



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

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.

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
PS C:\laragon\www\P1> cd ".\Minggu 10\"
PS C:\laragon\www\P1\Minggu 10> cd ".\PWL_POS\"
PS C:\laragon\www\P1\Minggu 10\PWL_POS> composer require tymon/jwt-auth:2.1.1

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0
./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
- Downloading stella-maris/clock (0.1.7)
- Downloading lcobucci/clock (2.3.0)
- Downloading lcobucci/jwt (4.0.4)
- Downloading tymon/jwt-auth (2.1.1)
3/4 [=====] 75% Failed to download tymon/jwt-auth from dist: curl error 28 while downloading https://api.github.com/repos/tymondesigns/jwt-auth/zipb
all/51620ebd5b68bb3ce9e66ba86bda303ae5f10f7f: Operation timed out after 10014 milliseconds with 0 out of 0 bytes received
Now trying to download from source
- Syncing tymon/jwt-auth (2.1.1) into cache
```

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL
INFO Discovering packages.

barryvdh/laravel-dompdf ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE
tymon/jwt-auth ..... DONE
yajra/laravel-datatables-buttons ..... DONE
yajra/laravel-datatables-editor ..... DONE
yajra/laravel-datatables-fractal ..... DONE
yajra/laravel-datatables-html ..... DONE
yajra/laravel-datatables-oracle ..... DONE

93 packages you are using are looking for funding.
Use the "composer fund" command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0
INFO No publishable resources for tag [laravel-assets].

Found 1 security vulnerability advisory affecting 1 package.
Run "composer audit" for a full list of advisories.
PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

3. Setelah berhasil menginstall JWT, lanjutkan dengan publish konfigurasi file dengan perintah berikut: `php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"`



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

```
PS C:\laragon\www\P1\Minggu 10\PWL_POS> php artisan vendor:publish --provider "Tymon\JWTAuth\Providers\LaravelServiceProvider"
Warning: Module "pdo_mysql" is already loaded in Unknown on line 0
[INFO] Publishing assets.
Copying file [C:\laragon\www\P1\Minggu 10\PWL_POS\vendor\tymon\jwt-auth\config\config.php] to [C:\laragon\www\P1\Minggu 10\PWL_POS\config\jwt.php] DONE
PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

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\P1\Minggu 10\PWL_POS> php artisan jwt:secret
Warning: Module "pdo_mysql" is already loaded in Unknown on line 0
jwt-auth secret [kfzLX6GDCQgCZNmJ7RJk9DWAYqMQDK1jB8jAon6vBExZ0tQEY9cNihcaqkZh6q6U] set successfully.
PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

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

```
37
38     'guards' => [
39         'web' => [
40             'driver' => 'session',
41             'provider' => 'users',
42         ],
43         'api' => [
44             'driver' => 'jwt',
45             'provider' => 'users',
46         ],
47     ],
48
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

```
158 references | 0 implementations | Windsurf: Refactor | Explain
29 class UserModel extends Authenticatable implements JWTSubject
30
31 { //jobsheet 10 praktikum 1
    0 references | 0 overrides | Windsurf: Refactor | Explain | X
32 public function getJWTIdentifier(): mixed
33 {
34     return $this->getKey();
35 }
36
    0 references | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
37 public function getJWTCustomClaims(): array
38 {
39     return [];
40 }
```

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.

```
php artisan make:controller Api/RegisterController

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0

[INFO] Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api\RegisterController.php] created successfully.

PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > RegisterController.php > PHP Intelephense > RegisterController > __invoke
0 references | 0 implementations | Windsurf: Refactor | Explain
class RegisterController extends Controller
{
    0 references | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
    public function __invoke(Request $request): JsonResponse|mixed{
        //set validation
        $validator = Validator::make($request->all(), [
            'username' => 'required',
            'nama' => 'required',
            'password' => 'required|min:5|confirmed',
            'level_id' => 'required',
        ]);

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

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

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

        //return response json user is created
        if($user){
            return response()->json([
                'success' => true,
                'user' => $user,
            ], 201);
        }

        //return response json fail
        return response()->json([
            'success' => false,
        ], 409);
    }
}
```

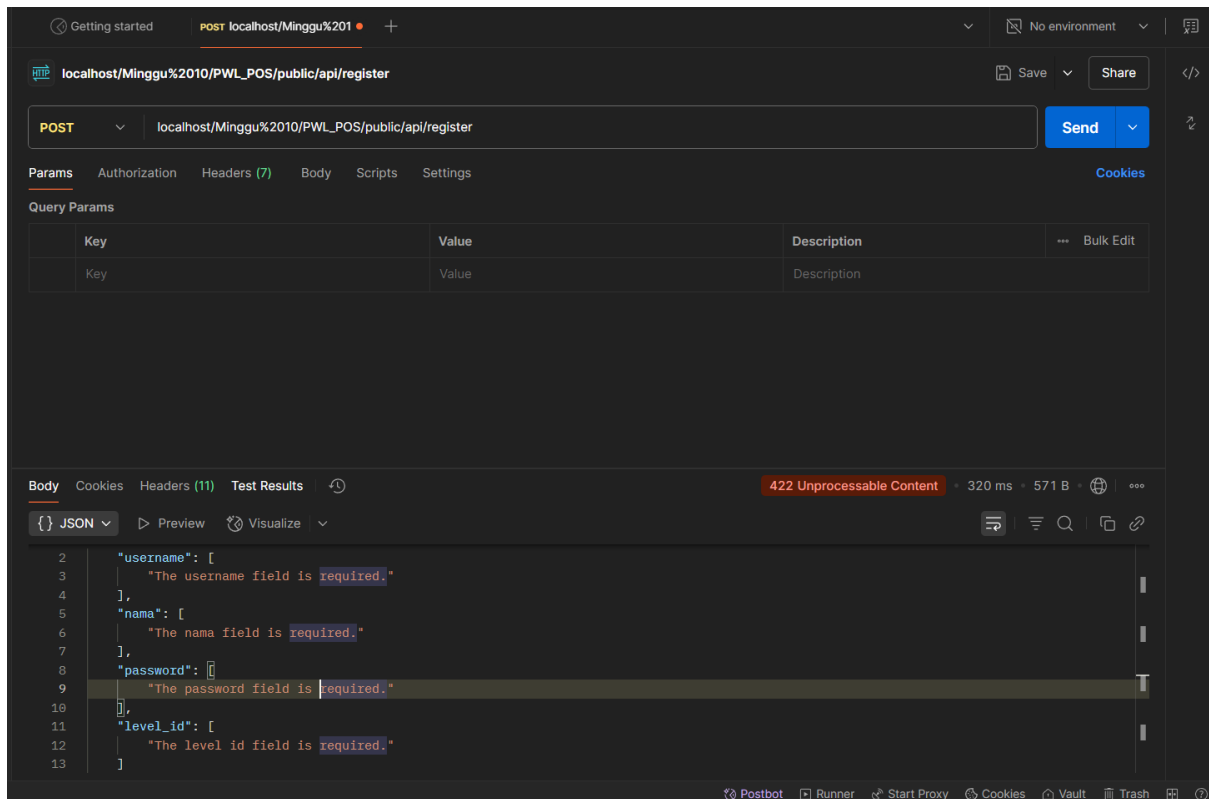
10. Selanjutnya buka routes/api.php, ubah semua kode menjadi seperti berikut

```
auth.php M  UserModel.php M  RegisterController.php U  api.php M X  web.php
Minggu 10 > PWL_POS > 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  use FontLib\Table\Type\name;
7
8  /*
9  |-----
10 | API Routes
11 |-----
12 |
13 | Here is where you can register API routes for your application. These
14 | routes are loaded by the RouteServiceProvider and all of them will
15 | be assigned to the "api" middleware group. Make something great!
16 |
17 */
18
19 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');
20
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

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.



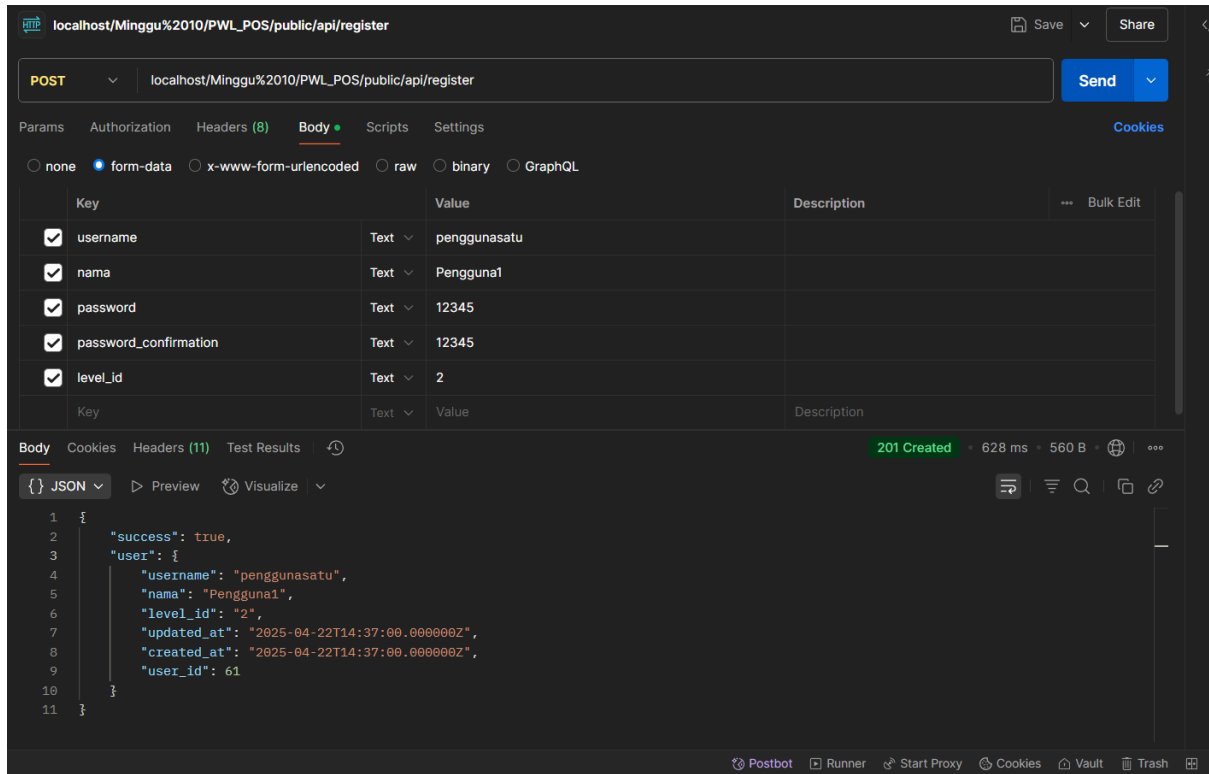
Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.

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.



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1



Lakukan percobaan yang sama dan berikan screenshoot hasil percobaan Anda.

13. Lakukan commit perubahan file pada Github.

Sudah



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

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\P1\Minggu 10\PWL_POS> php artisan make:controller Api/LoginController

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0

INFO Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api>LoginController.php] created successfully.

PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

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

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > LoginController.php > PHP Intelephense > LoginController > __invoke

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
10 0 references | 0 implementations | Windsurf: Refactor | Explain
11 class LoginController extends Controller
12 {
13     0 references | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
14     public function __invoke(Request $request): JsonResponse|mixed
15     {
16         //set validation
17         $validator = Validator::make(request()->all(), [
18             'username' => 'required',
19             'password' => 'required'
20         ]);
21
22         //if validation fails
23         if ($validator->fails()) {
24             return response()->json([
25                 'errors' => $validator->errors()
26             ], 422);
27         }
28
29         //ger credentials from request
30         $credentials = $request->only(['username', 'password']);
31
32         //if auth failed
33         if (!$token = auth()->attempt(credentials: $credentials)) {
34             return response()->json([
35                 'success' => false,
36                 'message' => 'Username atau password Anda salah'
37             ], 401);
38         }
39
40         //if auth success
41         return response()->json([
42             'success' => true,
43             'user' => auth()->guard('api')->user(),
44             'token' => $token,
45         ], 200);
46     }
47 }
```



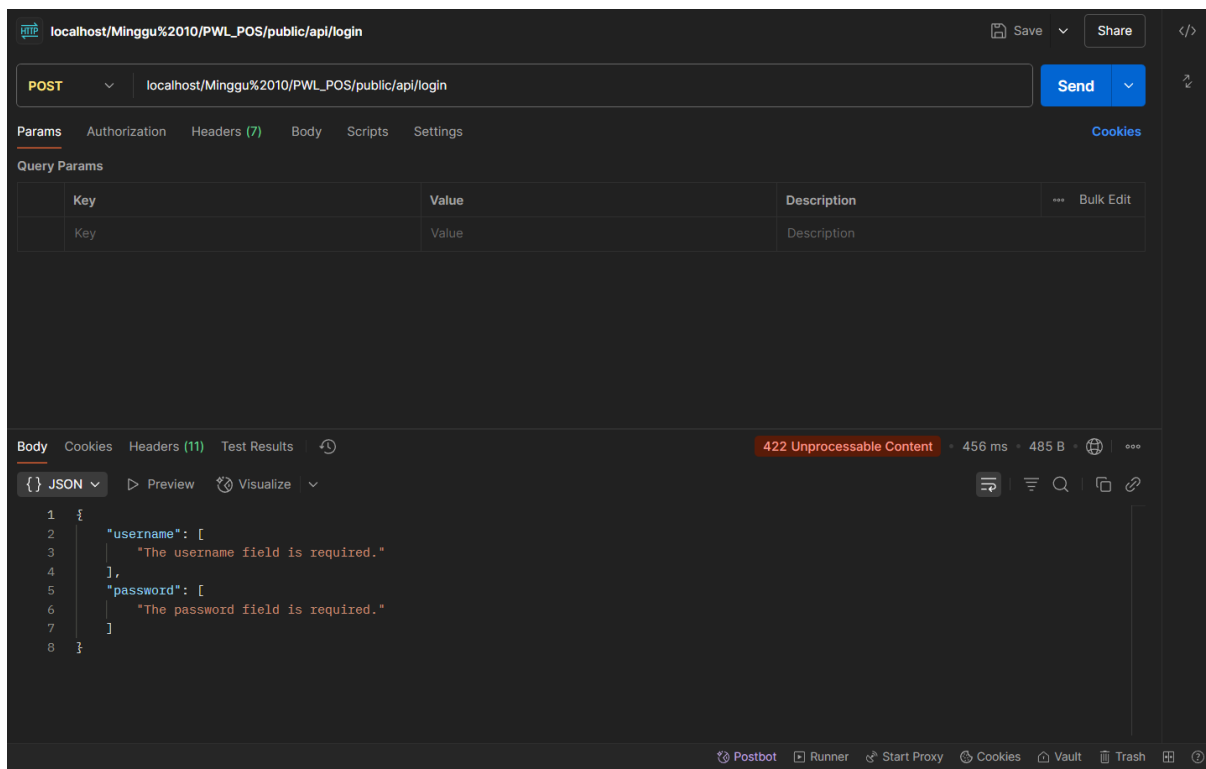
NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

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

```
17  
18  
19 Route::post('/register', App\Http\Controllers\Api\RegisterController::class)->name('register');  
20 Route::post('/login', App\Http\Controllers\Api>LoginController::class)->name('login');  
21 Route::middleware('auth:api')->get('/user', function (Request $request): mixed {  
22     return $request->user();  
23 });
```

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



localhost/Minggu%2010/PWL_POS/public/api/login

Save

Share

POST

localhost/Minggu%2010/PWL_POS/public/api/login

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	admin		
<input checked="" type="checkbox"/> password	12347		
Key	Value	Description	

Body

Cookies

Headers (11)

Test Results

401 Unauthorized

499 ms

444 B

🔍

⋮

{}

JSON

Preview

Visualize

🔍

📄

🔗

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

Postbot

Runner

Start Proxy

Cookies

Vault

Trash



The screenshot shows the Postman application interface. At the top, the URL bar displays 'localhost/Minggu%2010/PWL_POS/public/api/user'. The method is set to 'GET'. Below the URL bar, there are tabs for 'Params', 'Authorization', 'Headers (8)', 'Body', 'Scripts', and 'Settings'. The 'Authorization' tab is selected, showing 'Auth Type' as 'Bearer Token'. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about variables.' The 'Token' field contains the value 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9'. Below this, a text box explains that the authorization header will be automatically generated. The bottom section shows the 'Body' tab with a JSON response:

```
{  "user_id": 61,  "level_id": 2,  "username": "penggunasatu",  "foto": null,  "nama": "Pengguna1",  "created_at": "2025-04-22T14:37:00.000000Z",  "updated_at": "2025-04-22T14:37:00.000000Z"}
```

 The status bar at the bottom indicates a '200 OK' response with a time of 381 ms and a size of 562 B. The bottom navigation bar includes icons for Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and a menu icon.

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYWxob3N0L01pbmdndSUyMDEwL1BXTF9QT1MvcHVibGljL2FwaS9sb2dpbiIsImhhdCI6MTc0NTMzNjIxMCwiZXhwIjoxNzQ1MzU0DEwLCJuYmYiOiJlE3NDUzMzYyMTAsImp0aSI6IjhCTDYxR0t1MmxhZnpscE0iLCJzdWliOiI2MSIsInBydiI6IjQxZGY4ODM0ZjFiOThmNzBlZmE2MGFhZWRLZjQyMzQxMzcwMDY5MGMIifQ.y-EtNx1jyYyS0yf5vGLhO-vfgLu7E05xE7zEMqx_k-I

Sudah



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Praktikum 3 – Membuat RESTful API Logout

1. Tambahkan kode berikut pada file .env

JWT_SHOW_BLACKLIST_EXCEPTION=true

```
62 JWT_SHOW_BLACKLIST_EXCEPTION=true
```

2. Buat Controller baru dengan nama LogoutController.

php artisan make:controller Api/LogoutController

```
PS C:\laragon\www\P1\Minggu 10\PWL_POS> php artisan make:controller Api/LogoutController
Warning: Module "pdo_mysql" is already loaded in Unknown on line 0
[INFO] Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api\LogoutController.php] created successfully.
PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

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

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > LogoutController.php > ...
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Http\Request;
7 use Tymon\JWTAuth\Contracts\Providers\JWT;
8 use Tymon\JWTAuth\Exceptions\JWTException;
9 use Tymon\JWTAuth\Exceptions\TokenExpiredException;
10 use Tymon\JWTAuth\Exceptions\TokenInvalidException;
11 use Tymon\JWTAuth\Facades\JWTAuth;
12
13
14 class LogoutController extends Controller
15 {
16     public function __invoke(Request $request): JsonResponse|mixed{
17         //remove token
18         $removeToken = JWTAuth::invalidate(JWTAuth::getToken());
19
20         if ($removeToken) {
21             // return response json
22             return response()->json([
23                 'success' => true,
24                 'message' => 'Logout berhasil'
25             ]);
26         }
27     }
28 }
29
```



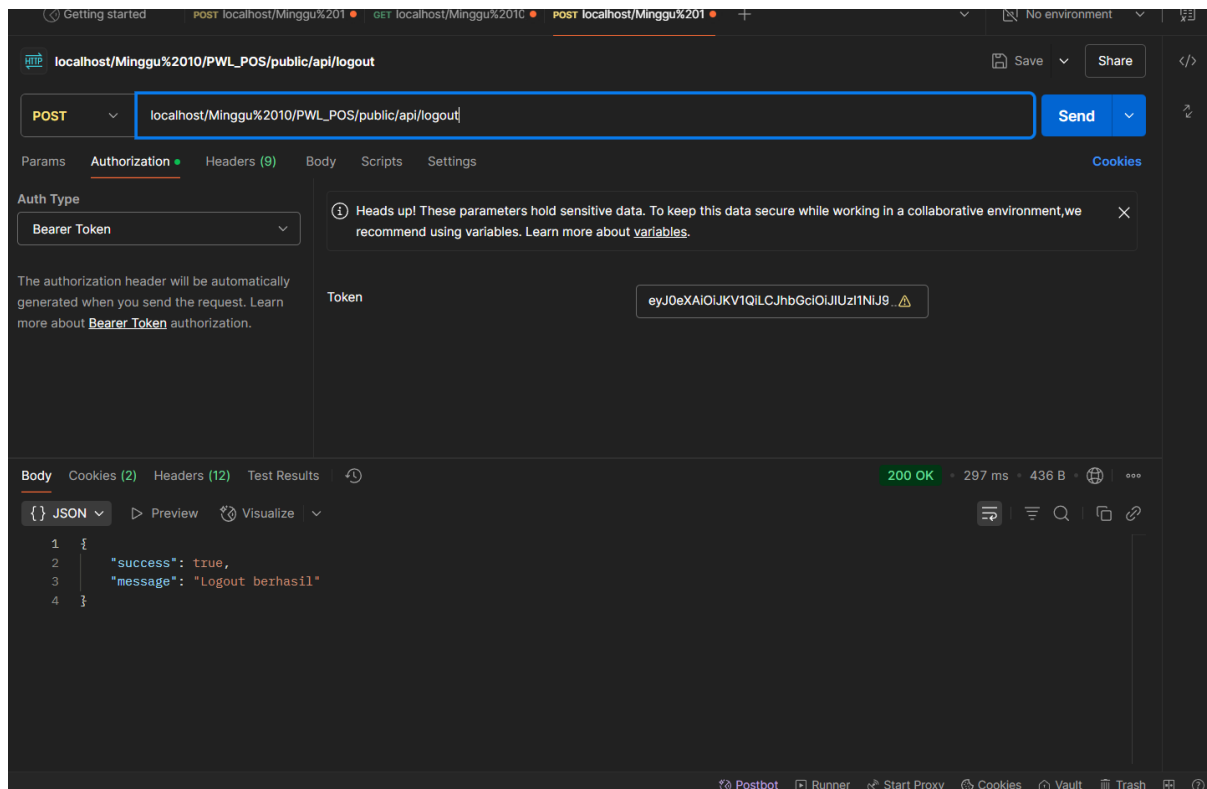
NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

4. Lalu kita tambahkan routes pada api.php

```
21 | 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.

Sudah



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

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\P1\Minggu 10\PWL_POS> php artisan make:controller Api/LevelController

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0

[INFO] Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api\LevelController.php] created successfully.

PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```

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

```
LoginController.php | LevelController.php | api.php | .env | LogoutController.php
Minggu 10 > PWL_POS > app > Http > Controllers > Api > LevelController.php > PHP Intelephense > LevelController > destroy

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



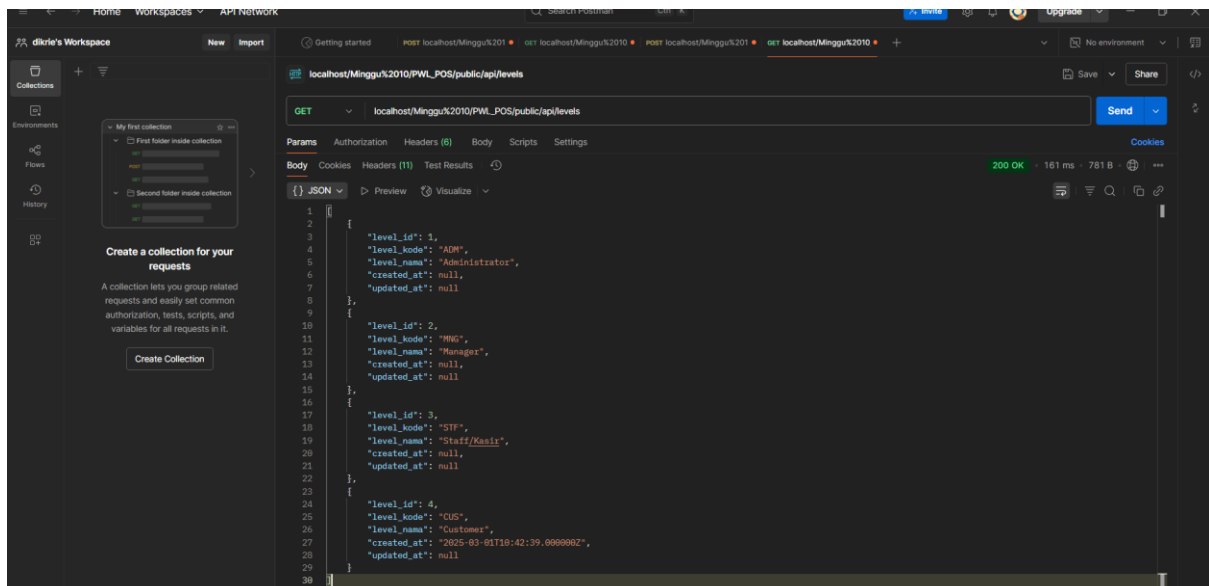
NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

3. Kemudian kita lengkapi routes pada api.php.

```
24  
25 Route::get('levels', [LevelController::class, 'index']);  
26 Route::post('levels', [LevelController::class, 'store']);  
27 Route::get('levels/{level}', [LevelController::class, 'show']);  
28 Route::put('levels/{level}', [LevelController::class, 'update']);  
29 Route::delete('levels/{level}', [LevelController::class, 'destroy']);  
30
```

4. Jika sudah. Lakukan uji coba API mulai dari fungsi untuk menampilkan data. Gunakan

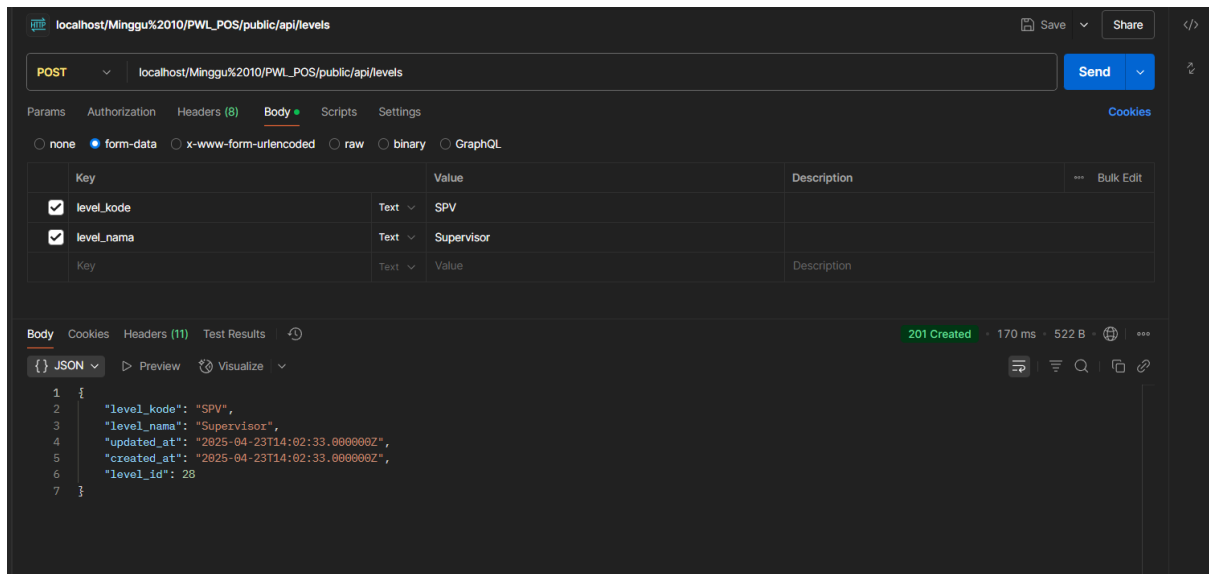
URL: localhost/PWL_POS-main/public/api/levels dan method GET. Jelaskan dan berikan screenshoot hasil percobaan Anda.



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



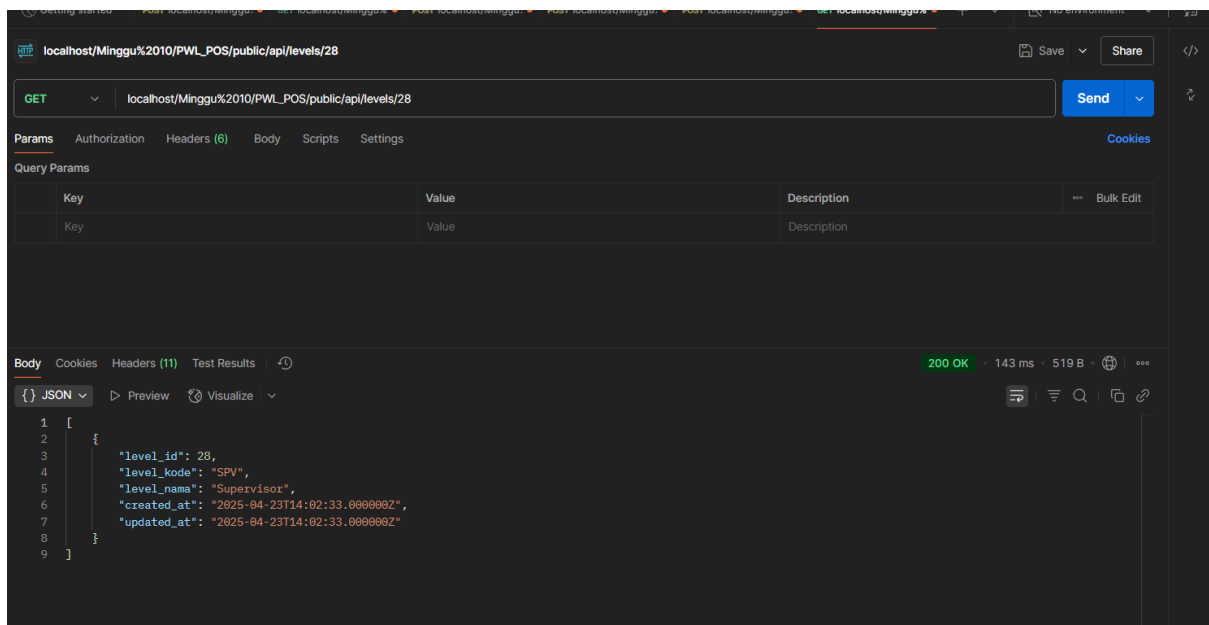
NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1



Membuat level baru dengan kode SPV dan nama Supervisor

Jelaskan dan berikan screenshoot hasil percobaan Anda.

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



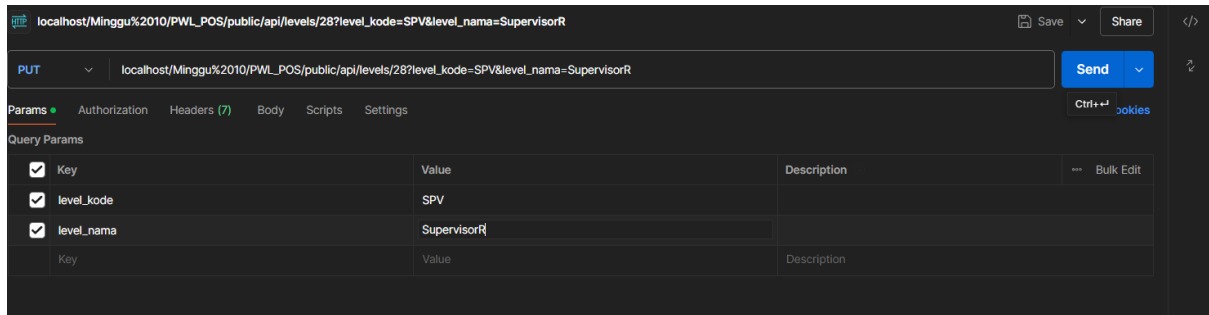
Menampilkan level dengan level id 28 yang tadi sudah dibuat



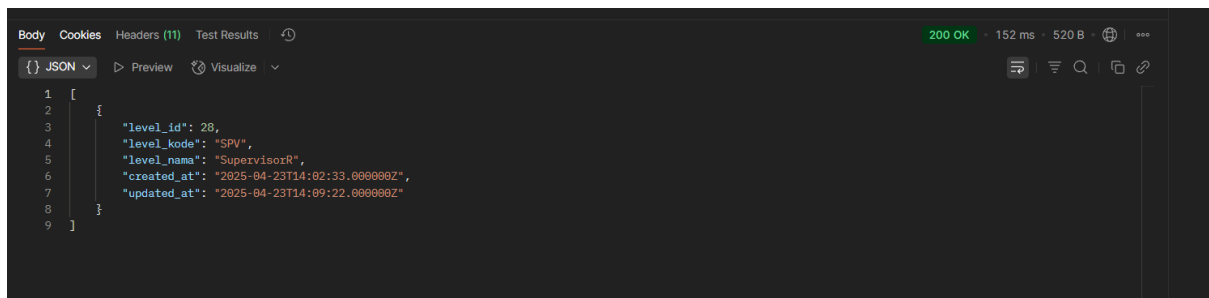
NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

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



Mengedit level nama menjadi SupervisorR

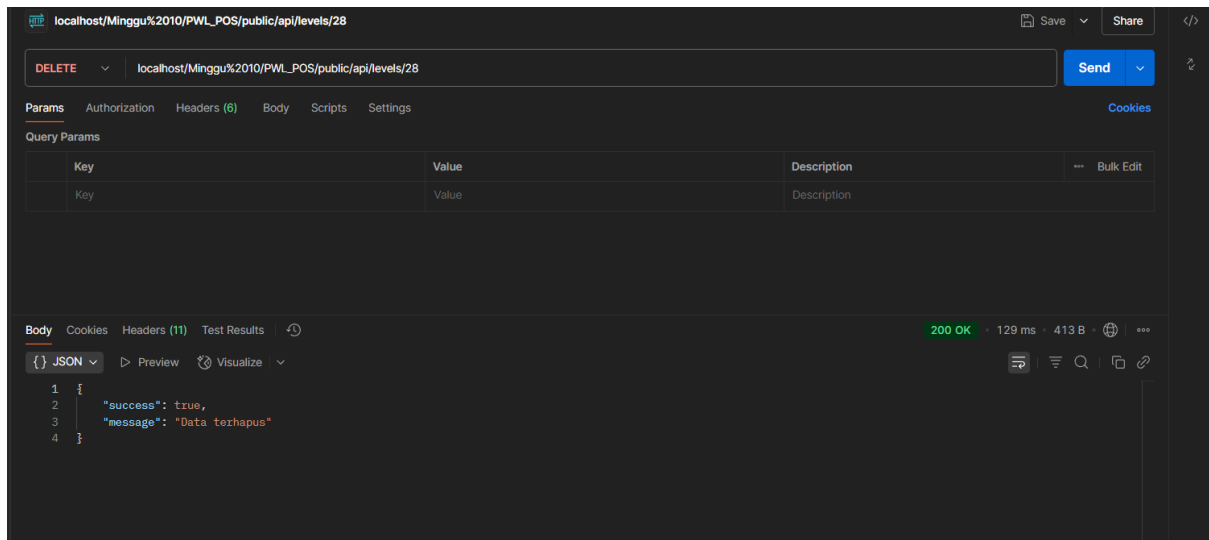


Level_nama telah berhasil diubah

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1



Berhasil menghapus Level dengan id 28

9. Lakukan commit perubahan file pada Github.

Sudah

TUGAS

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

- **M_user**

Membuat file UserController

```
PS C:\laragon\www\P1\Minggu 10\PWL_POS> php artisan make:controller Api/UserController

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0

INFO Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api\UserController.php] created successfully.

PS C:\laragon\www\P1\Minggu 10\PWL_POS>
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menambahkan function crud UserController

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > UserController.php > PHP Intelephense > UserController

2
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\UserModel;
7 use Illuminate\Http\Request;
8
9
10 6 references | 0 implementations | Windsurf: Refactor | Explain
11 class UserController extends Controller
12 {
13     //
14     1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
15     public function index(): Collection
16     {
17         return UserModel::all();
18     }
19
20     1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
21     public function store(Request $request): JsonResponse|mixed
22     {
23         $user = UserModel::create($request->all());
24         return response()->json($user, 201);
25     }
26
27     1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
28     public function show(UserModel $user): array|Collection|Model|UserModel|null
29     {
30         return UserModel::find($user);
31     }
32
33     1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
34     public function update(Request $request, UserModel $user): array|Collection|Model|UserModel|null
35     {
36         $user->update($request->all());
37         return UserModel::find($user);
38     }
39
40     1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
41     public function destroy(UserModel $user): JsonResponse|mixed{
42         $user->delete();
43         return response()->json([
44             'success' => true,
45             'message' => 'Data terhapus'
46         ]);
47     }
48 }
```

Menambahkan route user

```
32
33 Route::get('users', [UserController::class, 'index']);
34 Route::post('users', [UserController::class, 'store']);
35 Route::get('users/{user}', [UserController::class, 'show']);
36 Route::put('users/{user}', [UserController::class, 'update']);
37 Route::delete('users/{user}', [UserController::class, 'destroy']);
38
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menampilkan data user

GET http://localhost/Minggu%2010/PWL_POS/public/api/users

200 OK · 143 ms · 2.2 KB

```
{
  "user_id": 1,
  "level_id": 1,
  "username": "admin",
  "foto": "profile_1744478144_hiu.jpeg",
  "nama": "Administrator",
  "created_at": null,
  "updated_at": "2025-04-22T14:28:01.000000Z"
},
{
  "user_id": 2,
  "level_id": 2,
  "username": "manager",
  "foto": "profile_1744546225_img-0989.jpg",
  "nama": "Manager",
  "created_at": null,
  "updated_at": "2025-04-14T01:37:24.000000Z"
},
{
  "user_id": 3,
  "level_id": 3,
  "username": "staff",
  "foto": "profile_1744546163_img-1025.jpg",
  "nama": "Staff/Kasir",
  "created_at": null,
  "updated_at": "2025-04-14T01:37:32.000000Z"
},
{
  "user_id": 8,
  "level_id": 4,
  "username": "customer1",
  "created_at": null,
  "updated_at": null
}
```

Menambahkan data user baru

POST http://localhost/Minggu%2010/PWL_POS/public/api/users

201 Created · 391 ms · 532 B

Body

```
{
  "level_id": "1",
  "username": "dikaPost",
  "nama": "Dika Post",
  "updated_at": "2025-04-24T06:56:09.000000Z",
  "created_at": "2025-04-24T06:56:09.000000Z",
  "user_id": 62
}
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menampilkan data user dengan id

localhost/Minggu%2010/PWL_POS/public/api/users/62

GET localhost/Minggu%2010/PWL_POS/public/api/users/62 Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (11) Test Results 200 OK · 159 ms · 706 B

JSON Preview Visualize

```
11 {
12   "user_id": 62,
13   "level_id": 1,
14   "username": "dikaPost",
15   "foto": null,
16   "nama": "Dika Post",
17   "created_at": "2025-04-24T06:56:09.000000Z",
18   "updated_at": "2025-04-24T06:56:09.000000Z"
19 }
20 }
```

Mengedit nama user yang tadi dibuat

localhost/Minggu%2010/PWL_POS/public/api/users/62?level_id=2&username=postDika&nama=Dika Arie&password=11111

PUT localhost/Minggu%2010/PWL_POS/public/api/users/62?level_id=2&username=postDika&nama=Dika Arie&password=11111 Send

Params Authorization Headers (7) Body Scripts Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
level_id	2			
username	postDika			
nama	Dika Arie			
password	11111			

Body Cookies Headers (11) Test Results 200 OK · 362 ms · 706 B

JSON Preview Visualize

```
11 {
12   "user_id": 62,
13   "level_id": 2,
14   "username": "postDika",
15   "foto": null,
16   "nama": "Dika Arie",
17   "created_at": "2025-04-24T06:56:09.000000Z",
18   "updated_at": "2025-04-24T06:59:25.000000Z"
19 }
20 }
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menghapus data user dengan id

localhost/Minggu%2010/PWL_POS/public/api/users/62

DELETE localhost/Minggu%2010/PWL_POS/public/api/users/62 Send

Params Authorization Headers (6) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (11) Test Results 200 OK • 169 ms • 413 B

JSON Preview Visualize

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



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

- **M_Kategori**

Membuat file KategoriController

```
PS C:\laragon\www\P1\Minggu 10\PWL_POS> php artisan make:controller Api/KategoriController

Warning: Module "pdo_mysql" is already loaded in Unknown on line 0

INFO Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api\KategoriController.php] created successfully.

PS C:\laragon\www\P1\Minggu 10\PWL_POS> |
```

Menambahkan function crud di dalam KategoriController

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > KategoriController.php > PHP > KategoriController

1 <?php
2 namespace App\Http\Controllers\Api;
3 use App\Http\Controllers\Controller;
4 use App\Models\KategoriModel;
5 use Illuminate\Http\Request;
6
7
8 6 references | 0 implementations | Windsurf: Refactor | Explain
9 class KategoriController extends Controller
10
11 1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
12 public function index(): Collection
13 {
14     return KategoriModel::all();
15 }
16
17 1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
18 public function store(Request $request): JsonResponse|mixed
19 {
20     $kategori = KategoriModel::create($request->all());
21     return response()->json($kategori, 201);
22 }
23
24 1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
25 public function show(KategoriModel $kategori): array|Collection|KategoriModel|Model|null
26 {
27     return KategoriModel::find($kategori);
28 }
29
30 1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
31 public function update(Request $request, KategoriModel $kategori): array|Collection|KategoriModel|Model|null
32 {
33     $kategori->update($request->all());
34     return KategoriModel::find($kategori);
35 }
36
37 1 reference | 0 overrides | Windsurf: Refactor | Explain | Generate Function Comment | X
38 public function destroy(KategoriModel $kategori): JsonResponse|mixed
39 {
40     $kategori->delete();
41     return response()->json([
42         'success' => true,
43         'message' => 'Data berhasil dihapus'
44     ]);
45 }
46
47 }
```

Menambahkan route kategori

```
31
32 Route::get('kategoris', [KategoriController::class, 'index']);
33 Route::post('kategoris', [KategoriController::class, 'store']);
34 Route::get('kategoris/{kategori}', [KategoriController::class, 'show']);
35 Route::put('kategoris/{kategori}', [KategoriController::class, 'update']);
36 Route::delete('kategoris/{kategori}', [KategoriController::class, 'destroy']);
37
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menampilkan data kategori

GET localhost/Minggu%2010/PWL_POS/public/api/kategoris

200 OK · 257 ms · 937 B

```
{
  "kategori_id": 1,
  "kategori_kode": "ELC001",
  "kategori_nama": "Elektronik",
  "created_at": null,
  "updated_at": null
},
{
  "kategori_id": 2,
  "kategori_kode": "FNB002",
  "kategori_nama": "Makanan & Minuman",
  "created_at": null,
  "updated_at": null
},
{
  "kategori_id": 3,
  "kategori_kode": "FRN003",
  "kategori_nama": "Furniture",
  "created_at": null,
  "updated_at": null
},
{
  "kategori_id": 4,
  "kategori_kode": "CLT004",
  "kategori_nama": "Pakaian",
  "created_at": null,
  "updated_at": null
},
{
  "kategori_id": 5,
  "kategori_kode": "OR005",
  "kategori_nama": "Olahraga",
  "created_at": null,
  "updated_at": null
}
```

Menambahkan data kategori baru

POST localhost/Minggu%2010/PWL_POS/public/api/kategoris

201 Created · 152 ms · 542 B

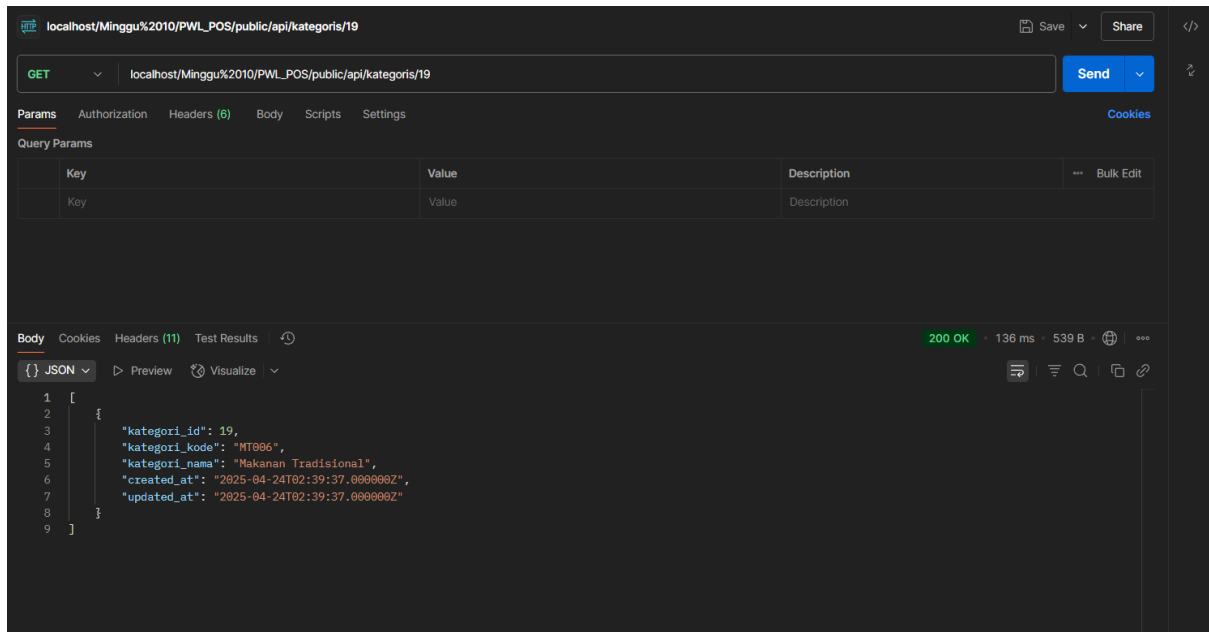
Key	Value	Description
kategori_kode	MT006	
kategori_nama	Makanan Tradisional	

```
{
  "kategori_kode": "MT006",
  "kategori_nama": "Makanan Tradisional",
  "updated_at": "2025-04-24T02:39:37.000000Z",
  "created_at": "2025-04-24T02:39:37.000000Z",
  "kategori_id": 19
}
```

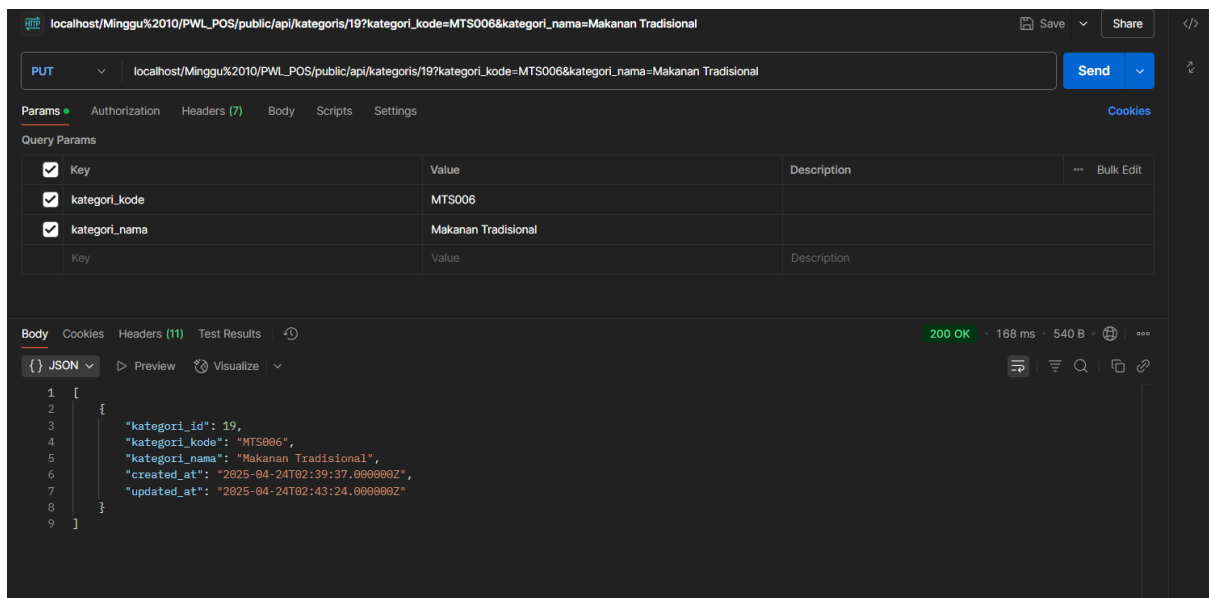


NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menampilkan detail kategori yang baru dibuat



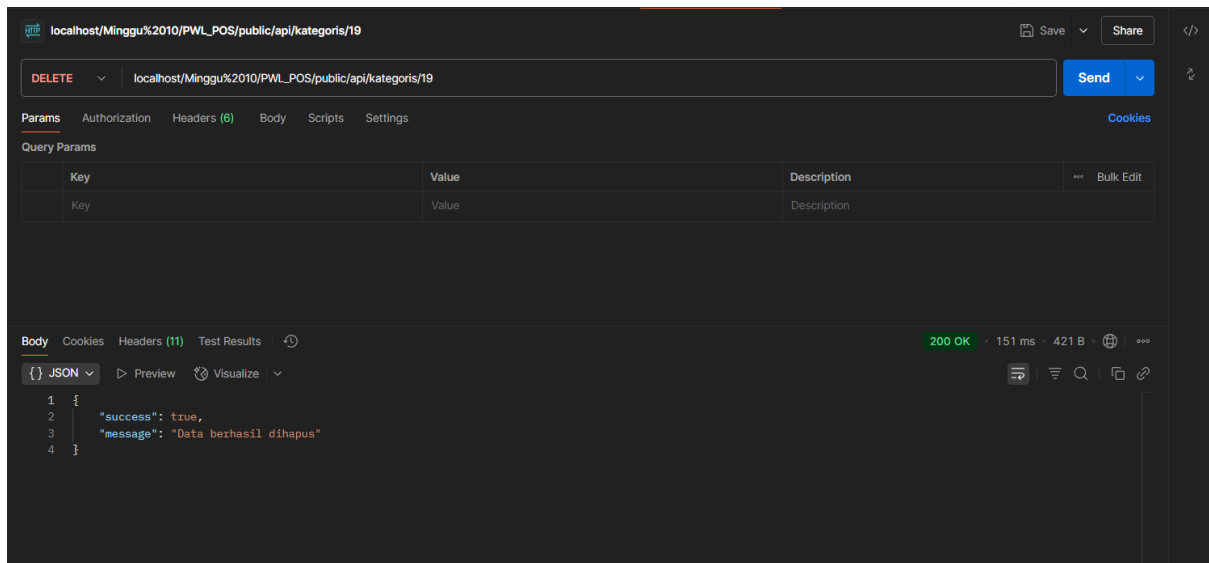
Mengedit kode_kategori yang tadi dibuat awalnya MT006 menjadi MTS006



Menghapus data kategori dengan id 19 yang tadi dibuat



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1



- M_barang

Membuat file BarangController

```
PS C:\laragon\www\P1\Minggu 10\PWL_POS> php artisan make:controller Api/BarangController
Warning: Module "pdo_mysql" is already loaded in Unknown on line 0
INFO Controller [C:\laragon\www\P1\Minggu 10\PWL_POS\app\Http\Controllers\Api\BarangController.php] created successfully.
PS C:\laragon\www\P1\Minggu 10\PWL_POS> 
```

Menambahkan function crud di BarangController



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

```
Minggu 10 > PWL_POS > app > Http > Controllers > Api > BarangController.php > PHP Intelephense > BarangController > destroy
1 <?php
2
3 namespace App\Http\Controllers\Api;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\BarangModel;
7 use Illuminate\Http\Request;
8
9 class BarangController extends Controller
10 {
11     //
12     public function index(): Collection
13     {
14         //
15         return BarangModel::all();
16     }
17     public function store(Request $request): JsonResponse|mixed
18     {
19         $barang = BarangModel::create($request->all());
20         return response()->json($barang, 201);
21     }
22     public function show(BarangModel $barang): array|BarangModel|Collection|Model|null
23     {
24         return BarangModel::find($barang);
25     }
26     public function update(Request $request, BarangModel $barang): JsonResponse|mixed
27     {
28         $barang->update($request->all());
29         return response()->json($barang, 200);
30     }
31     public function destroy(BarangModel $barang): JsonResponse|mixed
32     {
33         $barang->delete();
34         return response()->json([
35             'success' => true,
36             'message' => 'Data berhasil dihapus'
37         ]);
38     }
39 }
40
```

Menambahkan route barang

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

Menampilkan data barang



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

```
1 [
2   {
3     "barang_id": 1,
4     "kategori_id": 1,
5     "barang_kode": "TV001",
6     "barang_nama": "Televisi LED 32 inch",
7     "harga_beli": 2500000,
8     "harga_jual": 3000000,
9     "created_at": null,
10    "updated_at": null
11  },
12  {
13    "barang_id": 2,
14    "kategori_id": 1,
15    "barang_kode": "HP002",
16    "barang_nama": "Smartphone Android",
17    "harga_beli": 3500000,
18    "harga_jual": 4000000,
19    "created_at": null,
20    "updated_at": null
21  },
22  {
23    "barang_id": 3,
24    "kategori_id": 2,
25    "barang_kode": "SNK003",
26    "barang_nama": "Snack Coklat",
27    "harga_beli": 10000,
28    "harga_jual": 15000,
29    "created_at": null,
30    "updated_at": null
31  },
32  ]
```

Menambah data barang baru

```
1 {
2   "kategori_id": "1",
3   "barang_kode": "PS002",
4   "barang_nama": "Playstation 4",
5   "harga_beli": "3000000",
6   "harga_jual": "4000000",
7   "updated_at": "2025-04-24T07:08:02.000000Z",
8   "created_at": "2025-04-24T07:08:02.000000Z",
9   "barang_id": 51
10 }
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Mengedit data barang yang tadi dibuat

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

- URL:** `localhost/Minggu%2010/PWL_POS/public/api/barangs/51?kategori_id=1&barang_kode=PS004&barang_nama=Playstation 5&harga_beli=5000000&harga_jual=7000000`
- Method:** PUT
- Params:**

Key	Value	Description
kategori_id	1	
barang_kode	PS004	
barang_nama	Playstation 5	
harga_beli	5000000	
harga_jual	7000000	
- Body:** JSON

```
{  "barang_id": 51,  "kategori_id": "1",  "barang_kode": "PS004",  "barang_nama": "Playstation 5",  "harga_beli": "5000000",  "harga_jual": "7000000",  "created_at": "2025-04-24T07:08:02.000000Z",  "updated_at": "2025-04-24T07:17:32.000000Z"}
```
- Status:** 200 OK, 197 ms, 589 B

Menampilkan detail barang yang baru diedit

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

- URL:** `localhost/Minggu%2010/PWL_POS/public/api/barangs/51`
- Method:** GET
- Query Params:**

Key	Value	Description
Key	Value	Description
- Body:** JSON

```
[  {    "barang_id": 51,    "kategori_id": 1,    "barang_kode": "PS004",    "barang_nama": "Playstation 5",    "harga_beli": 5000000,    "harga_jual": 7000000,    "created_at": "2025-04-24T07:08:02.000000Z",    "updated_at": "2025-04-24T07:17:32.000000Z"  }]
```



NAMA : Dika Arie Arrifky
NIM : 2341720232
KELAS : TI-2A
MATERI : PWL - Jobsheet 10 - API Bagian 1

Menghapus data dengan id

HTTP localhost/Minggu%2010/PWL_POS/public/api/barangs/51

DELETE localhost/Minggu%2010/PWL_POS/public/api/barangs/51

Params Authorization Headers (6) Body Scripts Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (11) Test Results

{ } JSON Preview Visualize

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