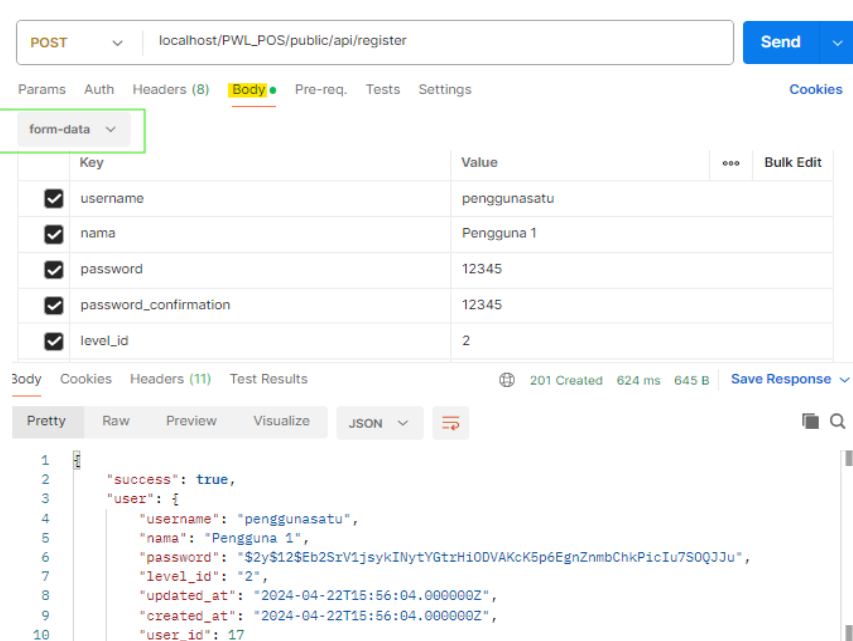
```
1 2 3 4 5 6 7 8 9 10 11
{
  "username": [
    "The username field is required."
  ],
  "nama": [
    "The nama field is required."
  ],
  "password": [
    "The password field is required."
  ]
}
```

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.

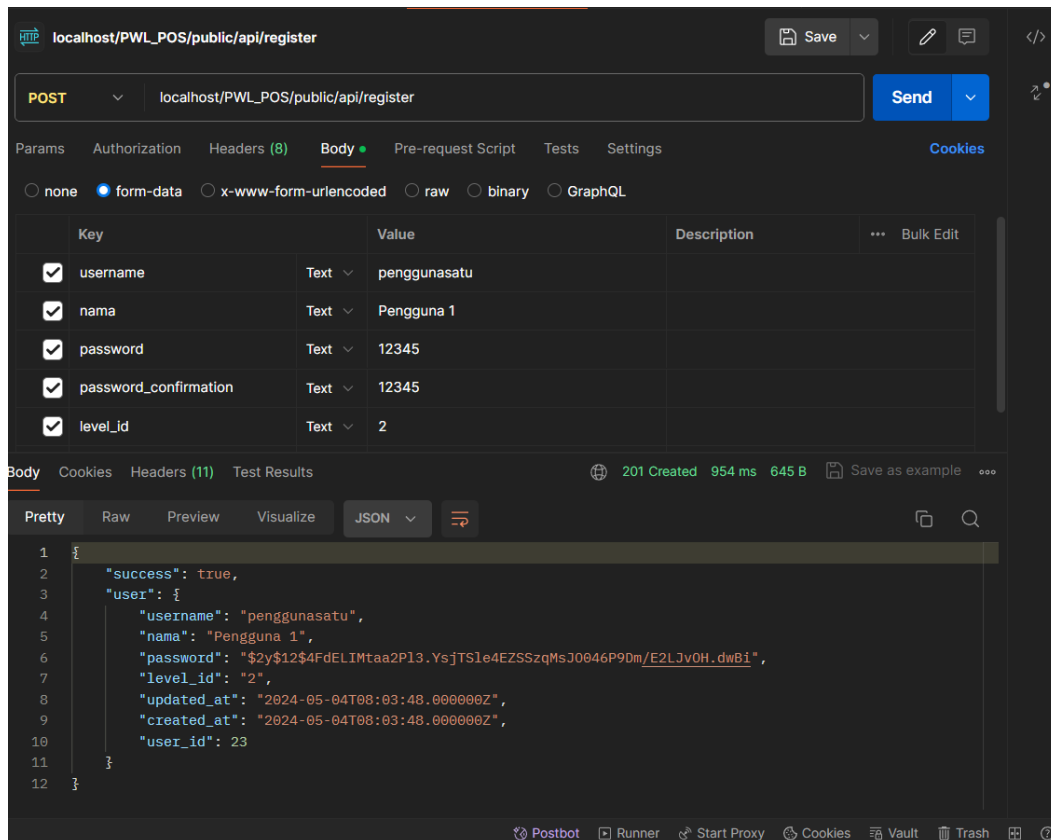






Setelah klik tombol Send, jika berhasil maka akan keluar pesan sukses seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot hasil percobaan Anda.



13. Lakukan commit perubahan file pada Github.

## Praktikum 2 – 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.

```
PS C:\laragon\www\PWL_POS> php artisan make:controller Api/LoginController

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

PS C:\laragon\www\PWL_POS>
```





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

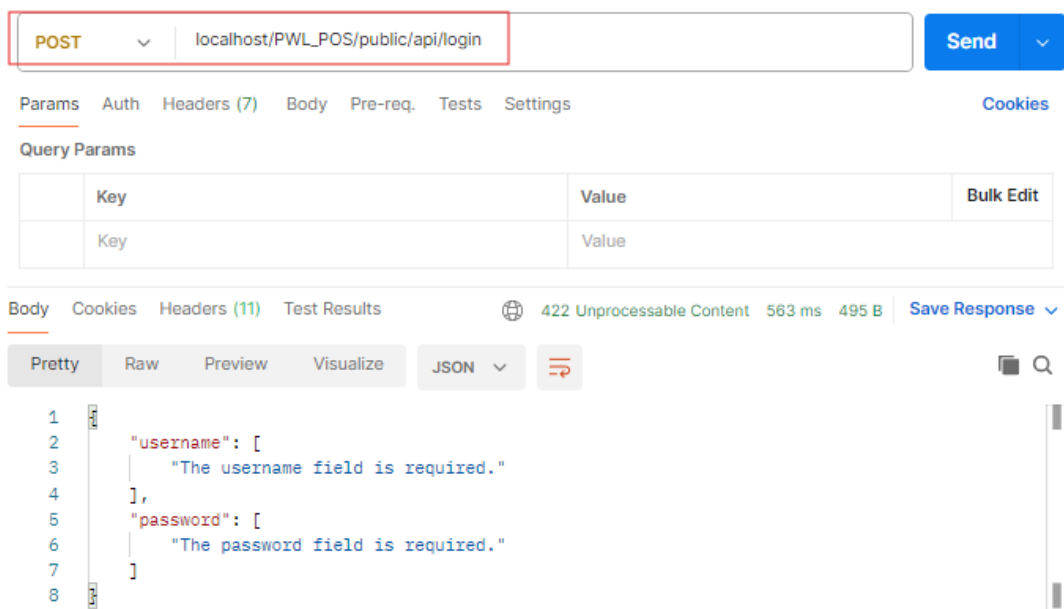
```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use Illuminate\Support\Facades\Validator;
8
9  class LoginController extends Controller
10 {
11     public function __invoke(Request $request)
12     {
13         //set validation
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()->user(),
39             'token' => $token
40         ], 200);
41     }
42 }
```

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



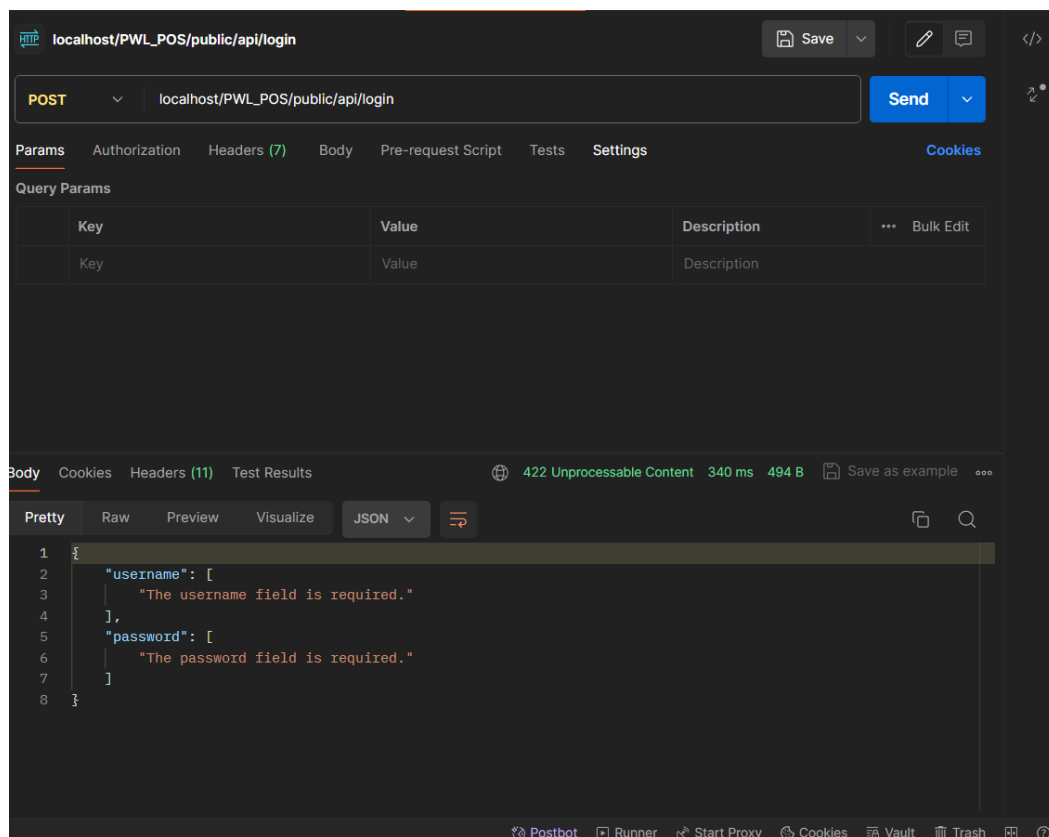
```
use App\Http\Controllers\Api\LoginController;  
  
Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');  
Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');  
Route::middleware('auth:api')->get('/user', function (Request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

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



Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot 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.



POST localhost/PWL\_POS/public/api/login Send

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

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

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

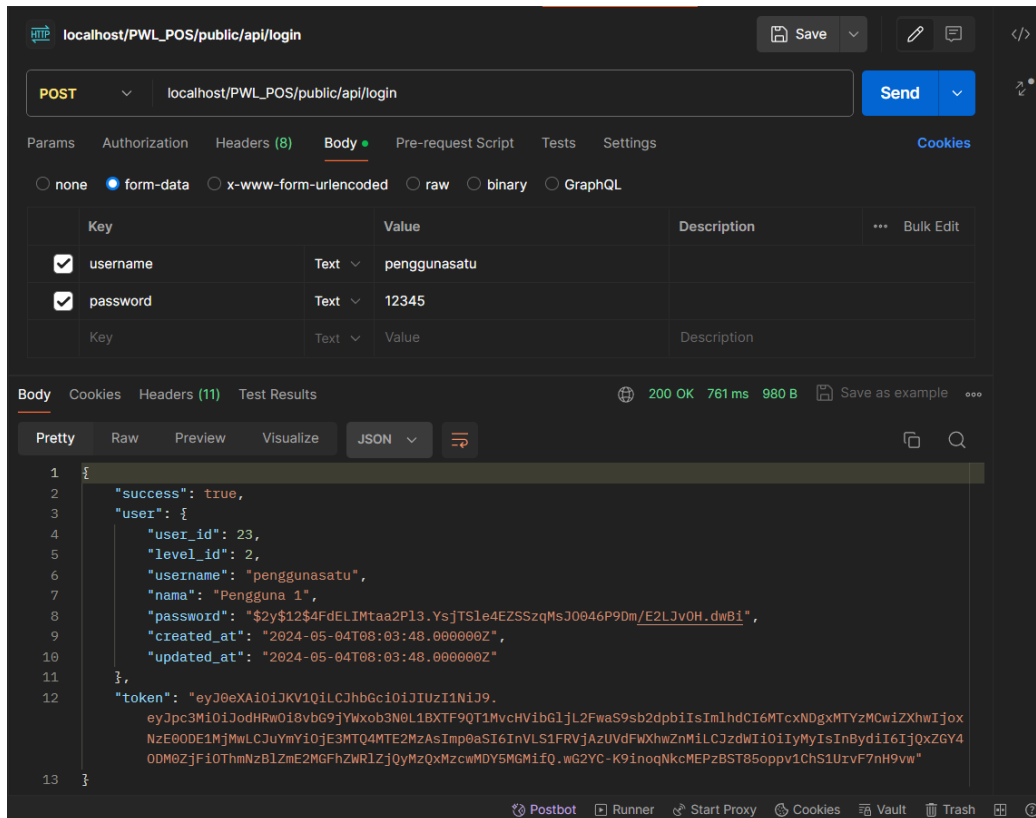
body Cookies Headers (11) Test Results Status: 200 OK Time: 1501 ms Size: 986 B Save Response

Pretty Raw Preview Visualize JSON

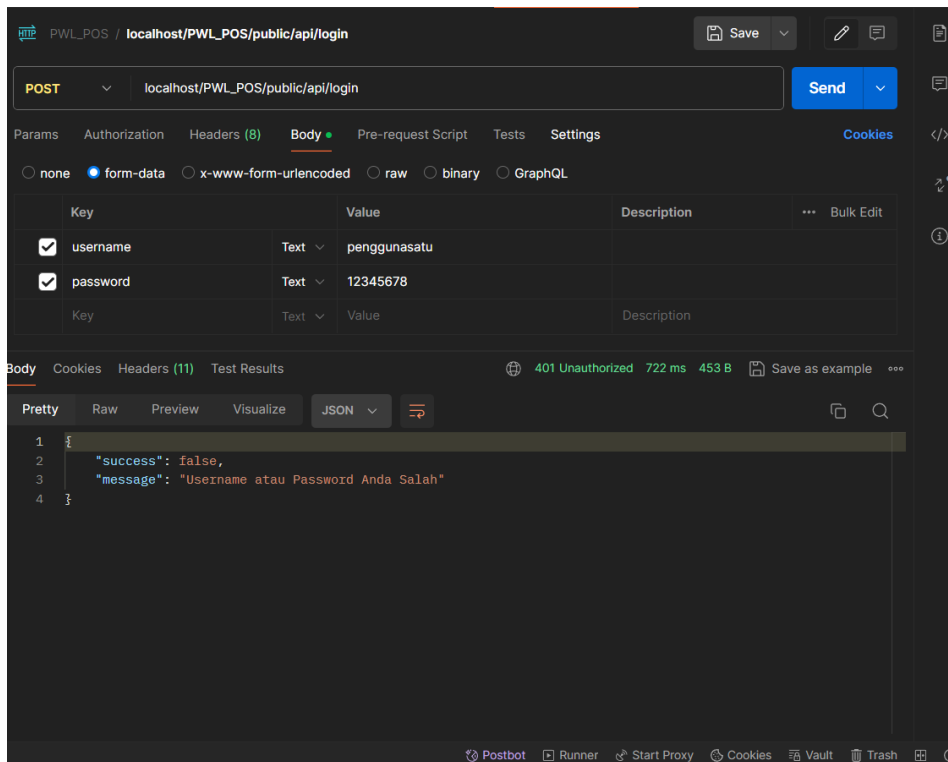
```
1  {
2    "success": true,
3    "user": {
4      "user_id": 17,
5      "level_id": 2,
6      "username": "penggunasatu",
7      "nama": "Pengguna 1",
8      "password": "$2y$12$Eb2SxV1jsykINytYGtzHi0DVAKcK5p6EgnZnmbChkPicIu7S0QJJJu",
9      "created_at": "2024-04-22T15:56:04.000000Z",
10     "updated_at": "2024-04-22T15:56:04.000000Z"
11   },
12   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ3c3MiOiJodHRwOi8vbG9jYXRob3N0L1BXTF9QT1MtbnVpbi9wdWJsaWVYbPL2xvZ21uIiwiaWF0Ij01"
13 }
```



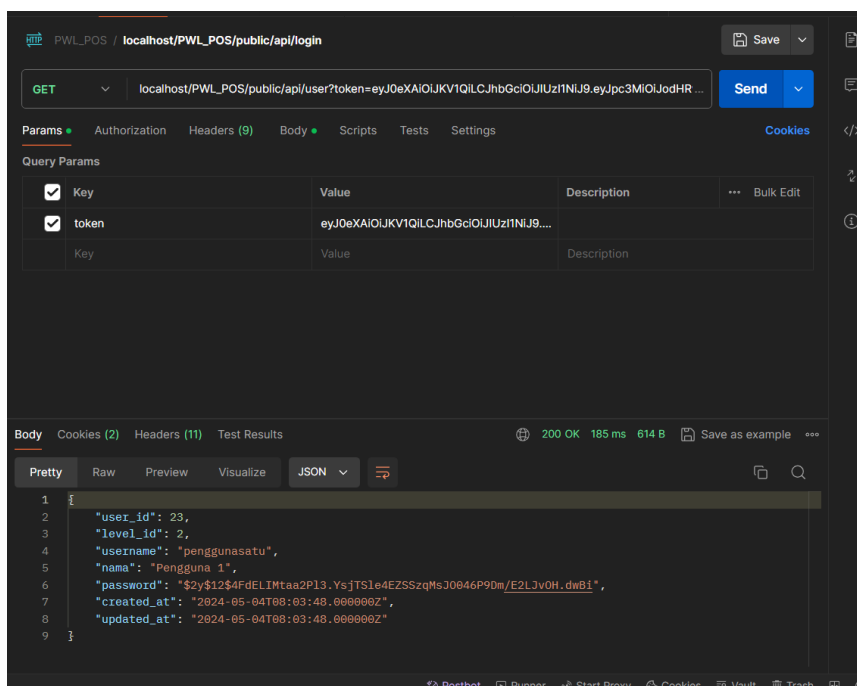
Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



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



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.**





Pada percobaan di atas, saya mengakses user berdasarkan token dari langkah login yang saya lakukan sebelumnya, lalu muncul output detail user seperti di atas, hal ini bisa terjadi karena saya memberi kode program Route middleware yang outputnya mengreturn semua data user (Langkah 3), sehingga output dapat muncul seperti diatas.

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`

```
PS C:\laragon\www\PWL_POS> php artisan make:controller Api/LogoutController  
  
INFO Controller [C:\laragon\www\PWL_POS\app\Http\Controllers\Api\LogoutController.php] created successfully.  
PS C:\laragon\www\PWL_POS>
```

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



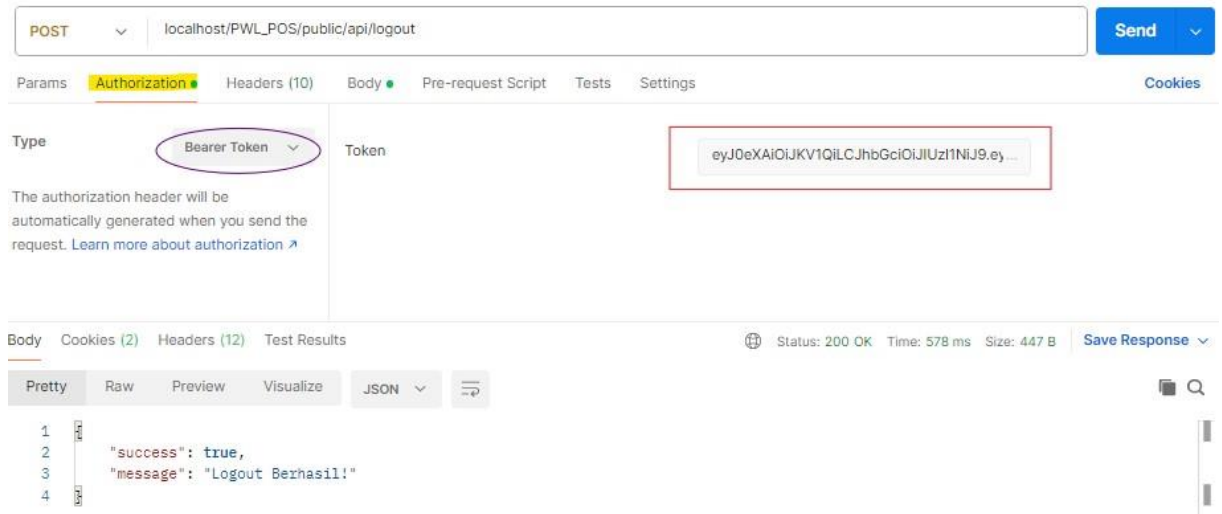


```
1  <?php
2
3  namespace App\Http\Controllers\Api;
4  use Illuminate\Http\Request;
5  use App\Http\Controllers\Controller;
6  use Tymon\JWTAuth\Facades\JWTAuth;
7  use Tymon\JWTAuth\Exceptions\JWTException;
8  use Tymon\JWTAuth\Exceptions\TokenExpiredException;
9  use Tymon\JWTAuth\Exceptions\TokenInvalidException;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

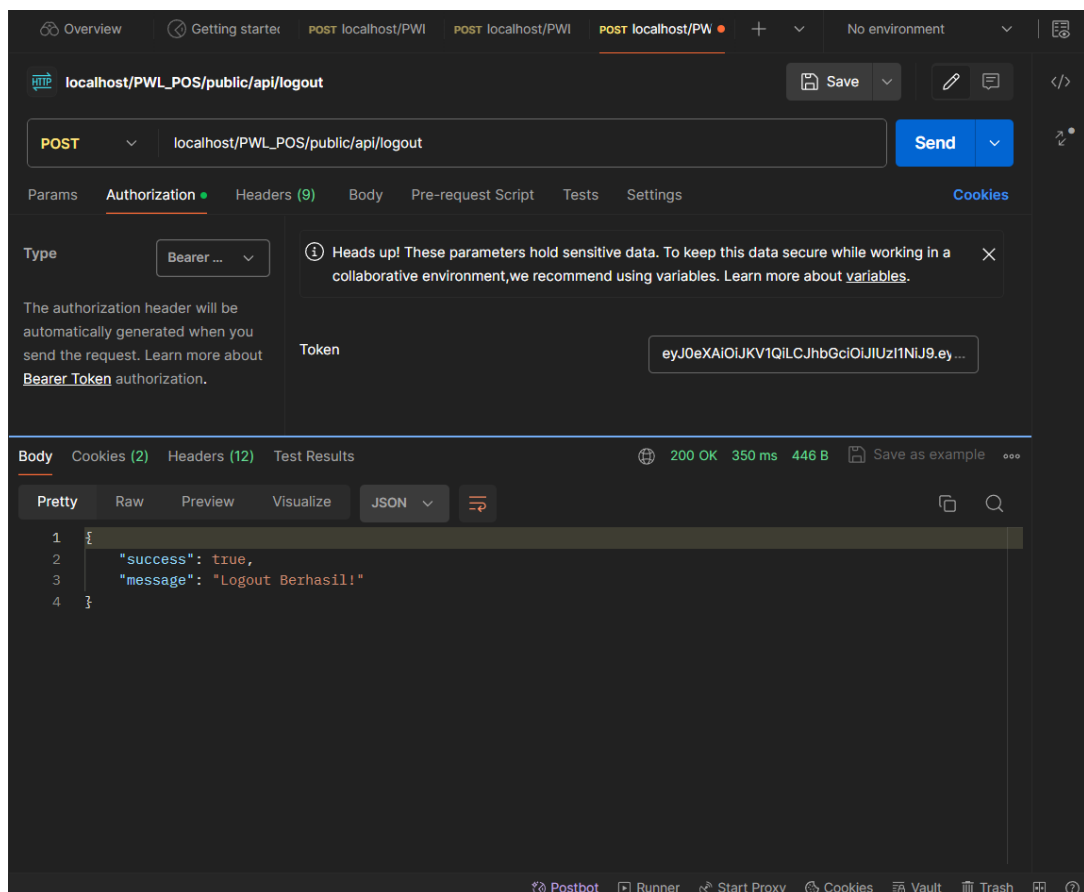
4. Lalu kita tambahkan routes pada api.php

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

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



Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.



7. Lakukan commit perubahan file pada Github.

## Praktikum 4 – Implementasi CRUD dalam RESTful API



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

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

`php artisan make:controller Api/LevelController`

```
PS C:\laragon\www\PWL_POS> php artisan make:controller Api/LevelController

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

PS C:\laragon\www\PWL_POS> |
```

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

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

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

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

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

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

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

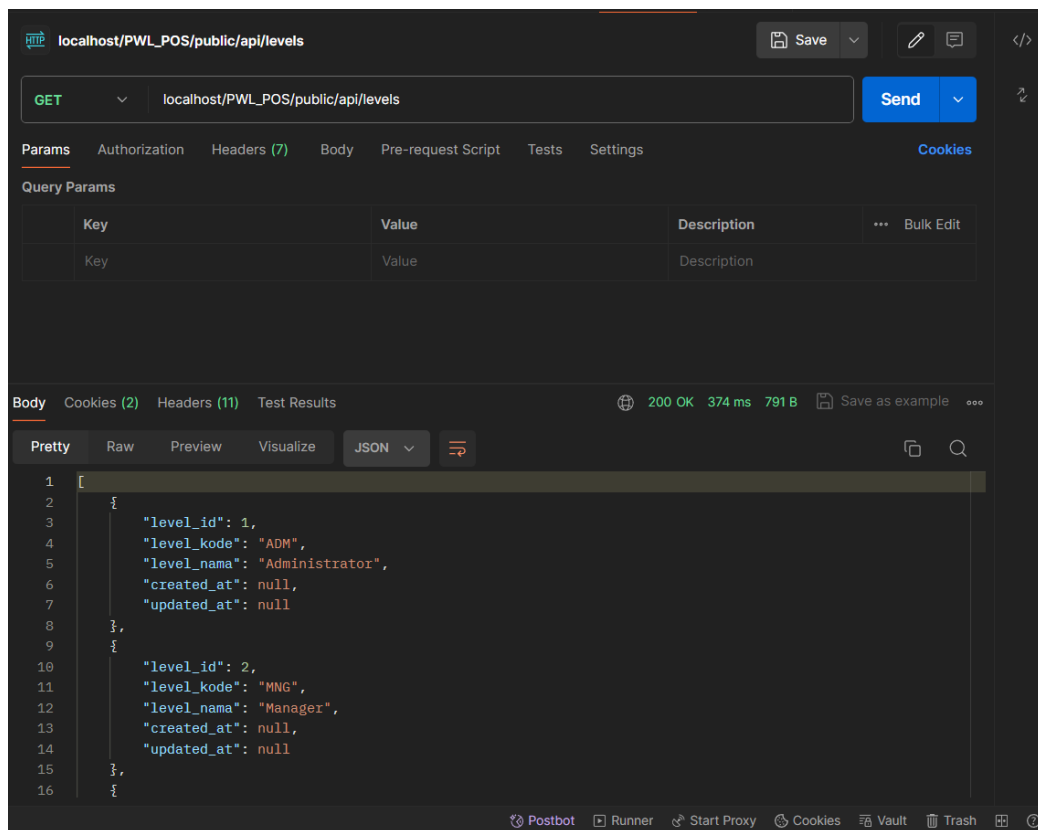
        return response()->json([
            'success' => true,
            'message' => 'Data terhapus',
        ]);
    }
}
```

3. Kemudian kita lengkapi routes pada `api.php`.



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

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



Percobaan diatas digunakan untuk menampilkan semua data level.

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



POST localhost/PWL\_POS/public/api/levels

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

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

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

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

Pretty Raw Preview Visualize JSON

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

Jelaskan dan berikan screenshoot hasil percobaan Anda.

localhost/PWL\_POS/public/api/levels

POST localhost/PWL\_POS/public/api/levels

Params Authorization Headers (10) **Body** Scripts Tests Settings Cookies

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

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

Body Cookies (2) Headers (11) Test Results 201 Created 329 ms 530 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "level_kode": "SPV",
3   "level_nama": "Supervisor",
4   "updated_at": "2024-05-04T09:17:43.000000Z",
5   "created_at": "2024-05-04T09:17:43.000000Z",
6   "level_id": 9
7 }
```

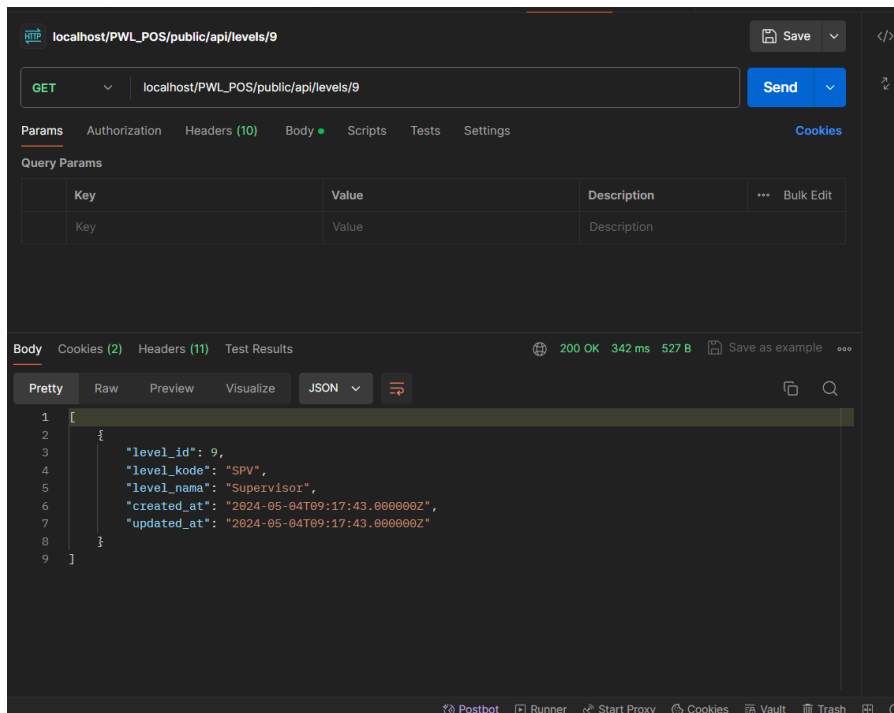
Percobaan diatas digunakan untuk menambahkan data pada tabel, pada percobaan diatas, saya menambahkan data dengan rincian :

- level\_id yang bernilai 9
- level\_kode yang berisi “SPV”
- level\_nama yang berisi “Supervisor”

Untuk updated\_at dan created\_at akan terisi secara otomatis.



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



Percobaan diatas digunakan untuk menampilkan data dari id tertentu, pada percobaan diatas, saya menampilkan data dari `level_id` yang bernilai 9.

7. 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.



PUT 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

Key	Value	Description	Bulk Edit
level_kode	SPR		

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 screenshot hasil percobaan Anda.

localhost/PWL\_POS/public/api/levels/5?level\_kode=SPR Save

PUT localhost/PWL\_POS/public/api/levels/5?level\_kode=SPR Send

Params Authorization Headers (10) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
level_kode	SPR		

Body Cookies (2) Headers (11) Test Results 200 OK 482 ms 527 B Save as example

Pretty Raw Preview Visualize JSON

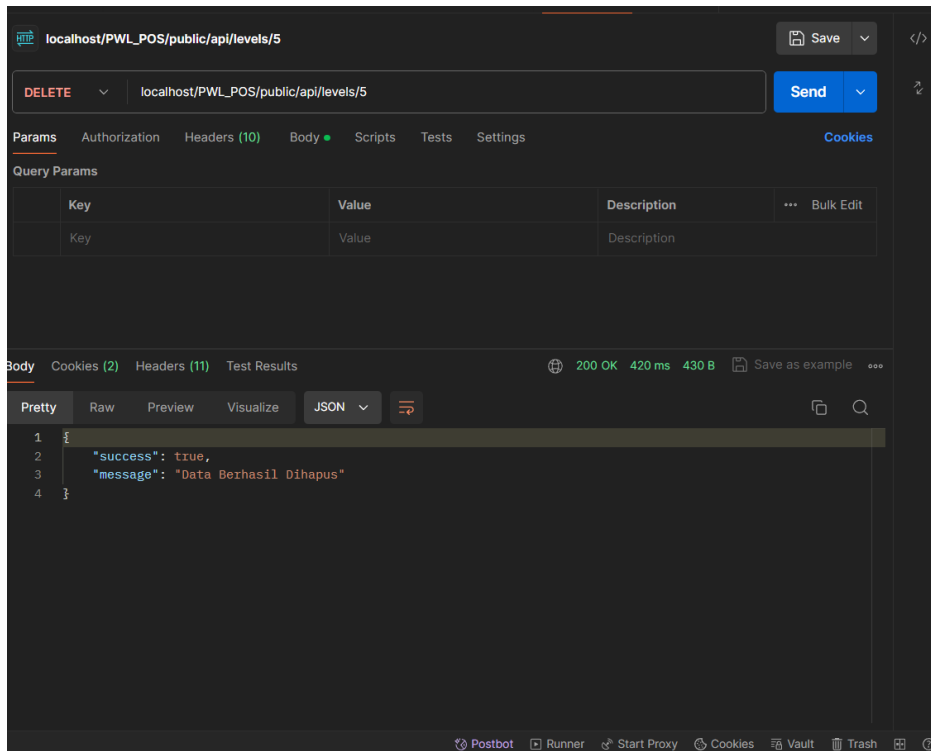
```
1 [
2   {
3     "level_id": 5,
4     "level_kode": "SPR",
5     "level_nama": "Supervisor",
6     "created_at": "2024-05-04T09:17:43.000000Z",
7     "updated_at": "2024-05-04T09:21:35.000000Z"
8   }
9 ]
```

Percobaan diatas digunakan untuk mengubah atau mengedit kolom tertentu, pada percobaan diatas, saya mengganti level\_kode yang sebelumnya bervalue “SPV” menjadi “SPR”





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



Percobaan diatas digunakan untuk menghapus data sesuai id yang dituju. Pada percobaan diatas, saya ingin menghapus data yang memiliki id yang bernilai 5.

9. Lakukan commit perubahan file pada Github.

## TUGAS

Implementasikan CRUD API pada tabel lainnya yaitu tabel m\_user, m\_kategori, dan m\_barang

1. CRUD API tabel m\_user



localhost/PWL\_POS/public/api/users

GET localhost/PWL\_POS/public/api/users

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (11) Test Results 200 OK 377 ms 2.91 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     "password": "$2y$12$2yeNlg8LTgWPOvKaM42cteBunGFuQch7oqwsbdSA0F6uQgNi0GoK1",
8     "created_at": null,
9     "updated_at": null
10  },
11  {
12    "user_id": 2,
13    "level_id": 2,
14    "username": "manager",
15    "nama": "Manager",
16    "password": "$2y$12$T/gcmUJtl1ezpMcBFDHfLeJhVR.CeWIM2DVC37xM3SvuUzX0K6Jke",
17    "created_at": null,
18    "updated_at": null
19  },
20 }
```

PWL\_POS / localhost/PWL\_POS/public/api/users

POST localhost/PWL\_POS/public/api/users

Params Authorization Headers (8) Body Scripts 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 1		
<input checked="" type="checkbox"/> username	Text lithiaaaaa		
<input checked="" type="checkbox"/> nama	Text Indra		
<input checked="" type="checkbox"/> password	Text 12345		
Key	Text Value	Description	

Body Cookies Headers (11) Test Results 201 Created 451 ms 558 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "level_id": "1",
3   "username": "lithiaaaaa",
4   "nama": "Indra",
5   "password": "12345",
6   "updated_at": "2024-05-04T13:55:04.000000Z",
7   "created_at": "2024-05-04T13:55:04.000000Z",
8   "user_id": 25
9 }
```



The image shows two screenshots of the Postman application interface. The top screenshot shows a GET request to `localhost/PWL_POS/public/api/users/25` with a status of 200 OK, 336 ms, and 732 B. The response is a JSON object with the following data: `{ "user_id": 25, "level_id": 1, "username": "lithiaaaaa", "nama": "Indra", "password": "12345", "created_at": "2024-05-04T13:55:04.000000Z", "updated_at": "2024-05-04T13:55:04.000000Z" }`. The bottom screenshot shows a PUT request to `localhost/PWL_POS/public/api/users/25?nama=Bagus Indra` with a status of 200 OK, 377 ms, and 738 B. The response is a JSON object with the following data: `{ "user_id": 25, "level_id": 1, "username": "lithiaaaaa", "nama": "Bagus Indra", "password": "12345", "created_at": "2024-05-04T13:55:04.000000Z", "updated_at": "2024-05-04T13:58:17.000000Z" }`.



DELETE localhost/PWL\_POS/public/api/users/25

Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (11) Test Results 200 OK 362 ms 430 B Save as example

Pretty Raw Preview Visualize JSON

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

Postbot Runner Start Proxy Cookies Vault Trash

## 2. CRUD API tabel m\_kategori

localhost/PWL\_POS/public/api/kategoris

GET localhost/PWL\_POS/public/api/kategoris

Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (11) Test Results 200 OK 2.42 s 921 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "kategori_id": 1,
4     "kategori_kode": "KTG001",
5     "kategori_nama": "Elektronik",
6     "created_at": null,
7     "updated_at": null
8   },
9   {
10    "kategori_id": 2,
11    "kategori_kode": "KTG002",
12    "kategori_nama": "Pakaian",
13    "created_at": null,
14    "updated_at": null
15  },
16 ]
```

Postbot Runner Start Proxy Cookies Vault Trash



Postman interface showing a successful POST request to `localhost/PWL_POS/public/api/kategoris`. The request body is a JSON object with the following data:

Key	Value	Description
kategori_kode	KTG006	
kategori_nama	Lainnya	

The response is a JSON object with the following data:

```
{  "kategori_kode": "KTG006",  "kategori_nama": "Lainnya",  "updated_at": "2024-05-04T13:26:15.000000Z",  "created_at": "2024-05-04T13:26:15.000000Z",  "kategori_id": 10}
```

Postman interface showing a successful GET request to `localhost/PWL_POS/public/api/kategoris/10`. The response is a JSON object with the following data:

Key	Value	Description
kategori_id	10	
kategori_kode	KTG006	
kategori_nama	Lainnya	
created_at	2024-05-04T13:26:15.000000Z	
updated_at	2024-05-04T13:26:15.000000Z	



PUT localhost/PWL\_POS/public/api/kategoris/10?kategori\_nama=Tanaman

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
kategori_nama	Tanaman	

Body Cookies Headers (11) Test Results 200 OK 352 ms 537 B Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "kategori_id": 10,
4     "kategori_kode": "KTG006",
5     "kategori_nama": "Tanaman",
6     "created_at": "2024-05-04T13:26:15.000000Z",
7     "updated_at": "2024-05-04T13:34:22.000000Z"
8   }
9 ]
```

DELETE localhost/PWL\_POS/public/api/kategoris/10

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
kategori_nama	Tanaman	

Body Cookies Headers (11) Test Results 200 OK 346 ms 430 B Save as example

Pretty Raw Preview Visualize JSON

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

### 3. CRUD API tabel m\_barang



PWL\_POS / localhost/PWL\_POS/public/api/users

GET localhost/PWL\_POS/public/api/barangs

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (11) Test Results 200 OK 338 ms 2.14 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "barang_id": 1,
3   "barang_kode": "BRG001",
4   "kategori_id": 1,
5   "barang_nama": "Smartphone",
6   "harga_beli": 1000000,
7   "harga_jual": 1500000,
8   "created_at": null,
9   "updated_at": null
10 },
11 {
12   "barang_id": 2,
13   "barang_kode": "BRG002",
14   "kategori_id": 1,
15   "barang_nama": "Laptop",
16   "harga_beli": 3000000,
17   "harga_jual": 4100000,
18   "created_at": null,
19   "updated_at": "2024-04-17T06:18:31.000000Z"
20 }
```

PWL\_POS / localhost/PWL\_POS/public/api/barangs

POST localhost/PWL\_POS/public/api/barangs

Params Authorization Headers (8) Body Scripts Tests Settings Cookies

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

Key	Value	Description	Bulk Edit
barang_kode	Text PUK		
kategori_id	Text 10		
barang_nama	Text Pupuk Jagung		

Body Cookies Headers (11) Test Results 201 Created 345 ms 597 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "barang_kode": "PUK",
3   "kategori_id": "10",
4   "barang_nama": "Pupuk Jagung",
5   "harga_beli": "50000",
6   "harga_jual": "60000",
7   "updated_at": "2024-05-04T13:44:37.000000Z",
8   "created_at": "2024-05-04T13:44:37.000000Z",
9   "barang_id": 13
10 }
```





PWL\_POS / localhost/PWL\_POS/public/api/barangs

GET localhost/PWL\_POS/public/api/barangs/1 Send

Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (11) Test Results 200 OK 255 ms 541 B Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "barang_id": 1,
3   "barang_kode": "BRG001",
4   "kategori_id": 1,
5   "barang_nama": "Smartphone",
6   "harga_beli": 1000000,
7   "harga_jual": 1500000,
8   "created_at": null,
9   "updated_at": null
10 }
11
12 ]
```

Postbot Runner Start Proxy Cookies Vault Trash

PWL\_POS / localhost/PWL\_POS/public/api/barangs

PUT localhost/PWL\_POS/public/api/barangs/13?harga\_beli=45000 Send

Params Authorization Headers (7) Body Scripts Tests Settings Cookies

Query Params

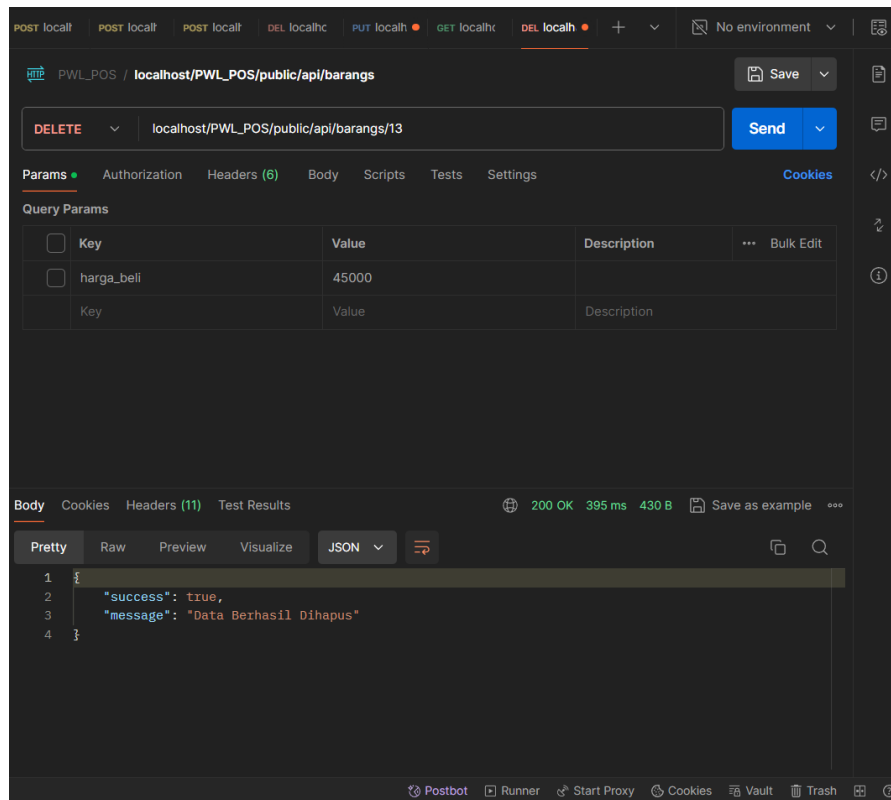
Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (11) Test Results 200 OK 336 ms 751 B Save as example

Pretty Raw Preview Visualize JSON

```
13 {
14   "barang_id": 13,
15   "barang_kode": "PUK",
16   "kategori_id": 10,
17   "barang_nama": "Pupuk Jagung",
18   "harga_beli": 45000,
19   "harga_jual": 60000,
20   "created_at": "2024-05-04T13:44:37.000000Z",
21   "updated_at": "2024-05-04T13:49:53.000000Z"
22 }
```

Postbot Runner Start Proxy Cookies Vault Trash



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