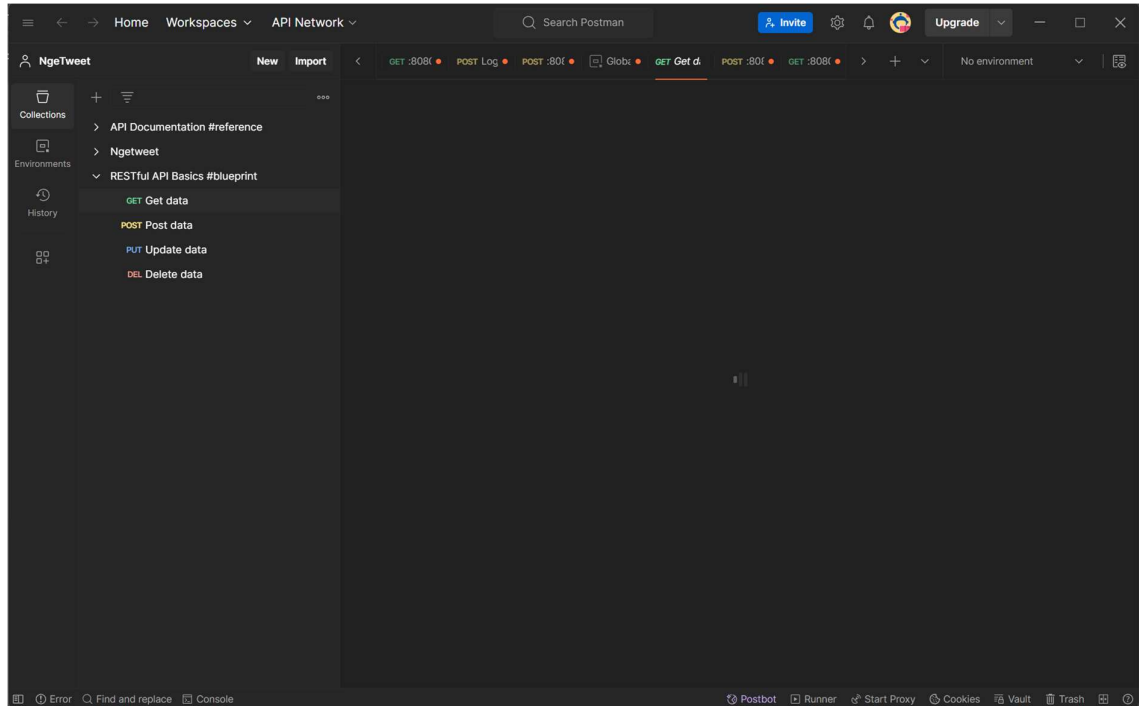


Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

1. Sebelum memulai membuat REST API, terlebih dahulu download aplikasi Postman di <https://www.postman.com/downloads>. Aplikasi ini akan digunakan untuk mengerjakan semua tahap praktikum pada Jobsheet ini.



2. Lakukan instalasi JWT dengan mengetikkan perintah berikut: `composer require tymon/jwt-auth:2.1.1` Pastikan Anda terkoneksi dengan internet.

```
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> composer require tymon/jwt-auth:2.1.1
./composer.json has been updated
Running composer update tymon/jwt-auth
Loading composer repositories with package information
Updating dependencies
Lock file operations: 4 installs, 0 updates, 0 removals
  - Locking lcobucci/clock (2.3.0)
  - Locking lcobucci/jwt (4.0.4)
  - Locking stella-maris/clock (0.1.7)
  - Locking tymon/jwt-auth (2.1.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 4 installs, 0 updates, 0 removals
  - 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)
  - Installing stella-maris/clock (0.1.7): Extracting archive
  - Installing lcobucci/clock (2.3.0): Extracting archive
  - Installing lcobucci/jwt (4.0.4): Extracting archive
  - Installing tymon/jwt-auth (2.1.1): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

jeroennoten/laravel-adminlte ..... DONE
laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... 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
```

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

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

```
No security vulnerability advisories found.      php artisan vendor:publish --provider="Tymon\JWTAuth\Providers\LaravelServiceProvider"

[INFO] Publishing assets.

Copying file [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\vendor\tymon\jwt-auth\config\config.php] to [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\config\jwt.php] DONE

PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

- Setelah itu jalankan perintah berikut untuk membuat secret key JWT. `php artisan jwt:secret`
Jika berhasil, maka pada file `.env` akan ditambahkan sebuah baris berisi nilai key `JWT_SECRET`.

```
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan jwt:secret
jwt-auth secret [V24hVCfByoDz2EmgMJvr21tYQgo9JDDdCvRy1cp2TQOMwdWqs02zHtEHsXg79FJUw] set successfully.
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

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

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

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

```
20 |
    0 references | 0 overrides
21 | public function getJWTIdentifier()
22 | {
23 |     return $this->getKey();
24 | }
25 |
    0 references | 0 overrides
26 | public function getJWTCustomClaims()
27 | {
28 |     return [];
29 | }
30 |
```

- 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`

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

```
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan make:controller Api/RegisterController

INFO Controller [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\app\Http\Controllers\Api\RegisterController.php] created successfully.

PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

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

```
0 references | 0 overrides
14 public function _invoke(Request $request)
15 {
16     $validator = Validator::make($request->all(), [
17         'username' => 'required',
18         'nama' => 'required',
19         'password' => 'required|min:5|confirmed',
20         'level_id' => 'required'
21     ]);
22
23     if ($validator->fails()) {
24         return response()->json($validator->errors(), 422);
25     }
26
27     $user = UserModel::create([
28         'username' => $request->username,
29         'nama' => $request->nama,
30         'password' => bcrypt($request->password),
31         'level_id' => $request->level_id
32     ]);
33
34     if ($user) {
35         return response()->json([
36             'success' => true,
37             'user' => $user,
38         ], 201);
39     }
40
41     return response()->json([
42         'success' => false,
43     ], 409);
44 }
45
46
47
```

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

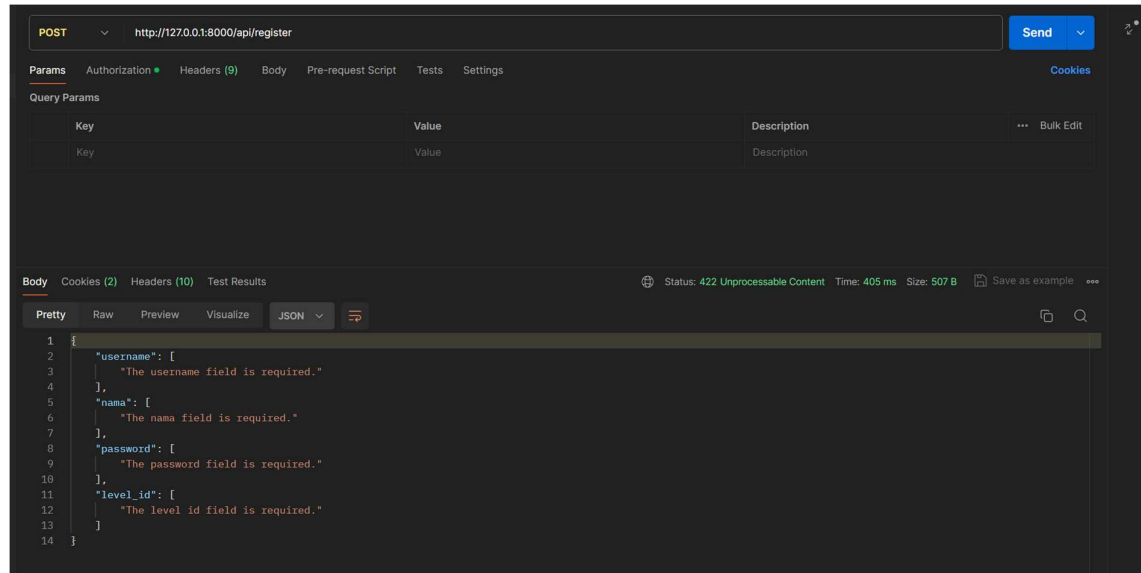
```
17
18 Route::post('/register', RegisterController::class)->name('register');
19
```

10. 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.

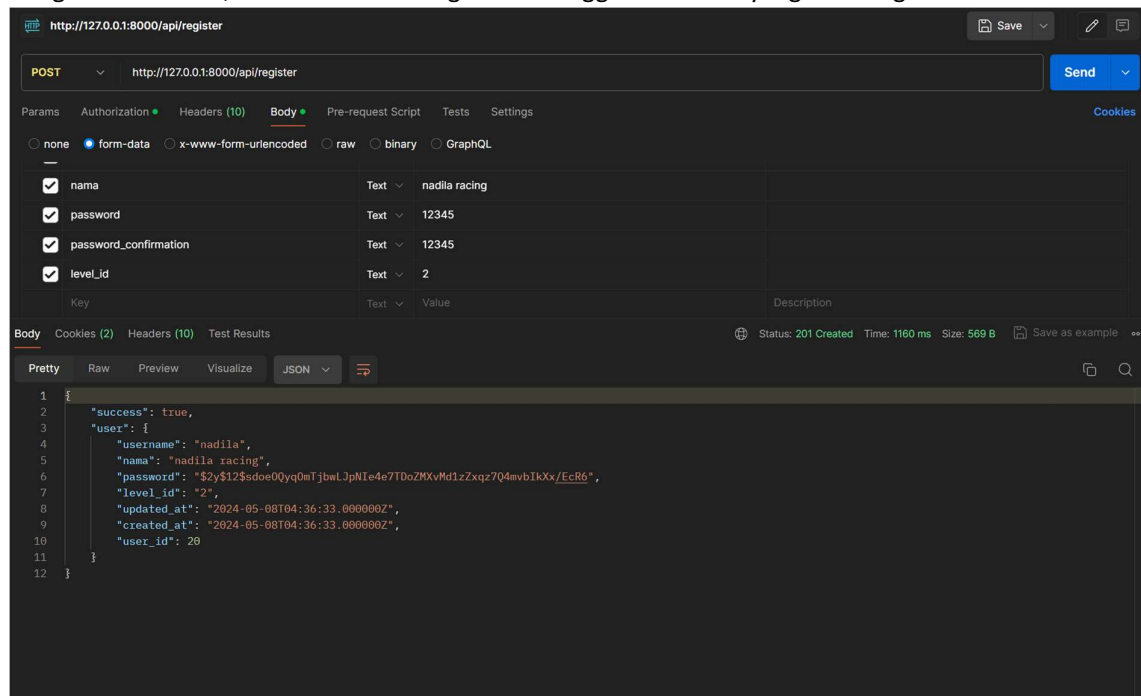
Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19



11. 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.



Praktikum 2

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 D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan make:controller Api/LoginController
INFO Controller [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\app\Http\Controllers\Api>LoginController.php] created successfully.
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

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

```
app > Http > Controllers > Api > LoginController.php > PHP > App\Http\Controllers\Api\LoginController.php
9      class LoginController extends Controller
10
11      //
12
13      0 references | 0 overrides
14      public function __invoke(Request $request)
15      {
16          $validator = Validator::make($request->all(), [
17              'username' => 'required',
18              'password' => 'required'
19          ]);
20
21          if ($validator->fails()) {
22              return response()->json($validator->errors(), 422);
23          }
24
25          $credentials = $request->only('username', 'password');
26
27          if (!$token = auth()->guard('api')->attempt($credentials)) {
28              return response()->json([
29                  'success' => false,
30                  'message' => 'Username atau Password Salah'
31              ]);
32          }
33
34          return response()->json([
35              'success' => 'true',
36              'user' => auth()->user(),
37              'token' => $token
38          ], 200);
39
40      }
41
42  }
```

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

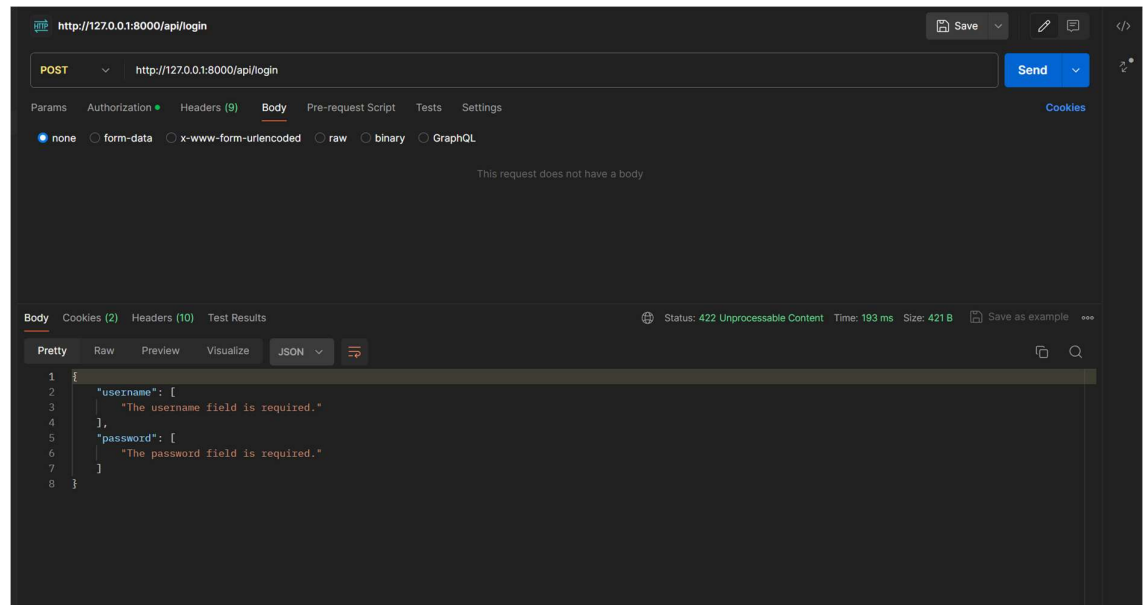
```
18
19  Route::post('/register', RegisterController::class)->name('register');
20  Route::post('/login', LoginController::class)->name('login');
21  Route::middleware('auth:api')->get('/user', function (Request $request) {
22      return $request->user();
23  });
24
```

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.

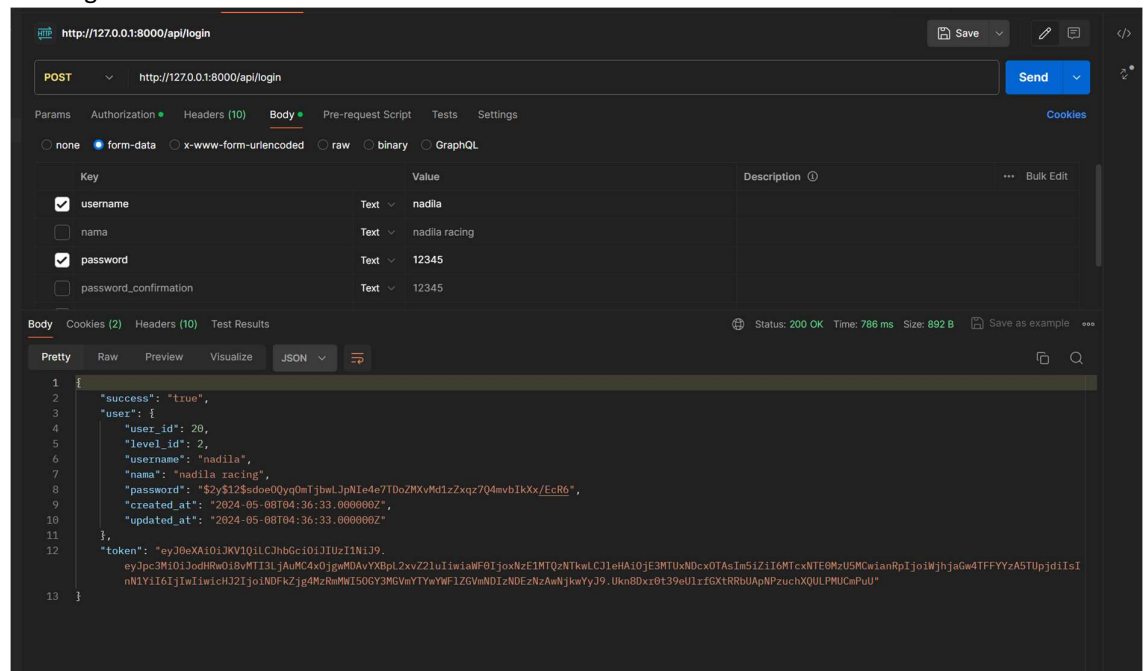
Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19



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.

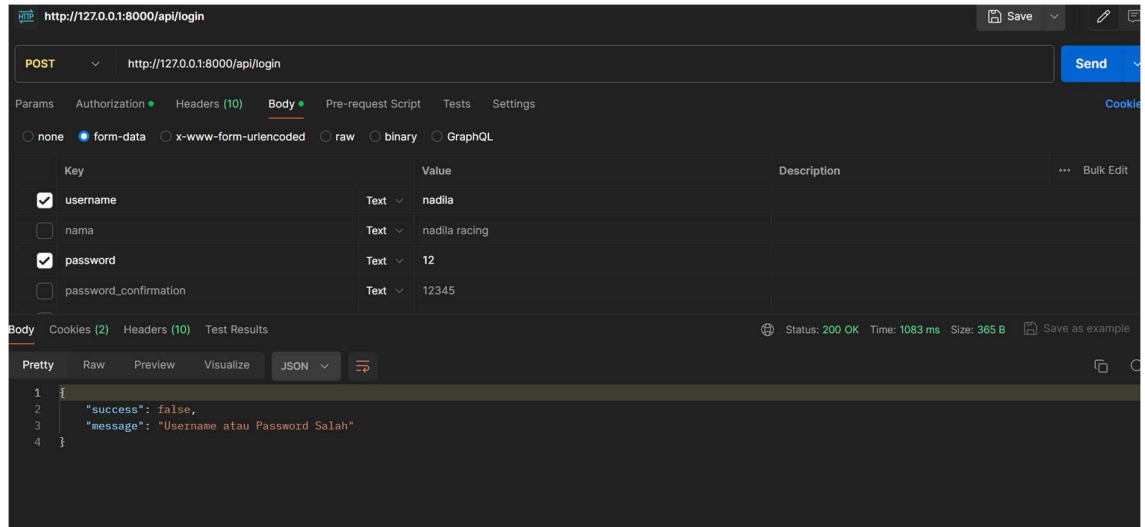


6. Untuk data salah

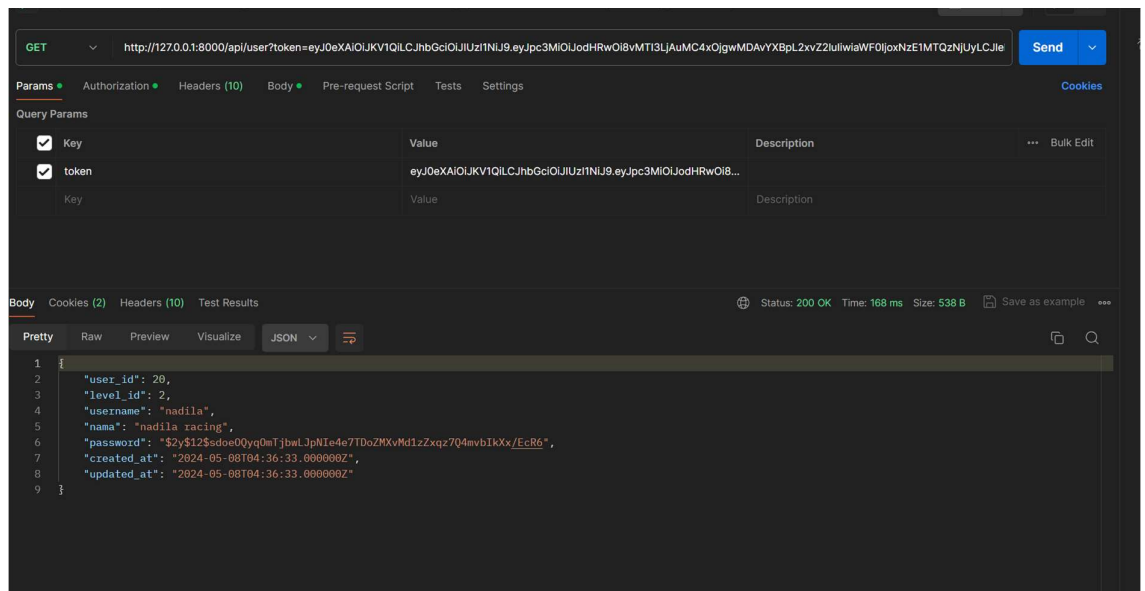
Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19



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.



Praktikum 3

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

```
60
61 JWT_SECRET=V24hVCfByoDz2EmgMJvr21tYQgo9JDdCvRy1cp2TQOMwdWqs02zHtEHsxg79FJUw
62 JWT_SHOW_BLACKLIST_EXCEPTION=true
```

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

```
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan make:controller Api/LogoutController
INFO Controller [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\app\Http\Controllers\Api\LogoutController.php] created successfully.
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

Absen : 19

- ```

2 references | 0 implementations
9 class LogoutController extends Controller
10 {
11 //
12
13 0 references | 0 overrides
14 public function __invoke(Request $request)
15 {
16 $removeToken = JWTAuth::invalidate(JWTAuth::getToken());
17
18 if ($removeToken) {
19 return response()->json([
20 'success' => true,
21 'message' => 'Logout Berhasil'
22]);
23 }
24 }

```

- ```
25  
26 Route::post('logout', LogoutController::class)->name('logout');
```

-

1. Pertama, buat controller untuk mengolah API pada data level. php artisan make:controller Api/LevelController

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

```
PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS> php artisan make:controller Api/LevelController

INFO Controller [D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS\app\Http\Controllers\Api\LevelController.php] created successfully.

PS D:\Kuliahe wong jenius\Semester 4\Web Lanjut\POS>
```

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

```
app > Http > Controllers > Api > LevelController.php > PHP > App\Http\Controllers\Api\LevelController > destroy()

9      class LevelController extends Controller
11     //
12     0 references | 0 overrides
13     public function index()
14     {
15         return LevelModel::all();
16     }
17
18     0 references | 0 overrides
19     public function store(Request $request)
20     {
21         $level = LevelModel::create($request->all());
22         return response()->json([
23             ]);
24     }
25
26     0 references | 0 overrides
27     public function show(LevelModel $level)
28     {
29         return LevelModel::find($level);
30     }
31
32     0 references | 0 overrides
33     public function update(Request $request, LevelModel $level)
34     {
35         $level->update($request->all());
36         return LevelModel::find($level);
37     }
38
39     0 references | 0 overrides
40     public function destroy(LevelModel $level)
41     {
42         $level->delete();
43         return response()->json([
44             'success' => true,
45             'message' => 'Data Terhapus'
46         ]);
47     }
48 }
```

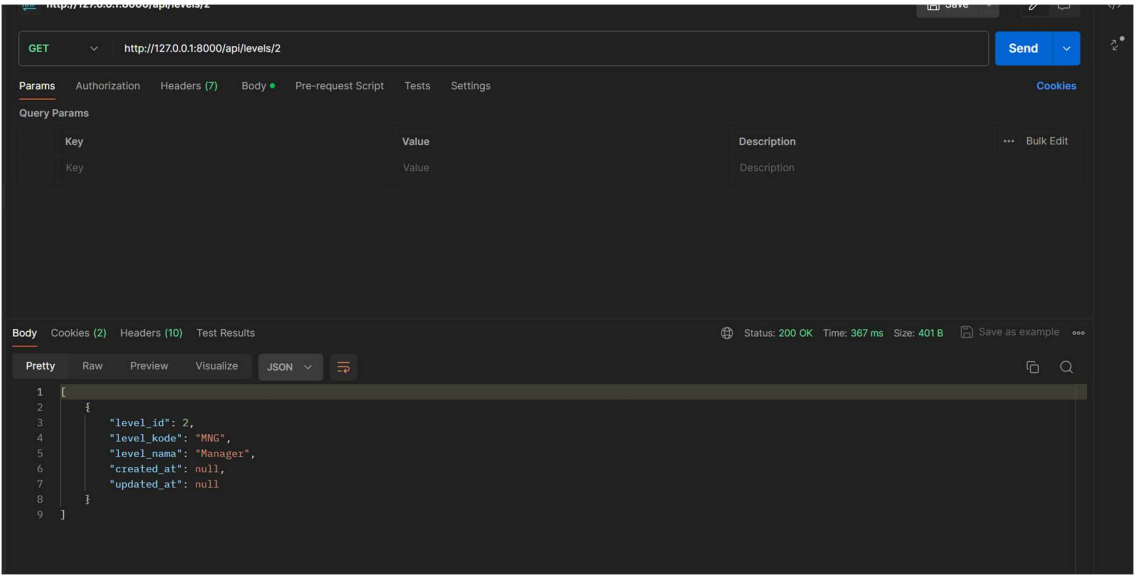
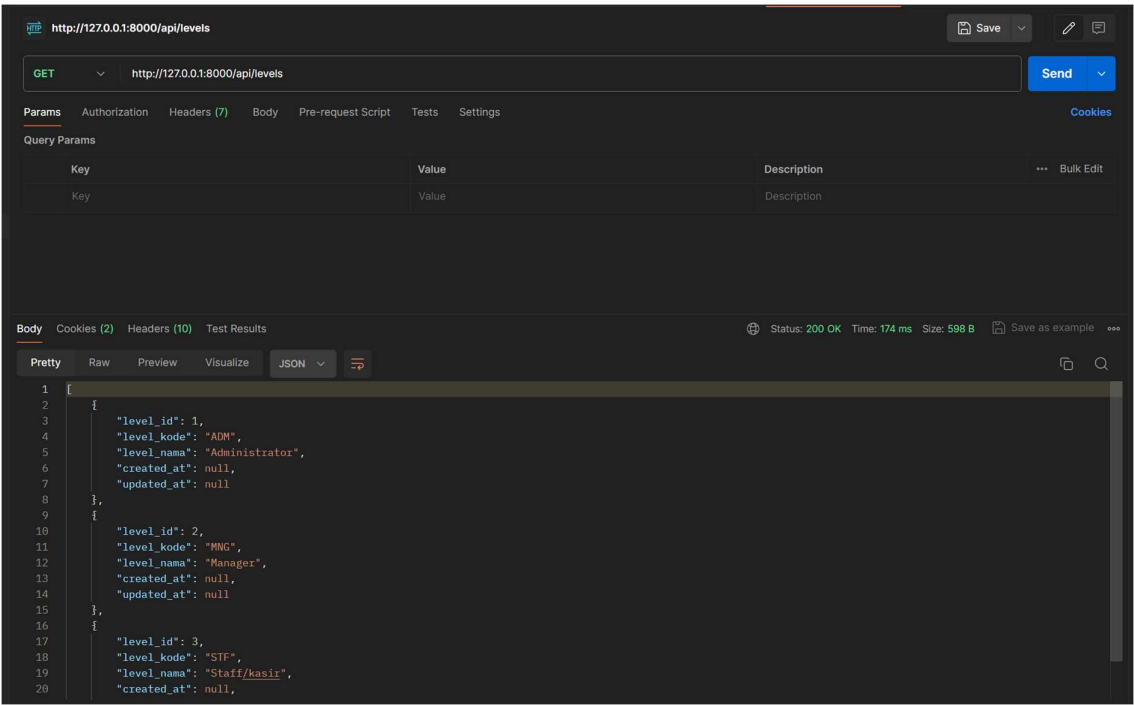
3. Kemudian kita lengkapi routes pada api.php

```
169
170 Route::get('levels', [LevelController::class, 'index']);
171 Route::post('levels', [LevelController::class, 'store']);
172 Route::get('levels/{$level}', [LevelController::class, 'show']);
173 Route::put('levels/{$level}', [LevelController::class, 'update']);
174 Route::delete('levels/{$level}', [LevelController::class, 'destroy']);
175
```

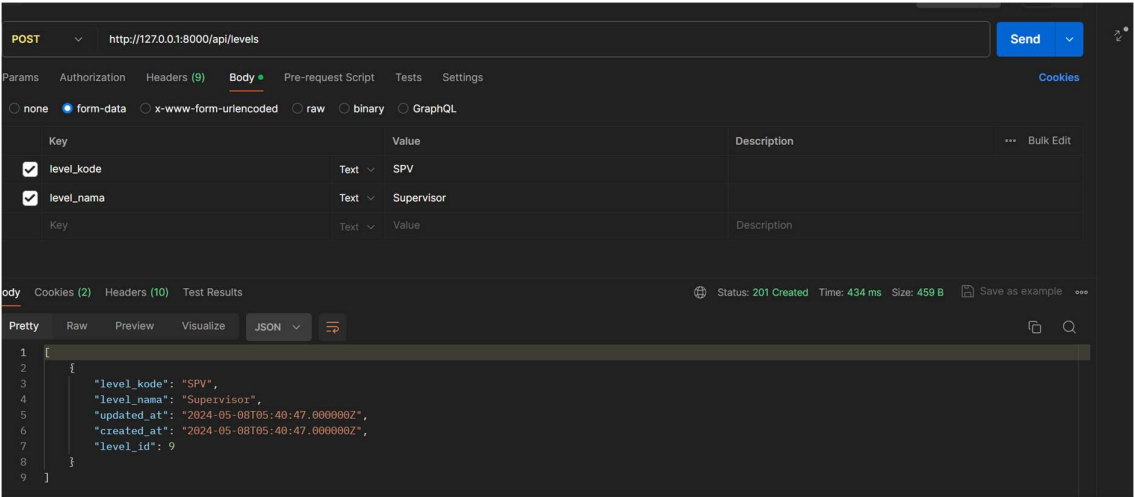
4. Jika sudah. Lakukan uji coba API mulai dari fungsi untuk menampilkan data. Gunakan URL: localhost/PWL_POS-main/public/api/levels dan method GET.

5. Show

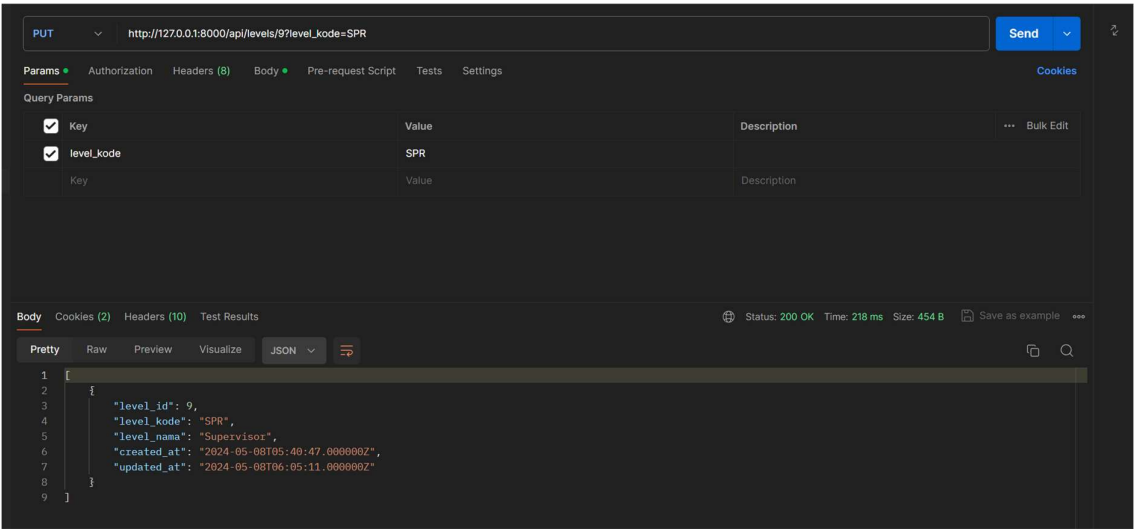
Nama : Krisna Andika Wijaya
Kelas : 2H
Absen : 19



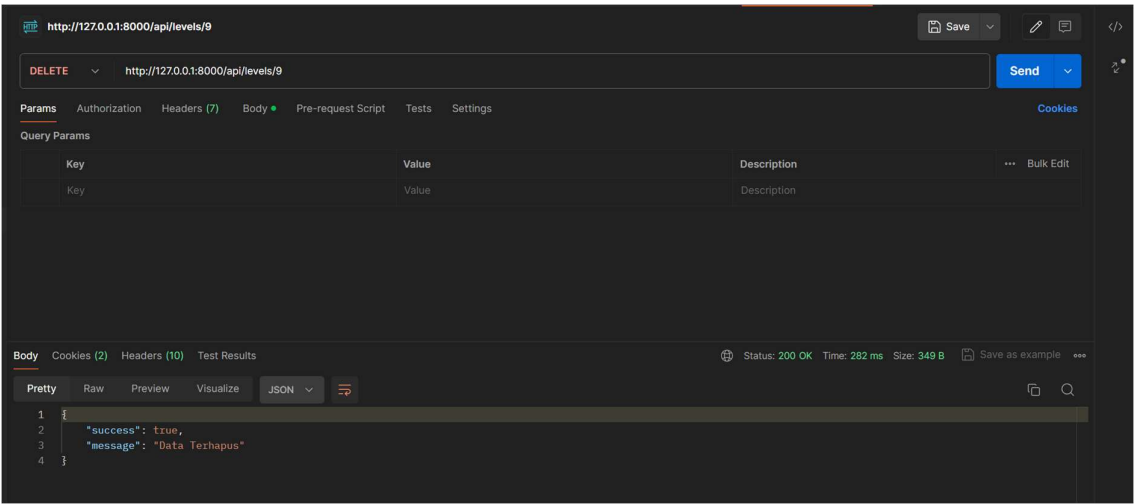
Nama : Krisna Andika Wijaya
Kelas : 2H
Absen : 19



7. Update



8. Delete



Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

Tugas

Kategori

```
5 references | 0 implementations
9 class KategoriController extends Controller
10 {
11     //
12
13     1 reference | 0 overrides
14     public function index()
15     {
16         return KategoriModel::all();
17     }
18
19     1 reference | 0 overrides
20     public function store(Request $request)
21     {
22         $level = KategoriModel::create($request->all());
23         return response()->json([
24             $level
25         ], 201);
26     }
27
28     1 reference | 0 overrides
29     public function show(KategoriModel $level)
30     {
31         return KategoriModel::find($level);
32     }
33
34     1 reference | 0 overrides
35     public function update(Request $request, KategoriModel $level)
36     {
37         $level->update($request->all());
38         return KategoriModel::find($level);
39     }
40
41     1 reference | 0 overrides
42     public function destroy(KategoriModel $level)
43     {
44         $level->delete();
45
46         return response()->json([
47             'success' => true,
48             'message' => 'Data Terhapus'
49         ]);
50     }
51 }
```

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

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

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

- Method:** GET
- URL:** http://127.0.0.1:8000/api/kategori
- Status:** 200 OK
- Time:** 2.33 s
- Size:** 986 B

The response body is displayed in JSON format, showing an array of three category objects:

```
[
  {
    "kategori_id": 1,
    "kategori_kode": "KTG01",
    "kategori_nama": "Alat Mandi",
    "created_at": null,
    "updated_at": null
  },
  {
    "kategori_id": 2,
    "kategori_kode": "KTG02",
    "kategori_nama": "Minuman",
    "created_at": null,
    "updated_at": null
  },
  {
    "kategori_id": 3,
    "kategori_kode": "KTG03",
    "kategori_nama": "Makanan",
    "created_at": null,
    "updated_at": null
  }
]
```

User

Nama : Krisna Andika Wijaya

Kelas : 2H

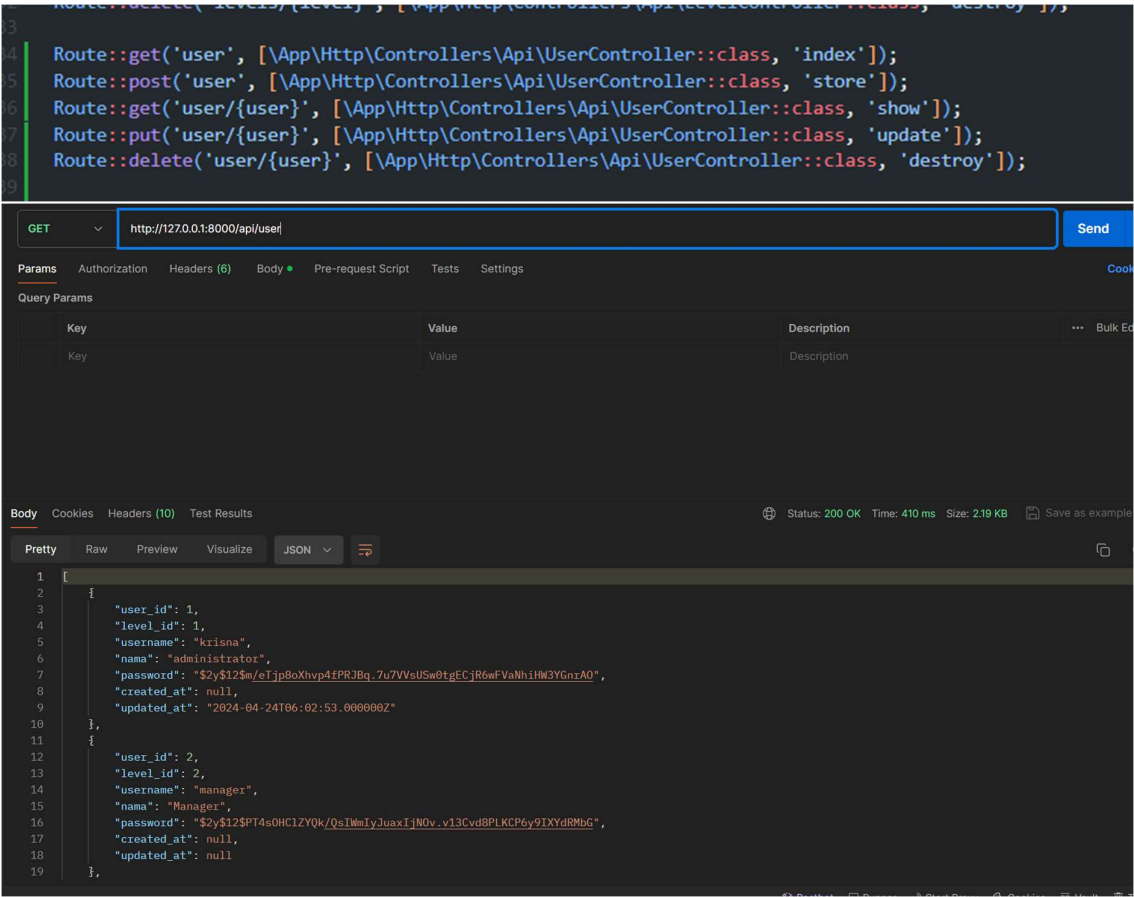
Absen : 19

```
App > Http > Controllers > Api > UserController.php > PHP > App\Http\Controllers\Api\UserCo
9  class UserController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index()
13     {
14         //
15         return UserModel::all();
16     }
17
18     1 reference | 0 overrides
19     public function store(Request $request)
20     {
21         //
22         $user = UserModel::create([
23             'level_id' => $request->level_id,
24             'username' => $request->username,
25             'nama' => $request->nama,
26             'password' => bcrypt($request->password),
27         ]);
28         return response()->json([
29             $user
30         ], 201);
31     }
32
33     1 reference | 0 overrides
34     public function show(UserModel $user)
35     {
36         //
37         return UserModel::findOrFail($user);
38     }
39
40     1 reference | 0 overrides
41     public function update(Request $request, UserModel $user)
42     {
43         //
44         $user->update(
45             [
46                 'level_id' => $request->level_id,
47                 'username' => $request->username,
48                 'nama' => $request->nama,
49                 'password' => bcrypt($request->password),
50             ]
51         );
52         return UserModel::find($user);
53     }
54 }
```


Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19



Barang

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

```
5 references | 0 implementations
 9 class BarangController extends Controller
10 {
11     //
12     1 reference | 0 overrides
13     public function index()
14     {
15         return BarangModel::all();
16     }
17
18     1 reference | 0 overrides
19     public function store(Request $request)
20     {
21         $level = BarangModel::create($request->all());
22         return response()->json([
23             $level
24         ], 201);
25     }
26
27     1 reference | 0 overrides
28     public function show(BarangModel $level)
29     {
30         return BarangModel::find($level);
31     }
32
33     1 reference | 0 overrides
34     public function update(Request $request, BarangModel $level)
35     {
36         $level->update($request->all());
37         return BarangModel::find($level);
38     }
39
40     1 reference | 0 overrides
41     public function destroy(BarangModel $level)
42     {
43         $level->delete();
44
45         return response()->json([
46             'success' => true,
47             'message' => 'Data Terhapus'
48         ]);
49     }
50 }
```

Nama : Krisna Andika Wijaya

Kelas : 2H

Absen : 19

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

GET http://127.0.0.1:8000/api/barang Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (10) Test Results Status: 200 OK Time: 196 ms Size: 1.93 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "barang_id": 1,
4     "kategori_id": 1,
5     "barang_kode": "SBN",
6     "barang_nama": "Sabun",
7     "harga_beli": 6000,
8     "harga_jual": 10000,
9     "created_at": null,
10    "updated_at": null
11  },
12  {
13    "barang_id": 2,
14    "kategori_id": 1,
15    "barang_kode": "SMP",
16    "barang_nama": "Sampo",
17    "harga_beli": 10000,
18    "harga_jual": 15000,
19    "created_at": null,
```