



Nama : Avicenna Mumtaza

Kelas/Absen : TI-2H/07

NIM : 2241720112

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 [Q]GeheQqTKy7smbZ8804akLEY1M1fAPeMsvxt8akI72lWR60eLE35ooBFx00DgND 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.



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>

```
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.

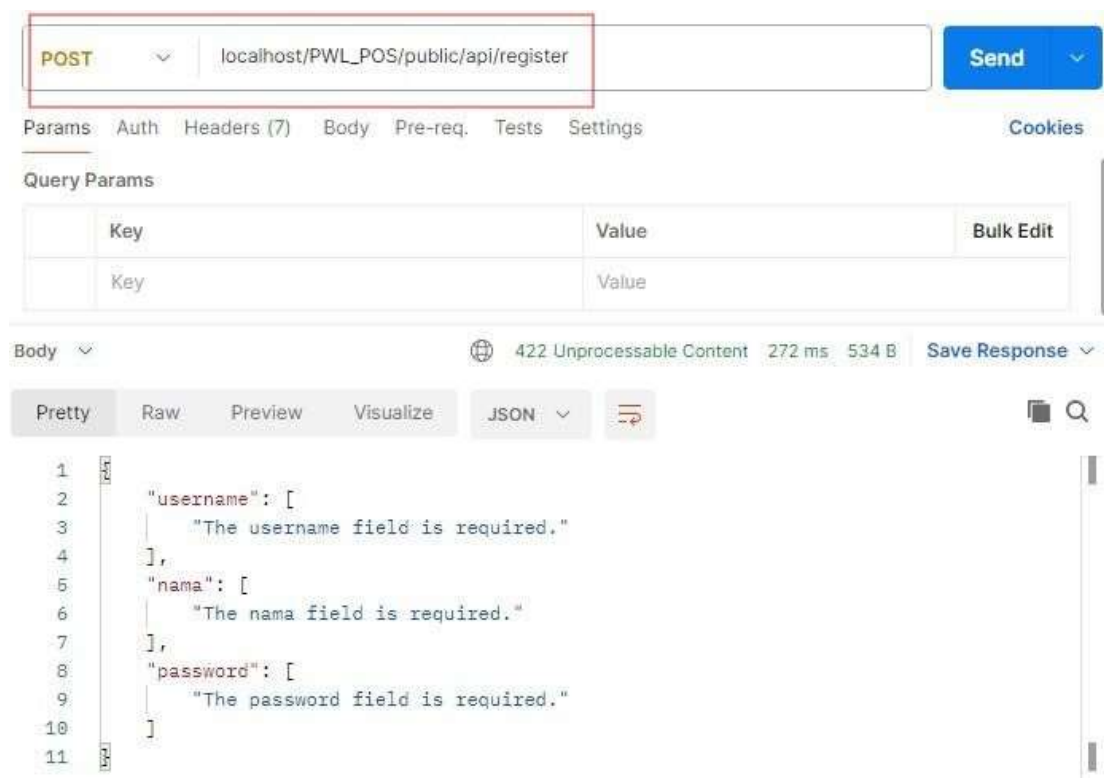


```
<?php
use App\Http\Controllers\Api\RegisterController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;

/*
|
| 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!
|
|-----
*/

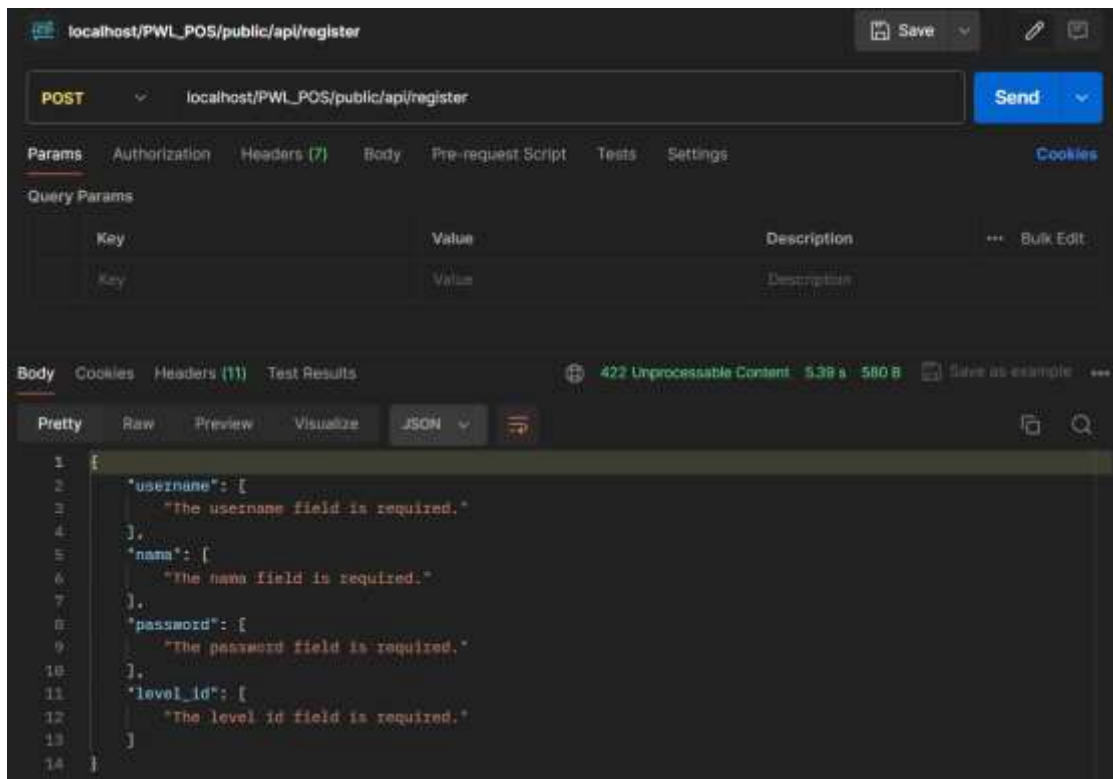
Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
```

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.

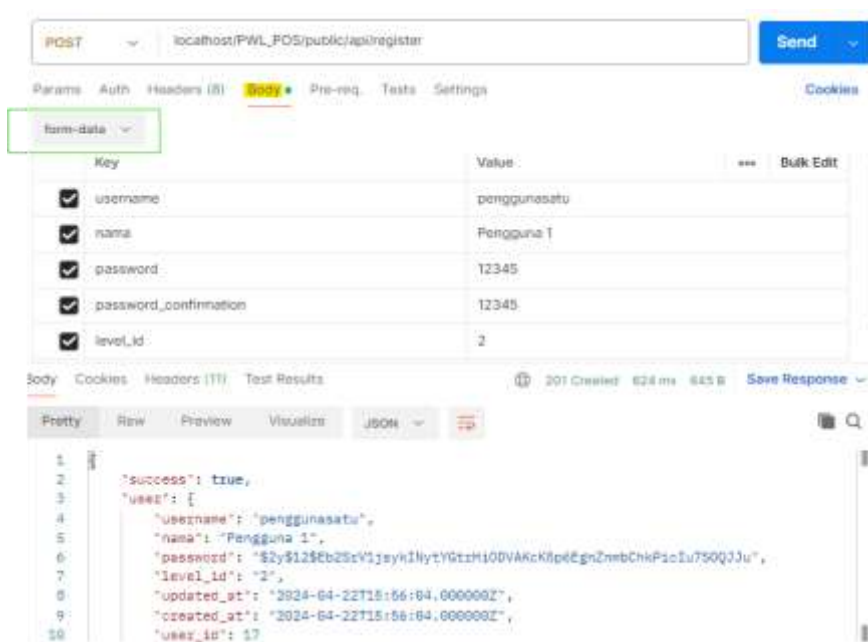


Jika berhasil akan muncul error validasi seperti gambar di atas.

Lakukan percobaan yang sama dan berikan screenshot 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.





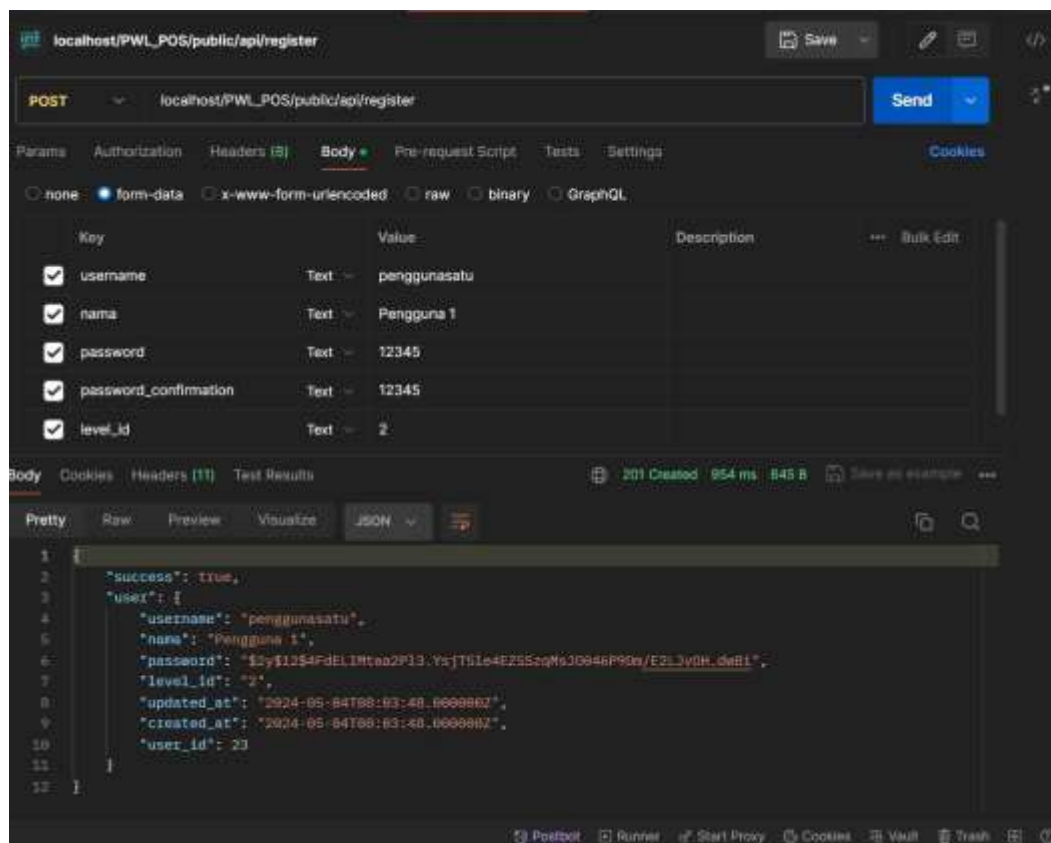
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420

<http://www.polinema.ac.id>

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.



```
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.

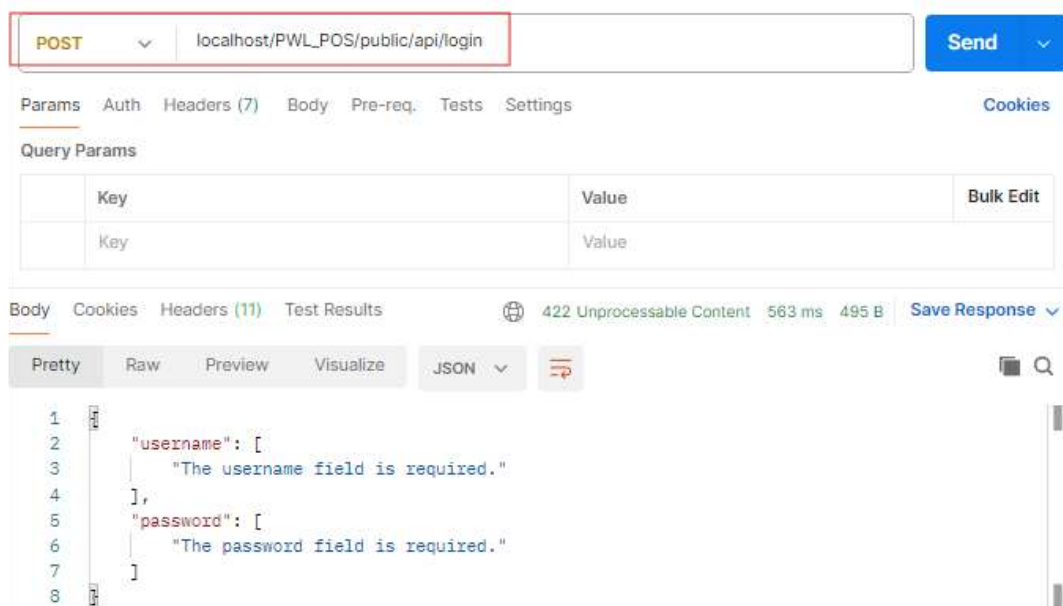
```
9  class LoginController extends Controller
10 {
11     public function __invoke(Request $request)
12     {
13         //set validation
14         $validator = Validator::make($request->all(), [
15             'username' => 'required',
16             'password' => 'required'
17         ]);
18
19         //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.



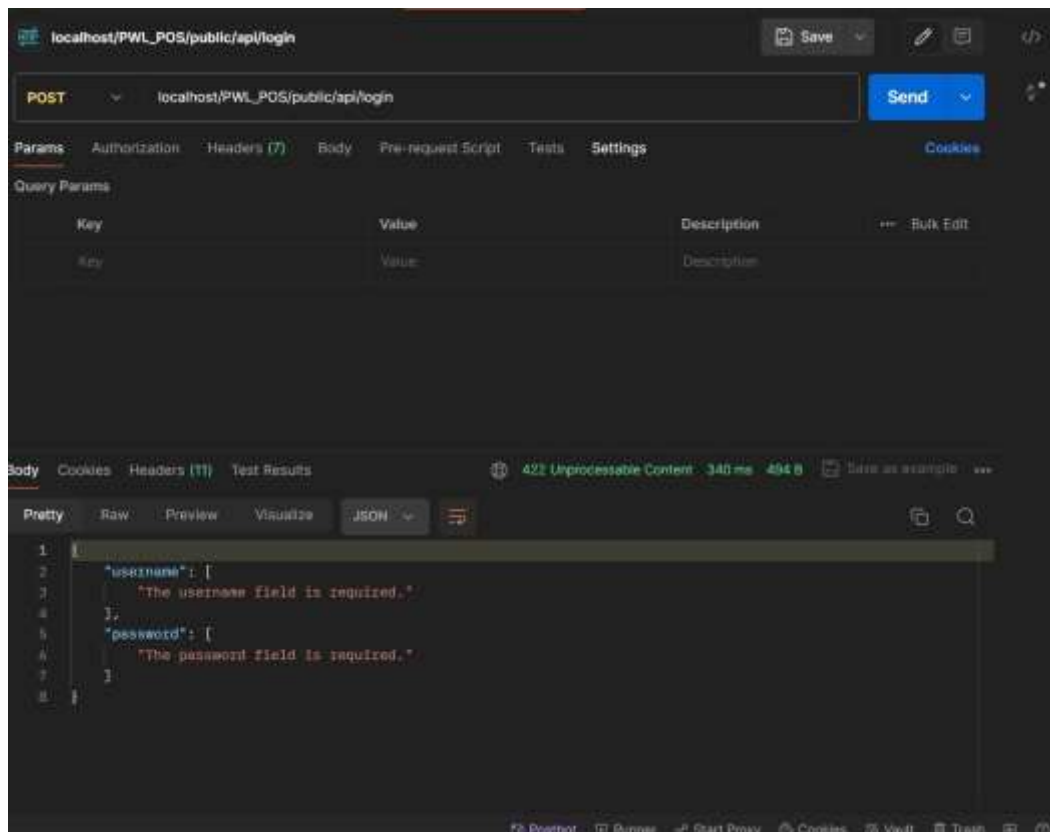
```
use App\Http\Controllers\Api\LoginController;  
  
Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');  
Route::post('/login', App\Http\Controllers\Api\LoginController::class)->name('login');  
Route::middleware('auth:api')->get('/user', function (Request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



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 localhost/PWL_POS/public/api/login Send

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

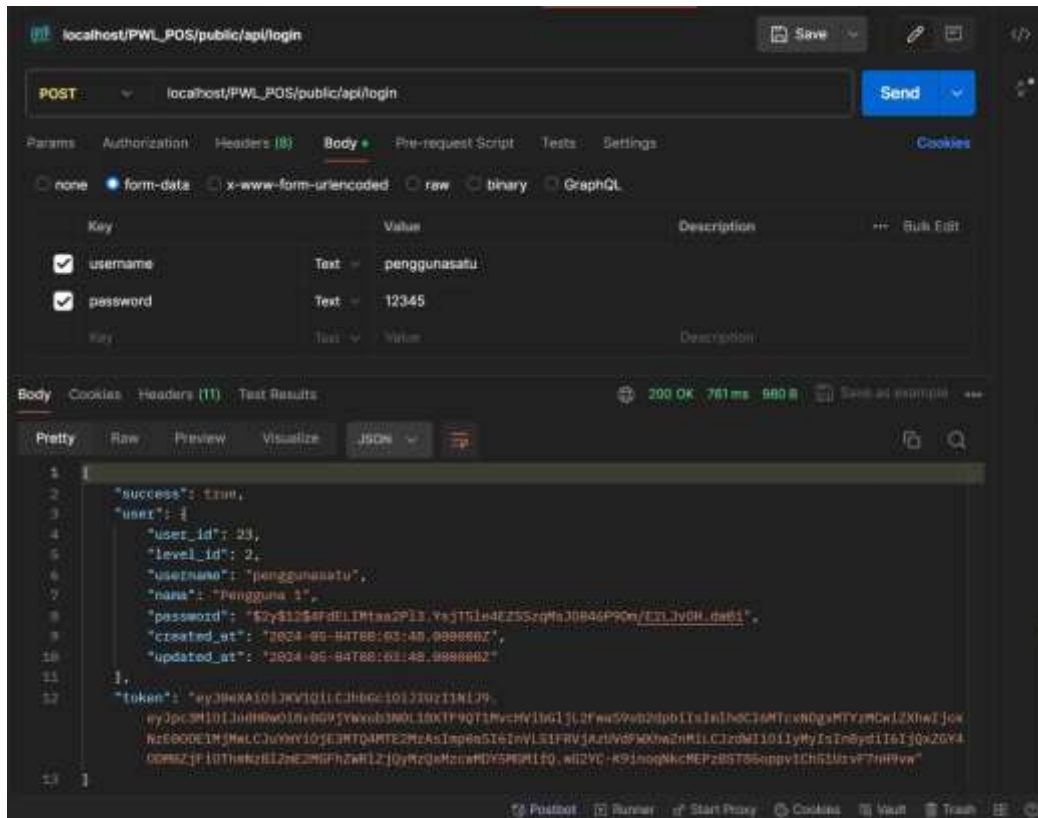
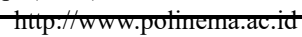
none **form-data** x-www-form-urlencoded raw binary

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

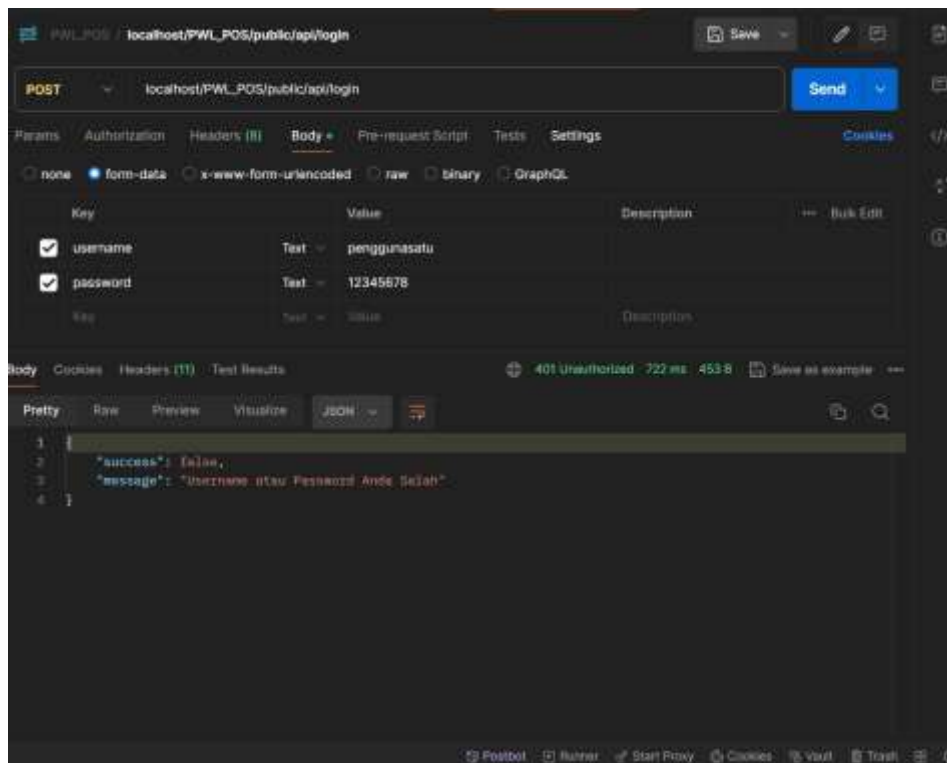
body Cookies Headers (11) Test Results Status: 200 OK Time: 1501 ms Size: 986 B Save Response

Pretty Raw Preview Visualize JSON

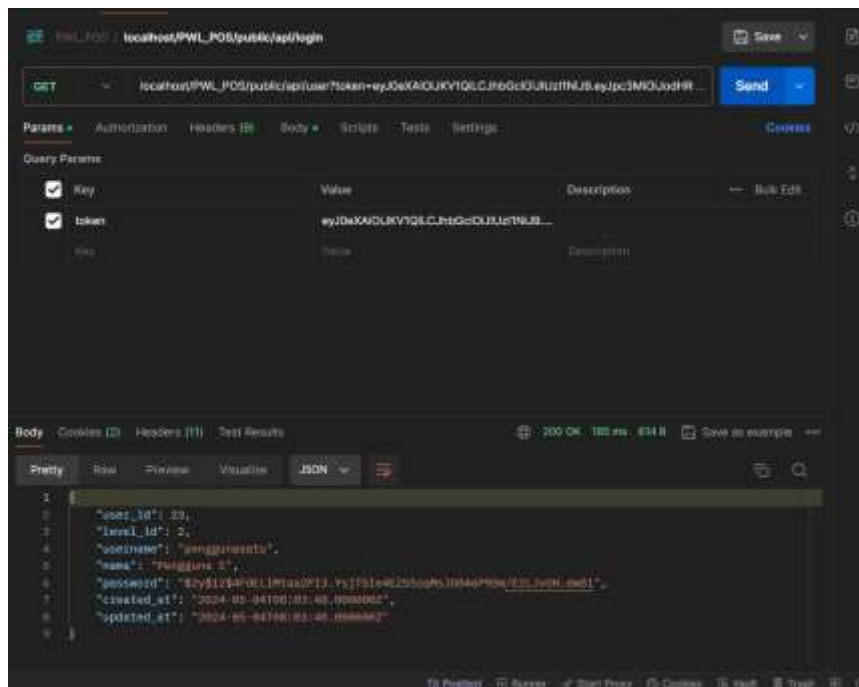
```
1 2 "success": true,
3  "user": {
4    "user_id": 17,
5    "level_id": 2,
6    "username": "penggunasatu",
7    "nama": "Pengguna 1",
8    "password": "$2y$12$Eb2SzV1jsykINytYGtrHi0DVAKcK5p6EgnZnmbChkPicIu7S0QJJJu",
9    "created_at": "2024-04-22T15:56:04.000000Z",
10   "updated_at": "2024-04-22T15:56:04.000000Z"
11 },
12 "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0L1BXTF9QT1MtbnVpb19wdWJsaWVYb2xvZ2luIiwiaWF0Ij0i"
13
```



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>
```



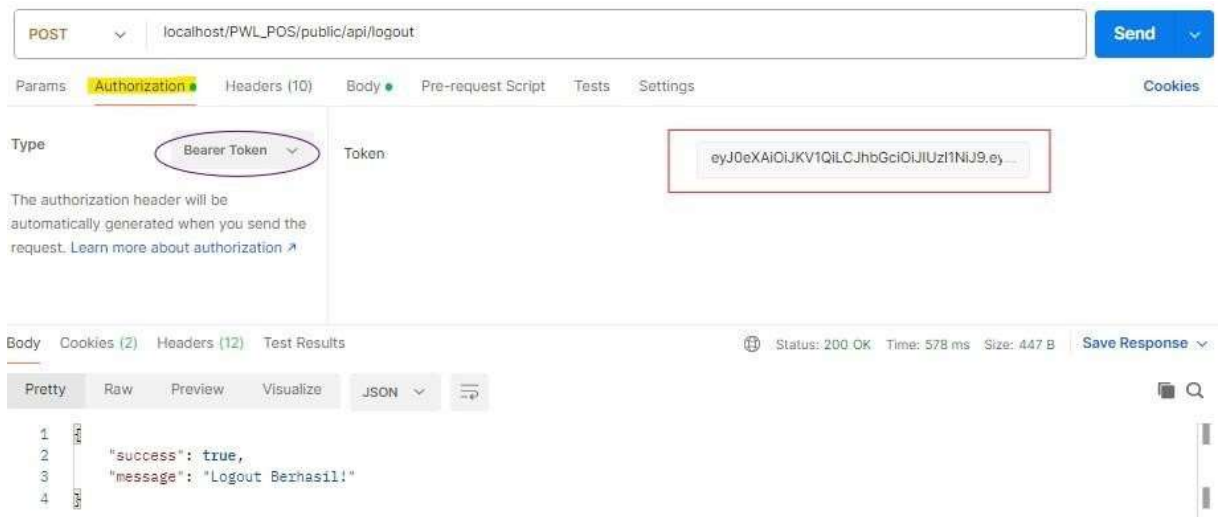
3. Buka file tersebut dan ubah kode menjadi seperti berikut.

```
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 }
```

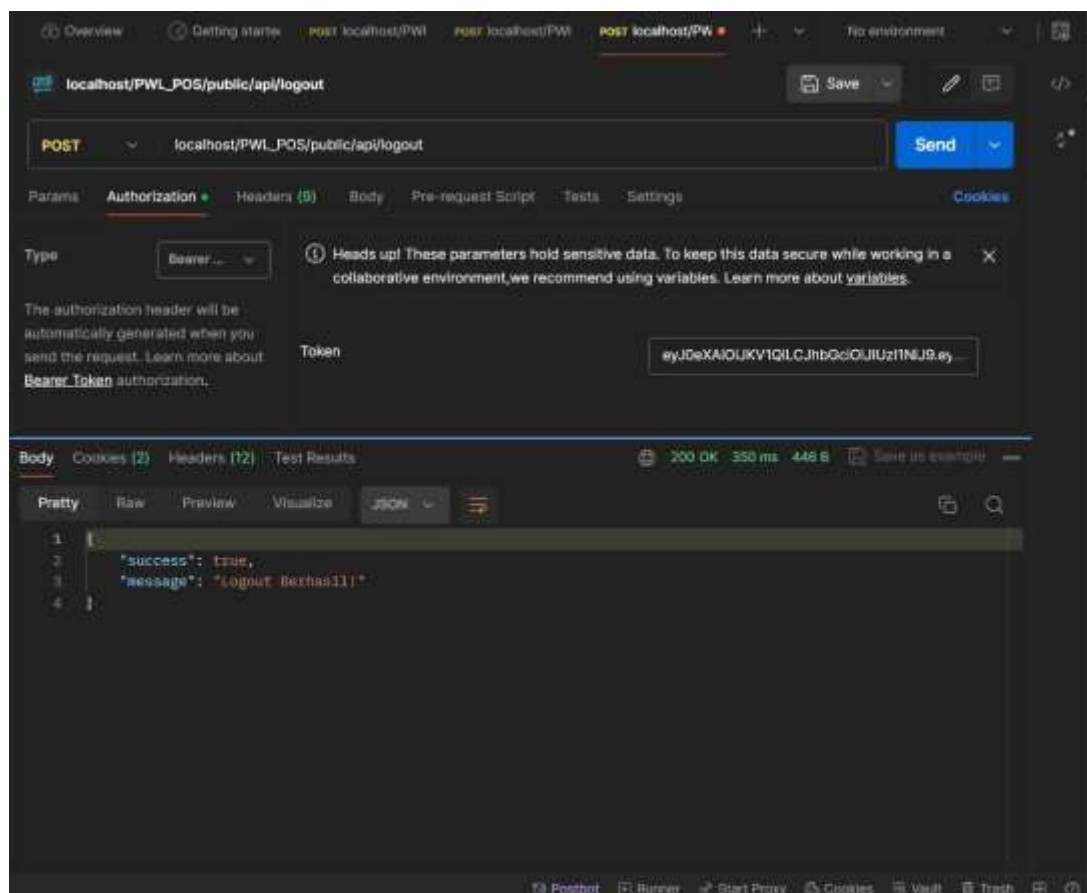
4. Lalu kita tambahkan routes pada api.php

```
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']);
```

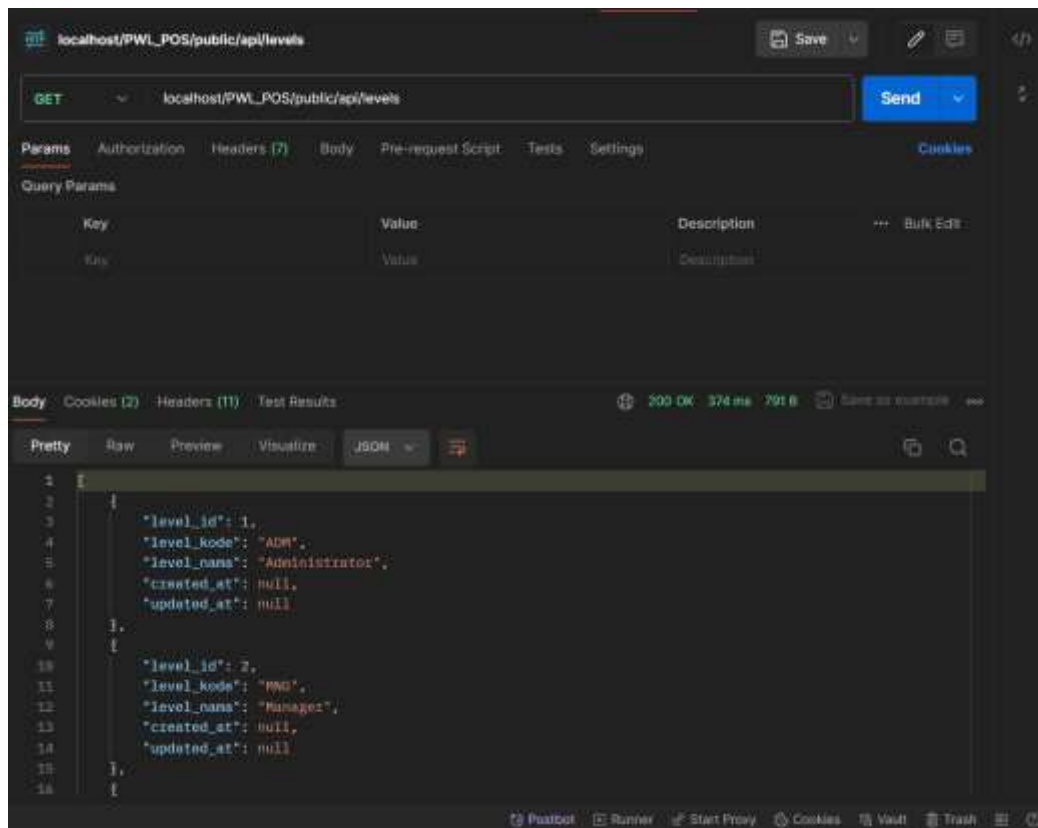
4. Jika sudah. Lakukan uji coba API mulai dari fungsi untuk menampilkan data. Gunakan URL: localhost/PWL_POS-main/public/api/levels dan method GET. **Jelaskan dan berikan screenshot hasil percobaan Anda.**



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI

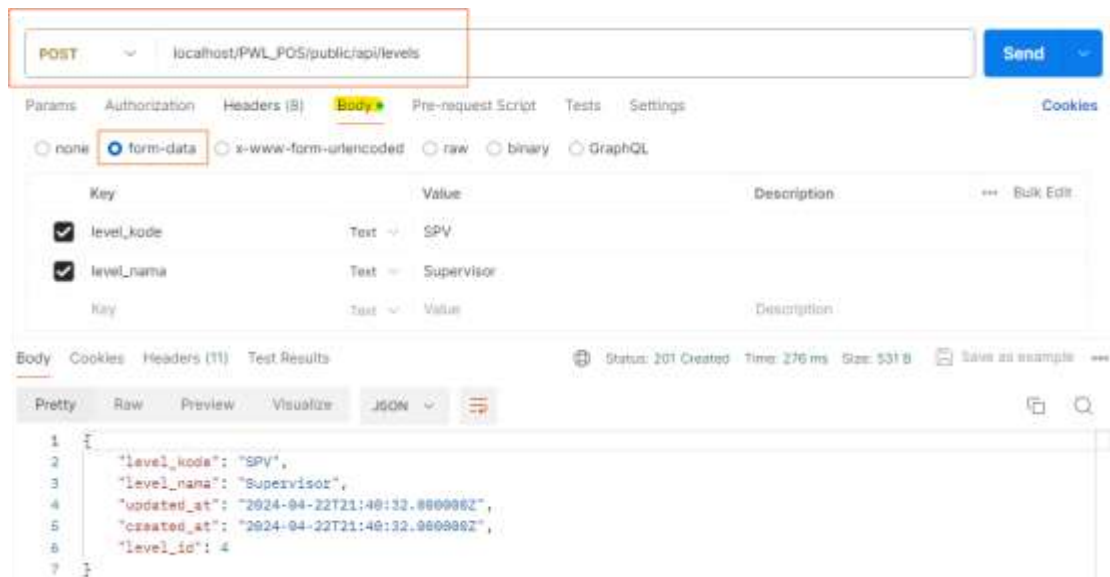
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420

<http://www.polinema.ac.id>

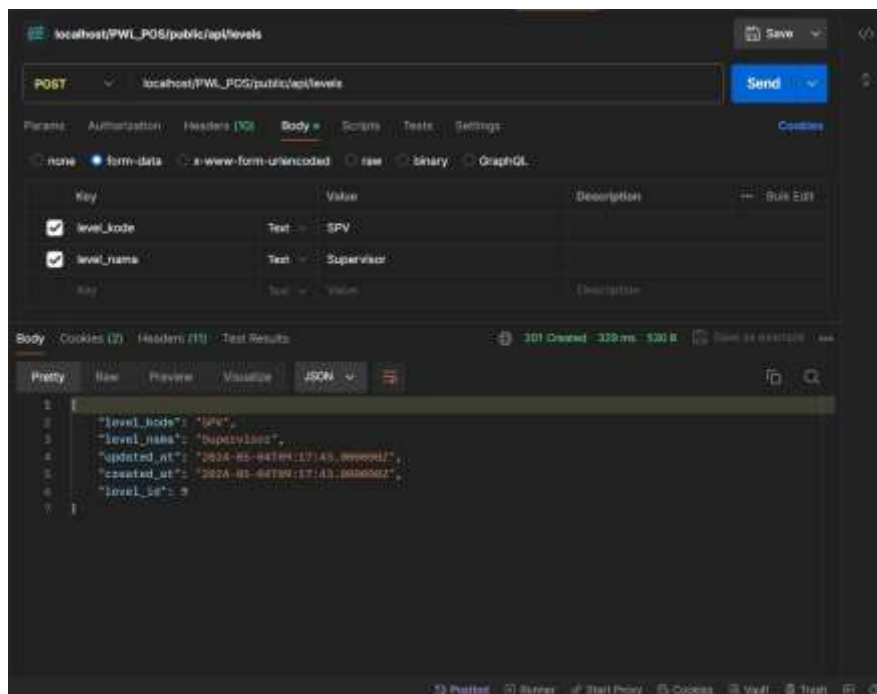


Percobaan diatas digunakan untuk menampilkan semua data level.

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



Jelaskan dan berikan screenshot hasil percobaan Anda.



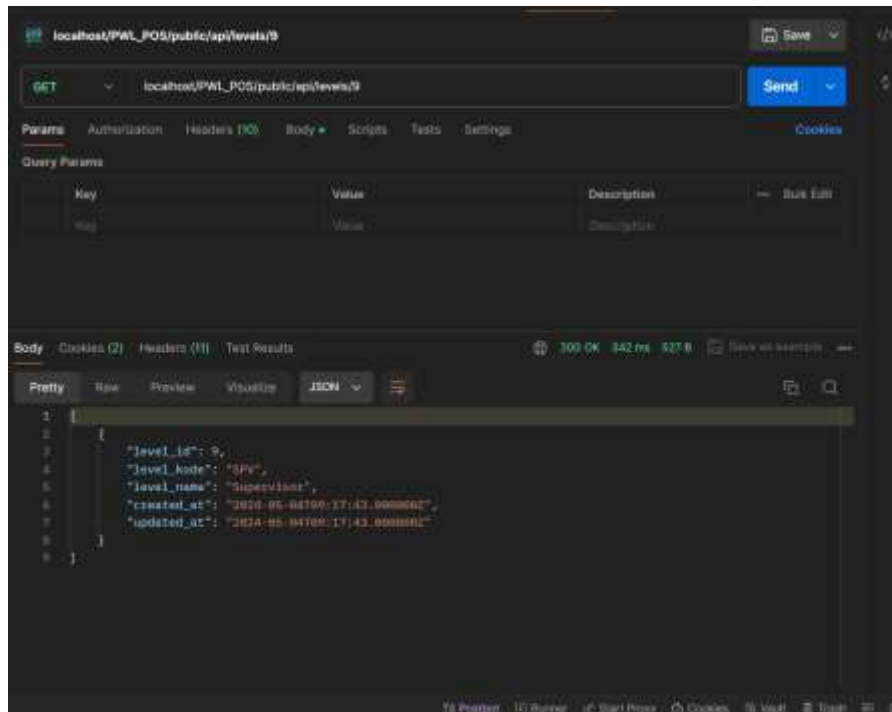
Percobaan diatas digunakan untuk menambahkan data pada tabel, pada percobaan diatas, saya menambahkan data dengan rincian :

- level_id yang bernilai 9
- level_kode yang berisi “SPV”
- level_nama yang berisi “Supervisor”



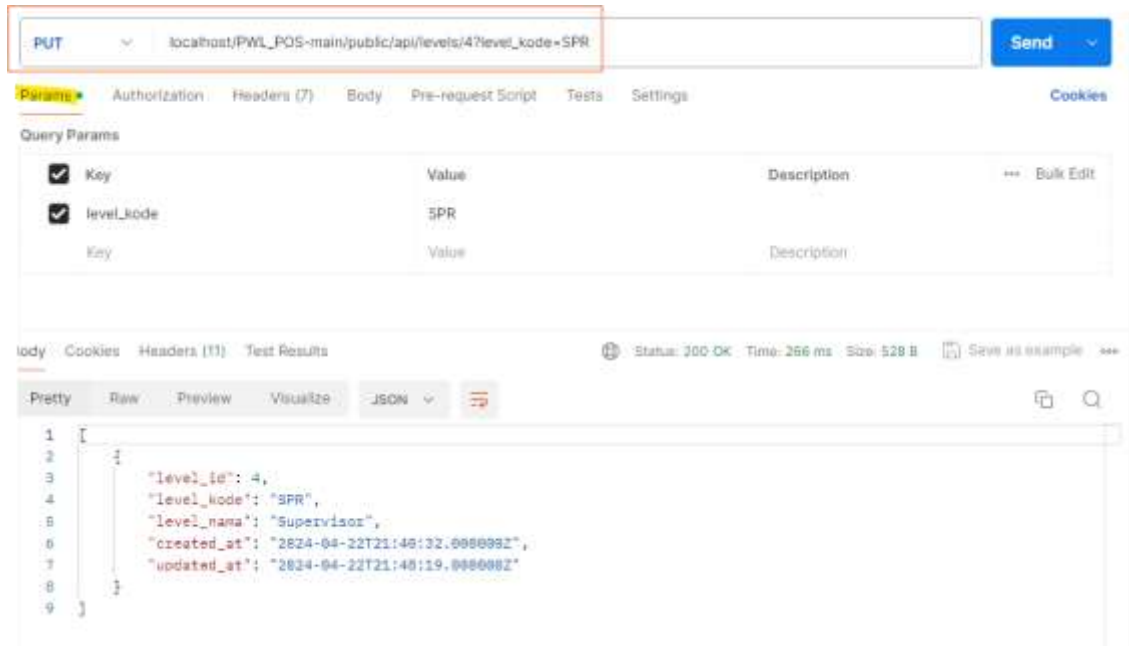
Untuk `updated_at` dan `created_at` akan terisi secara otomatis.

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

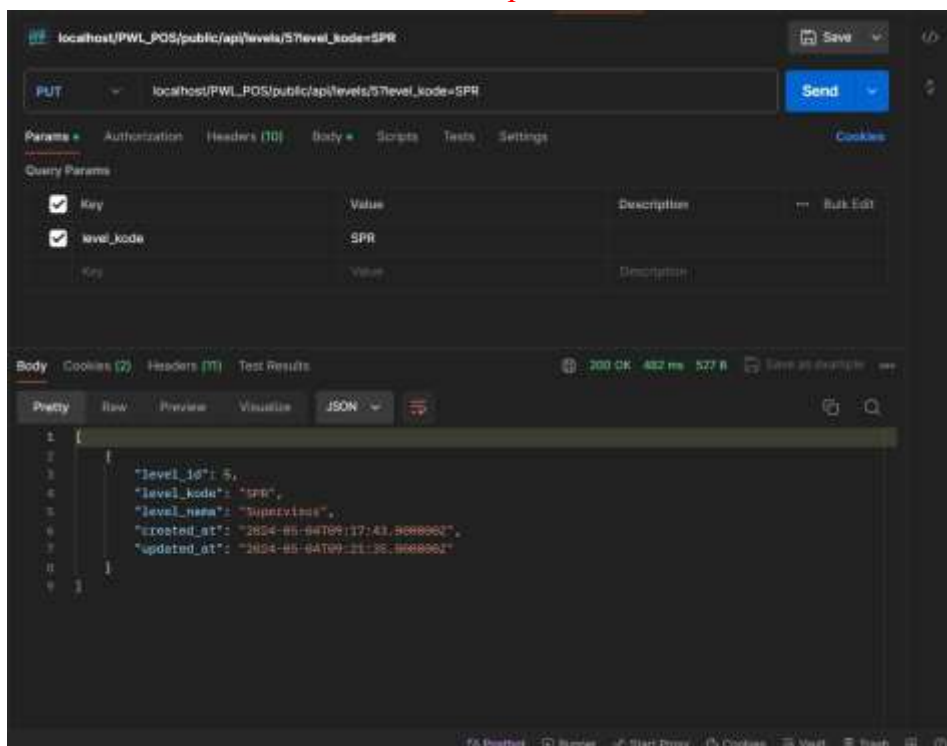


Percobaan diatas digunakan untuk menampilkan data dari id tertentu, pada percobaan diatas, saya menampilkan data dari `level_id` yang bernilai 9.

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.



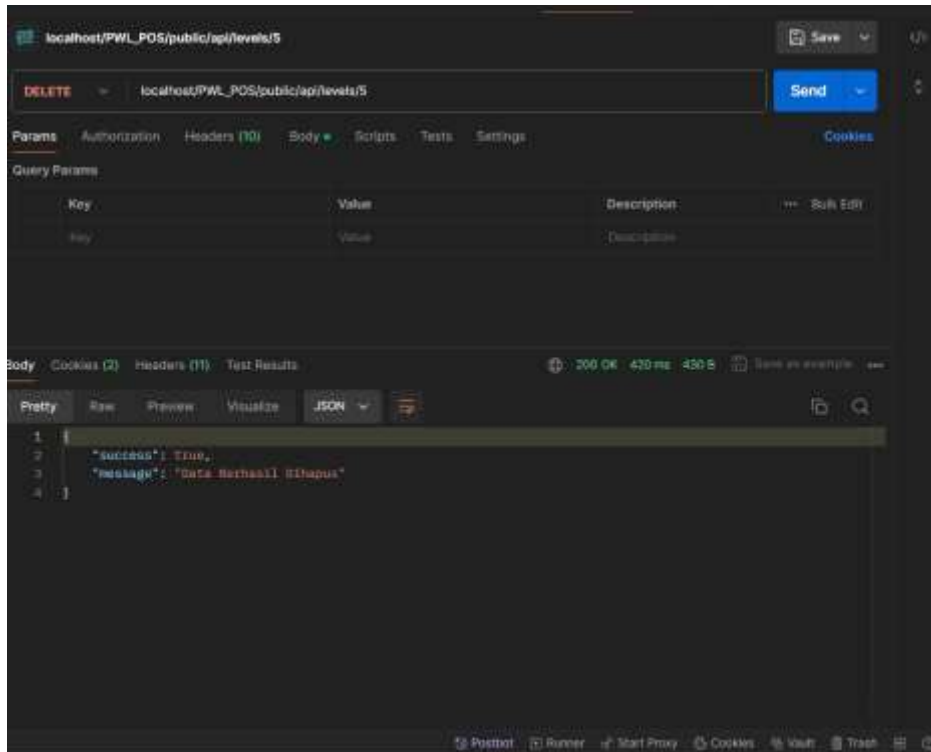
Jelaskan dan berikan screenshoot hasil percobaan Anda.



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 screenshot 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



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>

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
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     "name": "Administrator",
7     "password": "$y81217y81gS.TgWFOVKaM42ctedInGPruch7Hqeshb2SA0FsuqM10Gek1",
8     "created_at": null,
9     "updated_at": null
10  },
11  {
12    "user_id": 2,
13    "level_id": 2,
14    "username": "managet",
15    "name": "Manager",
16    "password": "$7y81217/cred1111azp8e870M1a3h9V_Cak1K30VCT7xM3SyuaXXbG2ke",
17    "created_at": null,
18    "updated_at": null
19  }
20 ]
```

PWL_POS / localhost/PWL_POS/public/api/users

DELETE localhost/PWL_POS/public/api/users/25

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 }
```



2. CRUD API tabel m_kategori

localhost/PWL_POS/public/api/kategoris

GET localhost/PWL_POS/public/api/kategoris

Send

Params Authorization Headers (0) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (11) Test Results

200 OK 3.42 s 321 B

Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "kategori_id": 1,
4     "kategori_kode": "KT0001",
5     "kategori_nama": "Elektronik",
6     "created_at": null,
7     "updated_at": null
8   },
9   {
10    "kategori_id": 2,
11    "kategori_kode": "KT0002",
12    "kategori_nama": "Pakaian",
13    "created_at": null,
14    "updated_at": null
15  },
16 }
```

localhost/PWL_POS/public/api/kategoris

POST localhost/PWL_POS/public/api/kategoris

Send

Params Authorization Headers (0) Body Body + Scripts Tests Settings Cookies

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

Key	Value	Description
<input checked="" type="checkbox"/> kategori_kode	Text KT0006	
<input checked="" type="checkbox"/> kategori_nama	Text Lainnya	
Key	Value	Description

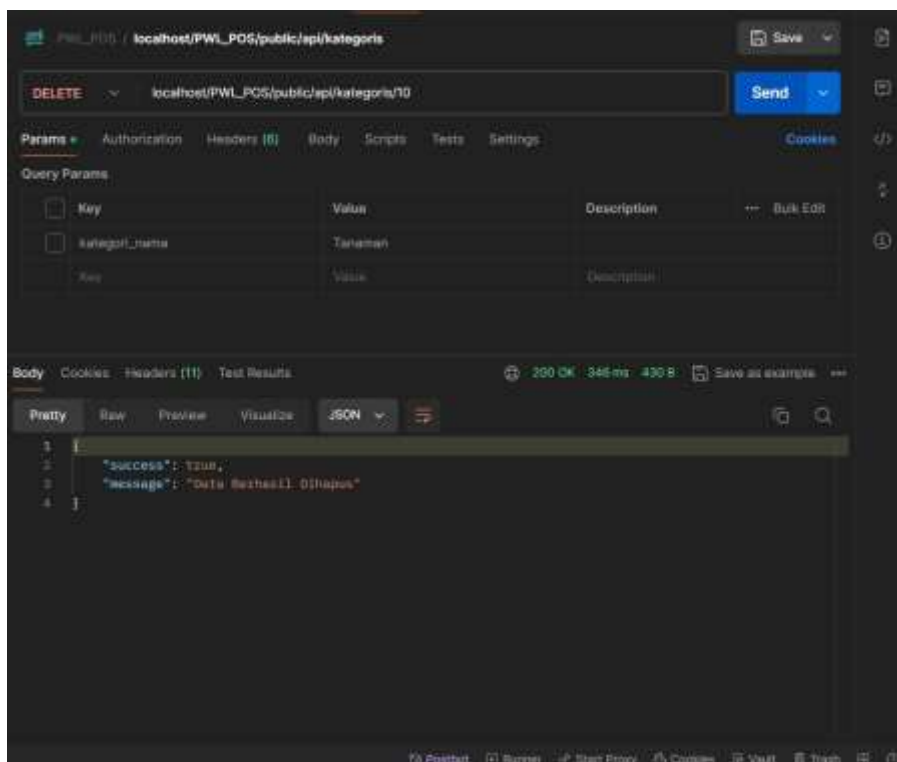
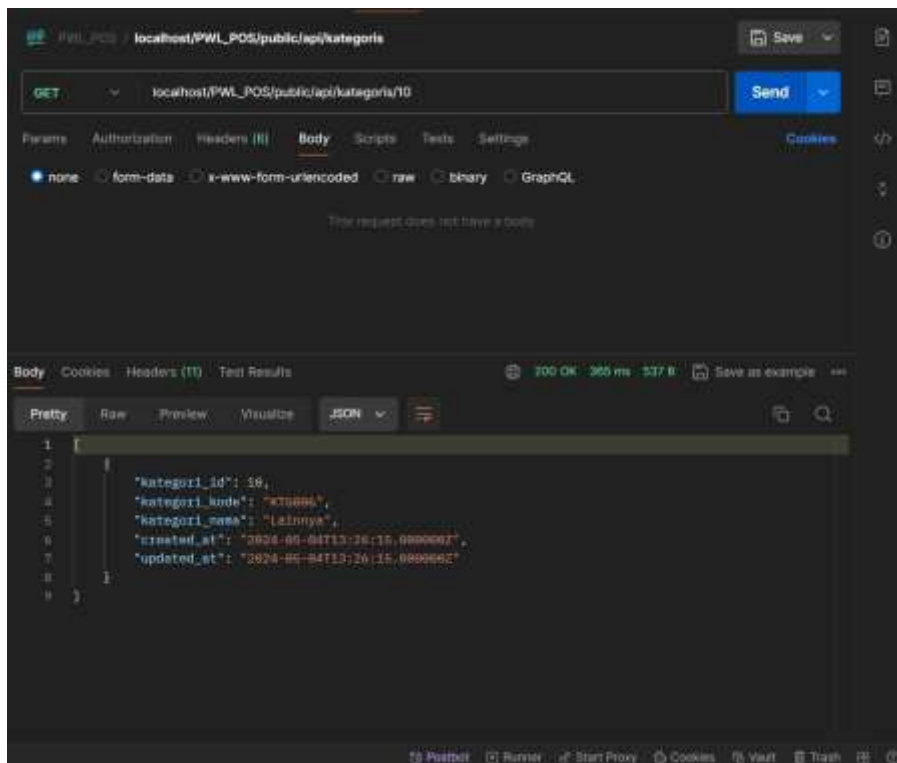
Body Cookies Headers (11) Test Results

201 Created 548 ms 540 B

Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   "kategori_kode": "KT0006",
3   "kategori_nama": "Lainnya",
4   "updated_at": "2024-05-04T13:26:15.000000Z",
5   "created_at": "2024-05-04T13:26:15.000000Z",
6   "kategori_id": 10
7 }
```



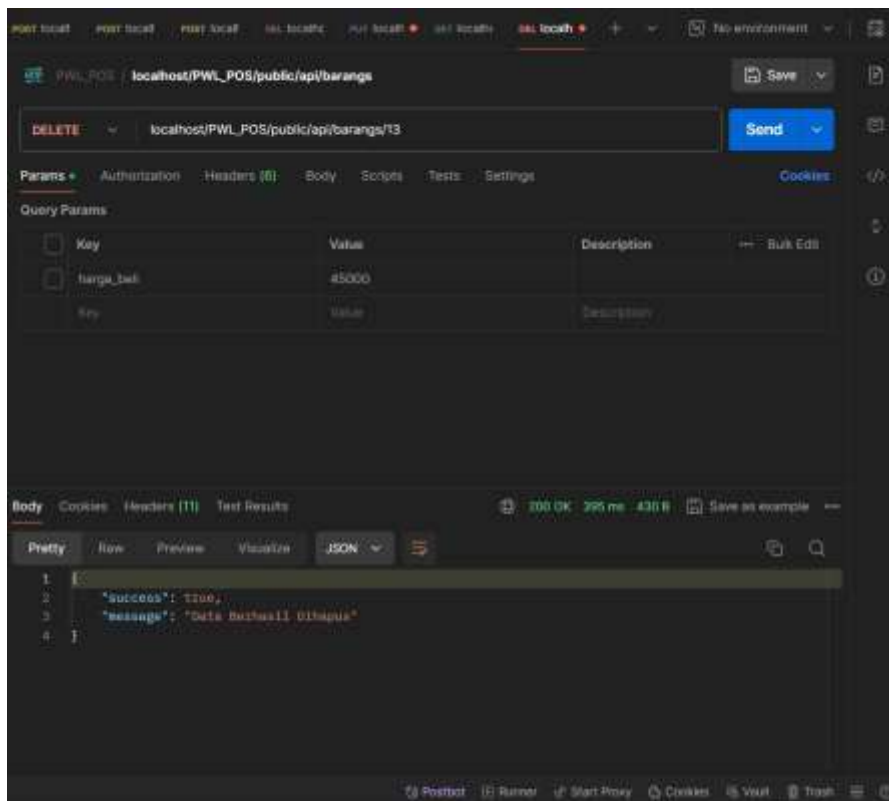
3. CRUD API tabel m_barang

The first screenshot shows a POST request to `localhost/PWL_POS/public/api/barangs` with a JSON body containing two items. The response is a 200 OK status with a 338 ms response time and a 2.14 KB body.

```
1 {
2   {
3     "barang_id": 1,
4     "barang_kode": "HP0001",
5     "kategori_id": 1,
6     "barang_nama": "Smartphone",
7     "harga_beli": 1000000,
8     "harga_jual": 1500000,
9     "created_at": null,
10    "updated_at": null
11  },
12  {
13    "barang_id": 2,
14    "barang_kode": "LAP0002",
15    "kategori_id": 1,
16    "barang_nama": "Laptop",
17    "harga_beli": 3000000,
18    "harga_jual": 4000000,
19    "created_at": null,
20    "updated_at": "2024-04-17T06:18:11.000000Z"
21  }
22 }
```

The second screenshot shows a GET request to `localhost/PWL_POS/public/api/barangs/1`. The response is a 200 OK status with a 255 ms response time and a 541 B body.

```
1 {
2   {
3     "barang_id": 1,
4     "barang_kode": "HP0001",
5     "kategori_id": 1,
6     "barang_nama": "Smartphone",
7     "harga_beli": 1000000,
8     "harga_jual": 1500000,
9     "created_at": null,
10    "updated_at": null
11  }
12 }
```



*** Sekian, dan selamat belajar ***